

CubeSuite Ver.1.40

Integrated Development Environment

User's Manual: Start

Target Device

78K0 Microcontroller

78K0R Microcontroller

V850 Microcontroller

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

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
 - “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
 - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
 - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

How to Use This Manual

This manual describes the role of the CubeSuite integrated development environment for developing applications and systems for 78K0 microcontrollers, 78K0R microcontrollers and V850 microcontrollers, and provides an outline of its features.

CubeSuite is an integrated development environment (IDE) for 78K0 microcontrollers, 78K0R microcontrollers and V850 microcontrollers, 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 GLOSSARY
APPENDIX C HOW THE MANUALS ARE ORGANIZED
APPENDIX D INPUT CONVENTIONS
APPENDIX E USING AN EXTERNAL BUILD TOOL
APPENDIX F Python CONSOLE/Python FUNCTIONS
APPENDIX G 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{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	This manual
	Analysis	R20UT0265E
	Programming	R20UT0266E
	Message	R20UT0267E
	Coding for CX compiler	R20UT0259E
	Build for CX compiler	R20UT0261E
	78K0 Coding	R20UT0004E
	78K0 Build	R20UT0005E
	78K0 Debug	R20UT0262E
	78K0 Design	R20UT0006E
	78K0R Coding	U19382E
	78K0R Build	U19385E
	78K0R Debug	R20UT0263E
	78K0R Design	R20UT0007E
	V850 Coding	U19383E
	V850 Build	U19386E
	V850 Debug	R20UT0264E
V850 Design	R20UT0257E	

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

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

[MEMO]

[MEMO]

[MEMO]

TABLE OF CONTENTS

CHAPTER 1 GENERAL ... 11

- 1.1 Introduction ... 11
- 1.2 Features ... 11
- 1.3 System Configuration ... 13
- 1.4 Operating Environment ... 14

CHAPTER 2 FUNCTIONS ... 15

- 2.1 Installing CubeSuite ... 15
- 2.2 Uninstalling CubeSuite ... 21
- 2.3 Using the License Management Feature ... 23
 - 2.3.1 Show license information ... 23
 - 2.3.2 Add a license ... 24
 - 2.3.3 Delete a license ... 24
- 2.4 Using the Update Feature ... 25
 - 2.4.1 The manual update feature ... 25
 - 2.4.2 The auto update feature ... 30
 - 2.4.3 Canceling an update ... 32
 - 2.4.4 Resuming an update ... 32
 - 2.4.5 Displaying the update history ... 33
 - 2.4.6 Restoring hidden updates ... 36
- 2.5 Start CubeSuite ... 38
- 2.6 Create a Project ... 39
 - 2.6.1 Start a project ... 42
 - 2.6.2 Create a new project ... 43
 - 2.6.3 Add a new subproject ... 46
- 2.7 Manipulate a Project ... 48
 - 2.7.1 Open a project ... 48
 - 2.7.2 Add an existing subproject ... 49
 - 2.7.3 Project is added to the favorites menu ... 50
 - 2.7.4 Remove a subproject from the project ... 51
 - 2.7.5 Change the project name ... 51
 - 2.7.6 Open a project folder in Explorer ... 52
 - 2.7.7 Convert a CA850 project into a CX project ... 53
 - 2.7.8 Convert a PM+ project into a CubeSuite project ... 57
- 2.8 Save the Project File ... 60
 - 2.8.1 Save the project file with a different name ... 60
 - 2.8.2 Save all files ... 61
 - 2.8.3 Pack and save the project and CubeSuite ... 61
 - 2.8.4 Close a project ... 64

- 2.9 Changing the Window Layout ... 65
 - 2.9.1 Automatically hide panels ... 65
 - 2.9.2 Floating a window ... 65
 - 2.9.3 Docking windows ... 66
 - 2.9.4 Displaying multiple panels ... 66
 - 2.9.5 Resetting the window layout ... 67
- 2.10 Execute Python Functions ... 68
- 2.11 Manipulate CubeSuite on the Command Line ... 69

APPENDIX A WINDOW REFERENCE ... 73

- A.1 Description ... 73

APPENDIX B GLOSSARY ... 236

APPENDIX C HOW THE MANUALS ARE ORGANIZED ... 237

APPENDIX D INPUT CONVENTIONS ... 239

- D.1 Input Conventions ... 239
- D.2 Displaying Icons at Locations of Input Errors ... 240

APPENDIX E USING AN EXTERNAL BUILD TOOL ... 241

- E.1 Overview ... 241
- E.2 Create a Debug-dedicated Project ... 242
- E.3 Add a File to a Project ... 243
 - E.3.1 Add a download file ... 243
 - E.3.2 Add C source files and other files ... 244
 - E.3.3 Remove an added file from a project ... 247
- E.4 Make Settings for Build Operations ... 248
 - E.4.1 Set the commands ... 248
 - E.4.2 Set the build mode ... 250
 - E.4.3 Set the target project for a build ... 253
- E.5 Run a Build ... 254
 - E.5.1 Run a build ... 255
 - E.5.2 Run a rebuild ... 256
 - E.5.3 Run a clean ... 256
 - E.5.4 Run a rapid build ... 258
 - E.5.5 Run a batch build ... 259
 - E.5.6 Stop running a build ... 259
 - E.5.7 Save the build results to a file ... 260

APPENDIX F Python CONSOLE/Python FUNCTIONS ... 261

- F.1 Overview ... 261
- F.2 Related File ... 261
- F.3 CubeSuite Python Functions ... 261

CHAPTER 1 GENERAL

This chapter describes the role of the CubeSuite integrated development environment for developing applications and systems for the 78K0 microcontroller, 78K0R microcontroller and V850 microcontroller, and provides an outline of its features.

1.1 Introduction

CubeSuite is an integrated development environment (IDE)^{Note} for the 78K0 microcontroller, 78K0R microcontroller and V850 microcontroller.

By integrating the necessary tools for each development phase, it is possible to perform all phases in software development using just this product, without the need to use many different tools separately. Emphasis is placed on making the tools work together, improving development efficiency in many different situations. As an example, the output from the design phase is automatically reflected in the debug phase.

CubeSuite also has an update feature to automatically obtain this product upgrades via the network, making it simple to maintain the environment required for software development (free downloadable tools excluded).

Note An integrated development environment is a development environment integrating the necessary tools for all phases of software development, including design, implementation (coding and building), and debugging, into a single platform framework.

Remark In addition to CubeSuite, emulators and on-chip debugging emulators (emulators for microcontrollers with on-chip debugging facilities built in) are provided, as well as a real-time OS package (for developing systems using real-time OSes), making a wide range of development possible.

1.2 Features

CubeSuite's features are shown below.

(1) Project management

Manage project information, including source-file structure, build options, and settings for connecting to the debug tool.

(2) Design

The pin assignment function makes it possible to output reports called "device pin list" and "device top view" as files, by inputting the pin-configuration status of the microcontroller.

The code generation function^{Note} can output source code (device driver programs) corresponding to peripheral functions provided by the microcontroller (e.g. systems, ports, and interrupt) by selecting and entering the information required for control in the CubeSuite panels.

Note The code generation function is an optional feature.

(3) Coding

A tree view of the files included in the project appears in a CubeSuite panel, and the files can be edited by linking various editors to CubeSuite.

(4) Build

You can configure optimization and other build options in the CubeSuite panels, enabling you to create efficient load module files and a library file.

Remark It is also possible to link to an external build tool and use it instead of the build tool provided by CubeSuite (see "[APPENDIX E USING AN EXTERNAL BUILD TOOL](#)").

(5) Debug

You can display your debugging tool's connection settings and debugging information in CubeSuite panels. There are also many methods for executing programs, enabling you to debug your programs efficiently.

(6) Analysis

You can analyze the source program and information while the program is executing, and display information about the functions and variables.

(7) Flash programming

You can execute commands on the microcontroller's onboard Flash memory, including blank check, erase, write, verify, and read, by configuring information on the CubeSuite panels.

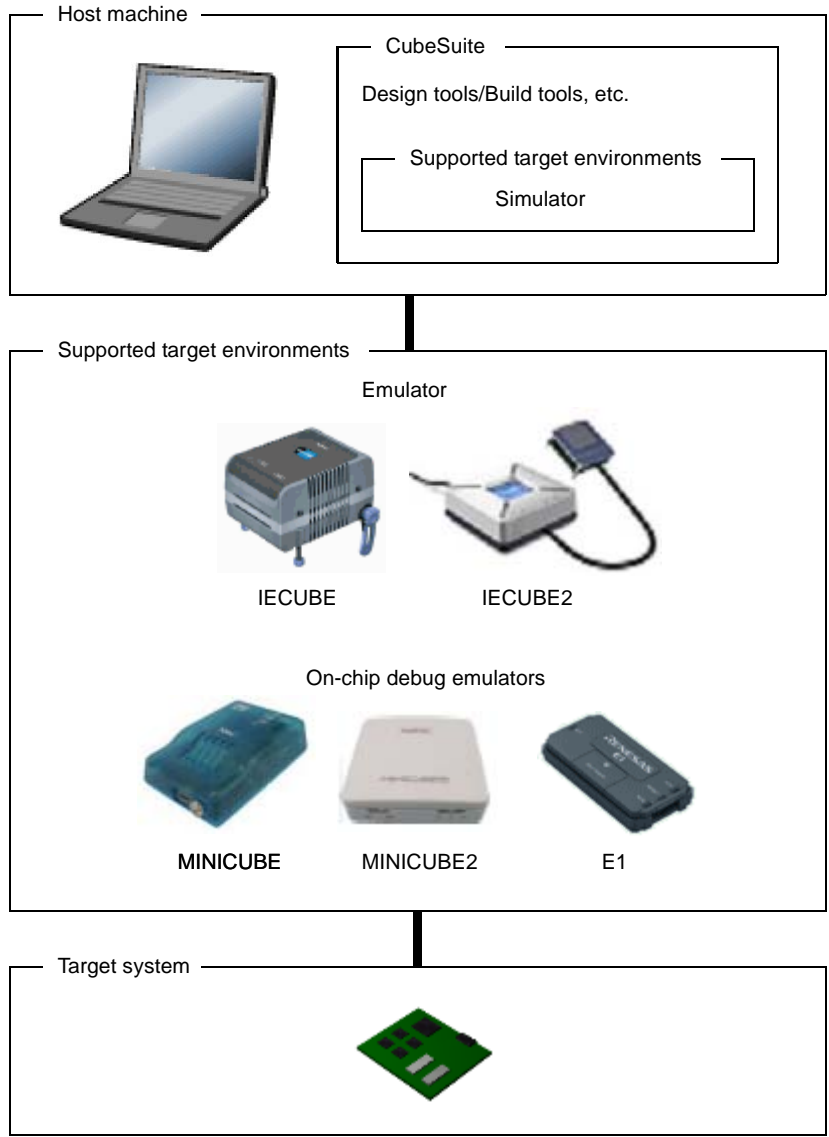
(8) Updates

Communicate with the update server to get the latest version of this product.

1.3 System Configuration

Below is shown an example of the system configuration.

Figure 1-1. System Configuration



1.4 Operating Environment

Below are the system requirements for this product.

(1) Hardware environment

Processor:	At least 1 GHz (support for hyper threading/multi-core CPU)
Main memory:	At least 512 MB (1 GB or higher recommended)
Display:	Resolution at least 1,204 x 768; at least 65,536 colors
Interface:	USB 2.0

(2) Software environment

- Windows XP (Only 32-bit OS)
 - Professional
 - Home Edition
- Windows Vista (32-bit OS, 64-bit OS)
 - Business Edition
 - Enterprise Edition
- Windows 7 (32-bit OS, 64-bit OS)
- .NET Framework
 - .NET Framework 2.0 (Windows XP)
 - .NET Framework 3.0 (Windows Vista)
 - .NET Framework 3.5.1 (Windows 7)

Remark .NET Framework 3.5 Client Profile is not supported

- Runtime library of Microsoft Visual C++ 2005 SP1
- Internet Explorer 6.0 or higher

Remark Installation of the latest service pack is recommended.

(3) Supported target environments

- Emulator
 - IECUBE
 - IECUBE2
- On-chip debug emulators
 - MINICUBE
 - MINICUBE2
 - E1
- Simulator

CHAPTER 2 FUNCTIONS

This chapter describes how to install CubeSuite, how to use the license management and update feature, and the procedure from launching CubeSuite to starting development.

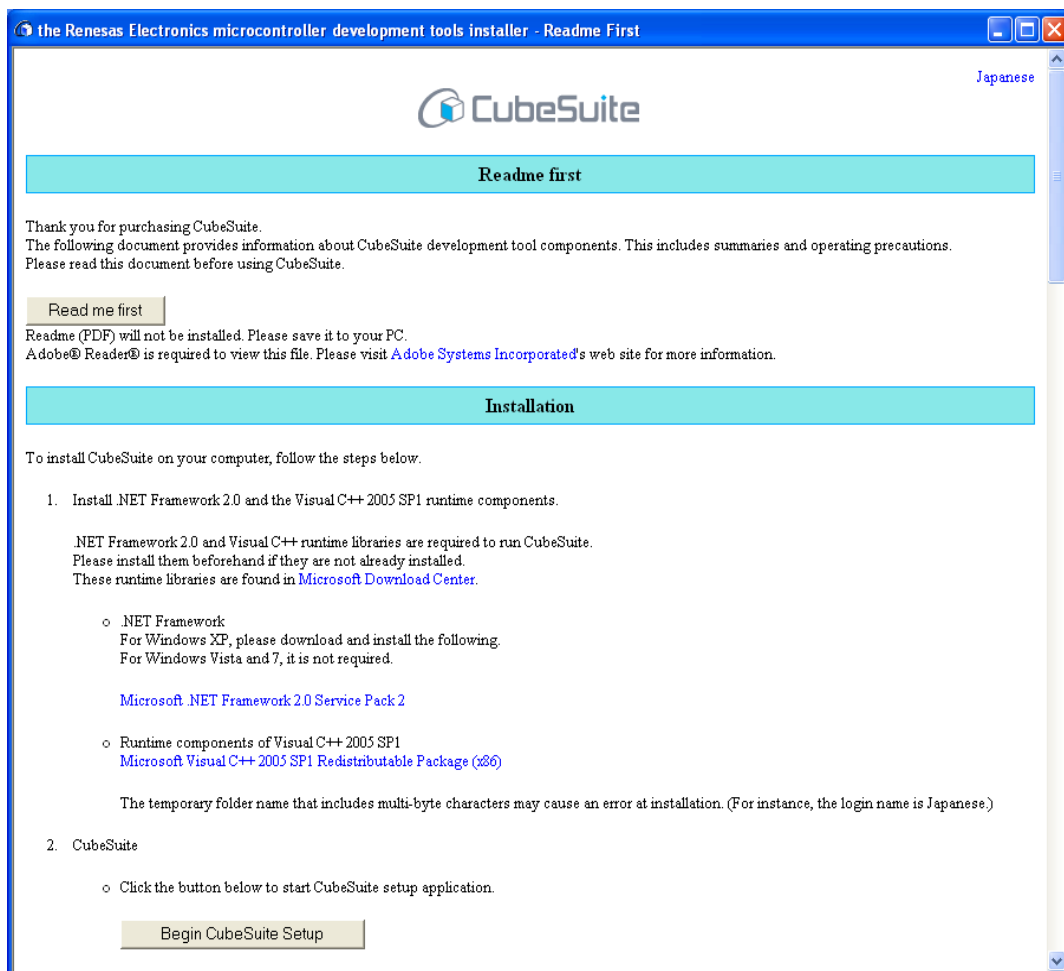
2.1 Installing CubeSuite

This section describes how to install CubeSuite.

(1) Insert the CD-ROM into the drive.

The Preparing to Install page appears automatically.

Figure 2-1. Preparing to Install Page



Click the [Begin CubeSuite Setup] button, CubeSuite setup begins.

- Cautions 1. CubeSuite setup must be conducted with administrator privileges.**
2. **In order to install CubeSuite, you must have the .NET Framework (2.0 for Windows XP, 3.0 for Windows Vista, 3.5.1 for Windows 7) and Runtime library of Microsoft Visual C++ 2005 SP1. Please install these runtime components before installing CubeSuite. Each runtime component can be installed from the Preparing to Install window.**

Remark If the page does not appear automatically, open "Install.hta" in the CD-ROM.

(2) Confirm the product information, etc.

Product information, etc. appears.

Figure 2-2. Initial Window

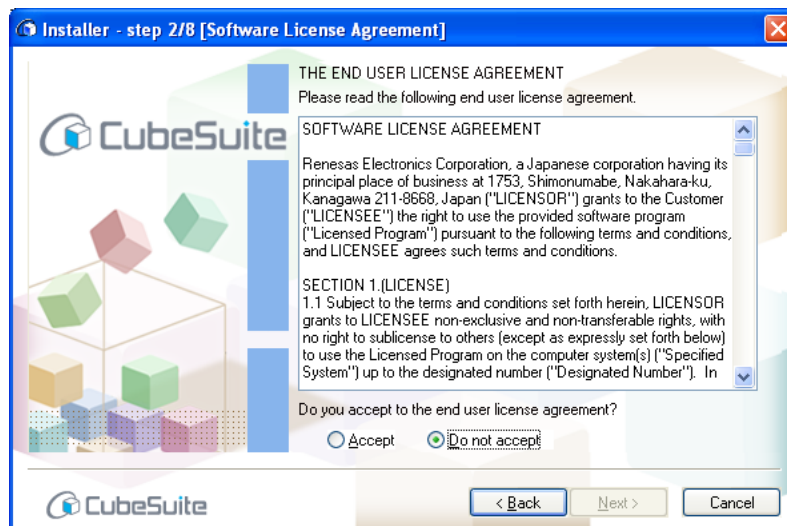


Check the information, then click the [Next] button.

(3) Confirm the software license agreement.

The installer asks if you agree to the SOFTWARE LICENSE AGREEMENT.

Figure 2-3. Software License Agreement Window



Check the information, and if you agree it, select "Accept", then click the [Next] button.

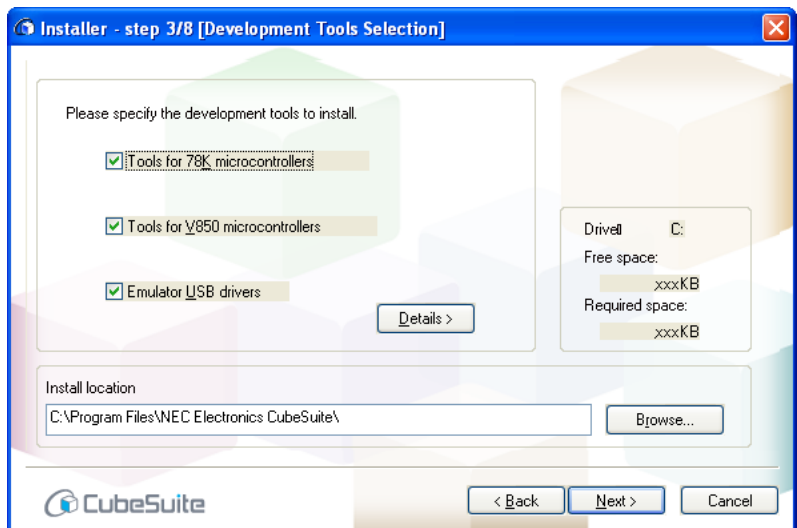
Caution If you select "Do not accept", you cannot continue with the installation.

(4) Select the install target and the installation location.

Select the check boxes of the install target.

If you wish to change the installation location, edit it in the [Install location] area.

Figure 2-4. Development Tools Selection Window

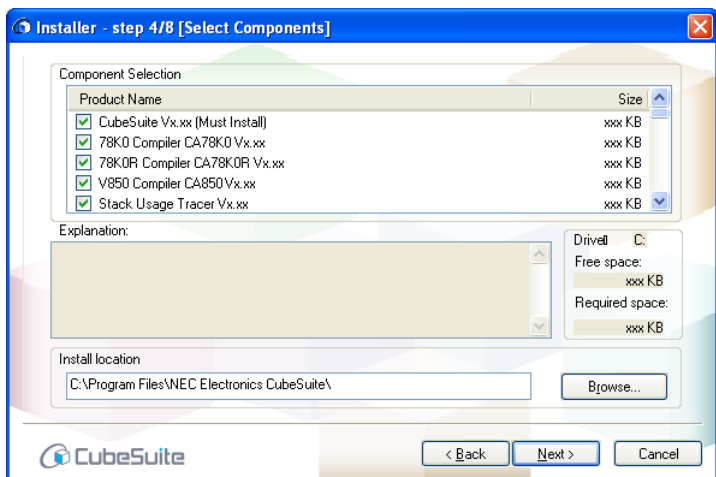


After you select the install target and installation location, click the [Next] button.

- Cautions**
1. If you wish to install the code generation plug-in, click the [Details] button to select it in the **Select Components window**.
 2. You cannot change the install location on a host machine where CubeSuite have been installed. If you wish to change the installation location, uninstall CubeSuite, then perform the installation.

Remark If you wish to specify the install target in detail, click the [Details] button. The Select Components window appears. The [Component Selection] area shows the details of the install target you selected in the Development Tools Selection window. After select the check boxes of the tools in the [Component Selection] area, click the [Next] button.

Figure 2-5. Select Components Window



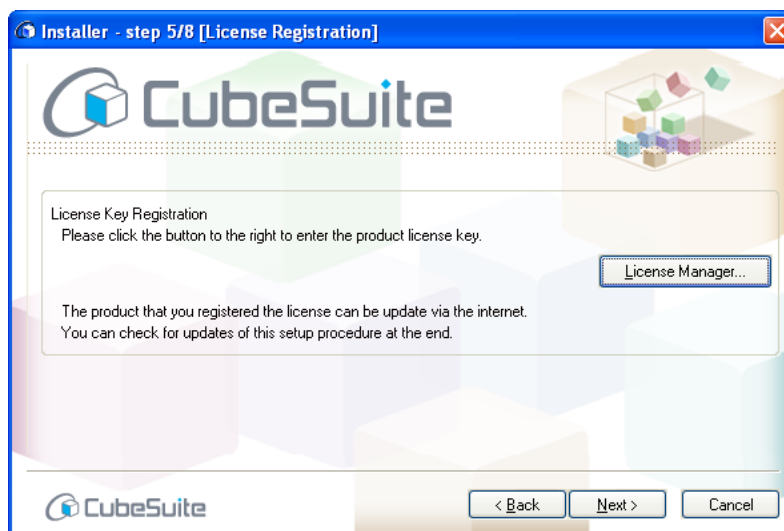
Caution The check boxes of products that must be installed cannot be cleared.

(5) Enter your license key.

Registering a license will remove the following limitations.

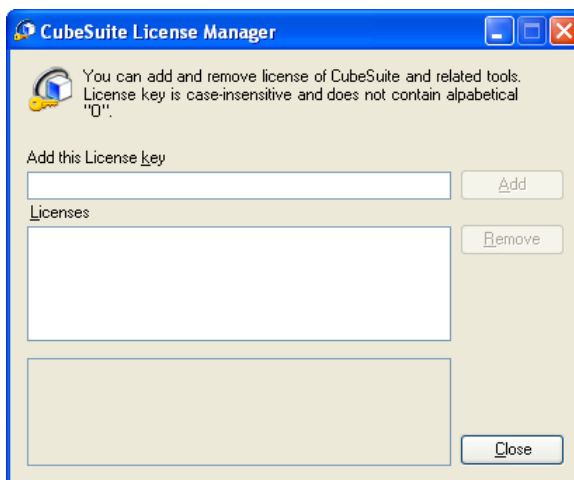
- Update Manager will not be available.
- Below are the maximum code sizes that can be developed.
 - 78K0 microcontroller: 32 Kbytes
 - 78K0R microcontroller: 64 Kbytes
 - V850 microcontroller: 128 Kbytes
- The Tool Support Center Contact Information feature will not be available.

Figure 2-6. License Registration Window



Click the [License Manager...] button.

Figure 2-7. CubeSuite License Manager Window



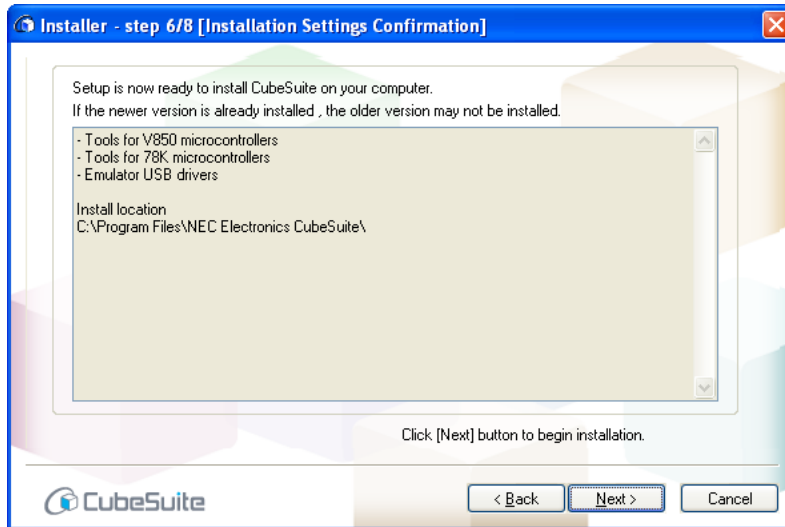
After you register the license in the [CubeSuite License Manager window](#), click the [Next] button in the License Registration Window.

Caution The installer cannot be manipulated while the **CubeSuite License Manager window** is running.

(6) Check the installation targets and installation.

The settings made in the Development Tools Selection window or Select Components window are appeared.

Figure 2-8. Installation Settings Confirmation Window



Check the information, and click the [Next] button.

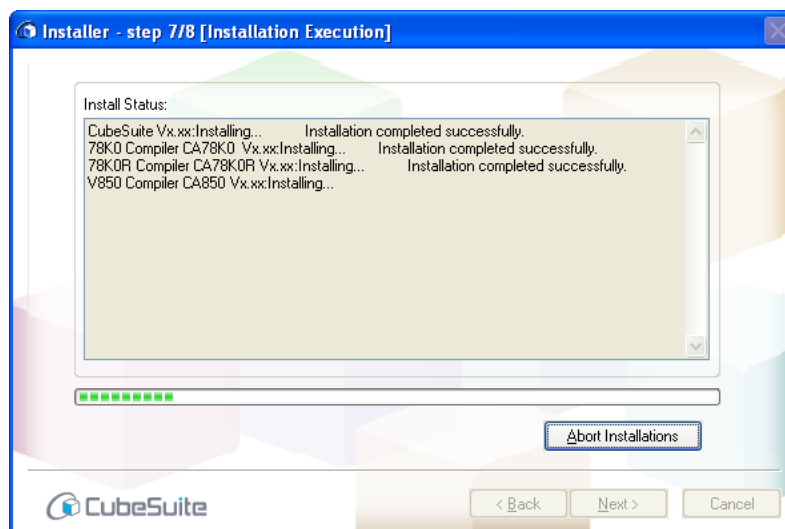
Caution If a emulator USB driver is installed on Windows Vista or Windows 7, the warning dialog box may appear.

(7) Confirm the installation progress.

A progress bar displays the progress of the installation.

The installation progress of each tool appear in the [Install Status] area.

Figure 2-9. Installation Execution Window



When all installations of the install targets are finished, the [Next] button becomes enabled.

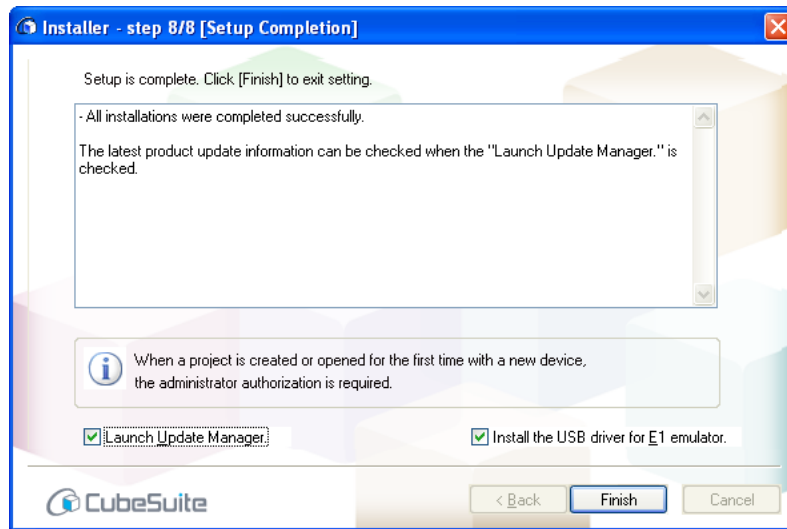
Check the information, and click the [Next] button.

Remark If you click the [Abort Installations] button, a message stating that the user chose to cancel the installation appears in the [Install Status] area. The installation will halt when the installation of the tool currently being installed finishes.

(8) Confirm the results of the installation.

When all setup-related tasks are finished, the results of the installation appear.

Figure 2-10. Setup Completion Window



Click the [Finish] button to complete the installation.

Remark When having a check in a [Launch Update Manager.] check box, the [CubeSuite Update Manager window](#) opens after completing the installation. See "[2.4 Using the Update Feature](#)" for the update method.

2.2 Uninstalling CubeSuite

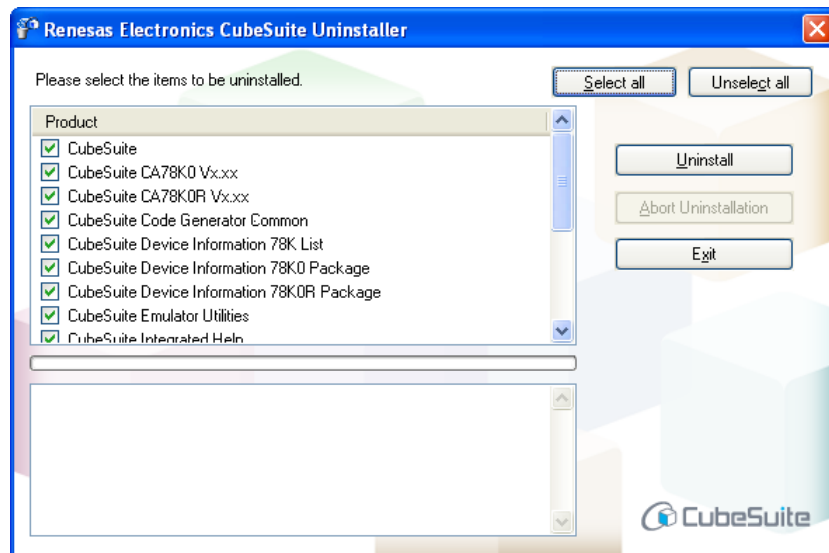
You can uninstall CubeSuite using the CubeSuite Uninstaller.

(1) Launch CubeSuite Uninstaller.

From the Windows the [Start] menu, select [Programs] >> [NEC Electronics CubeSuite] >> [CubeSuite Uninstaller].

The [CubeSuite Uninstaller window](#) is launched.

Figure 2-11. CubeSuite Uninstaller Window



Click the [Select all] button to select all the check boxes.

Click the [Unselect all] button to clear all the check boxes.

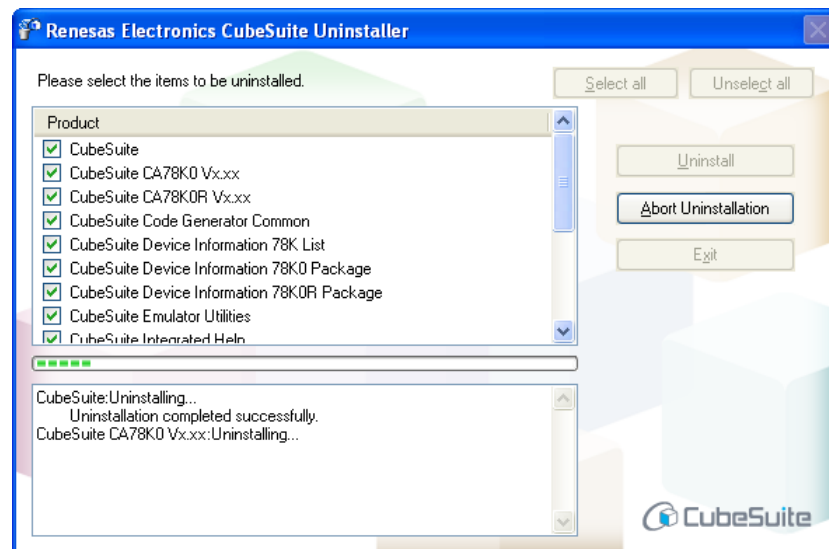
Click the [Uninstall] button.

(2) Confirm the uninstallation progress.

A progress bar displays the progress of the uninstallation.

The uninstallation progress of each tool appear in the progress details area.

Figure 2-12. Uninstallation Execution Window (Progress)

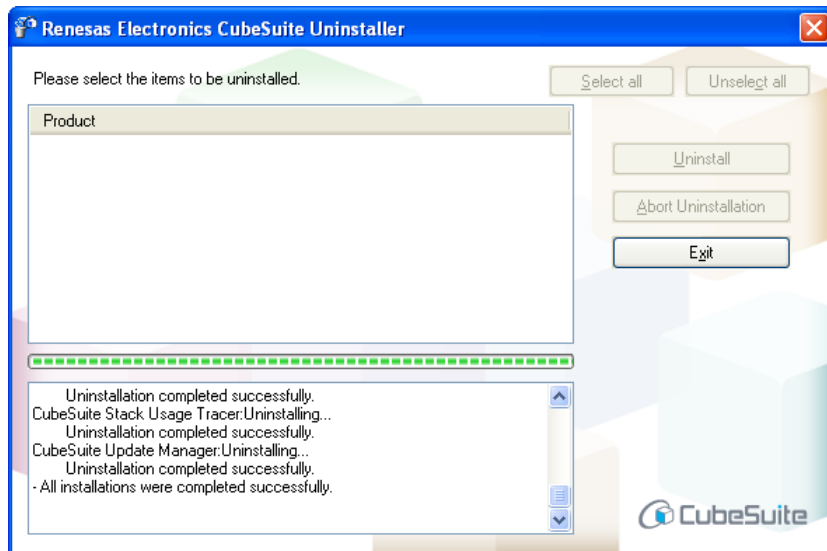


Remark If you click the [Abort Uninstallations] button, a message stating that the user chose to cancel the uninstallation appears in the progress details area.

(3) Confirm the results of the uninstallation.

When all setup-related tasks are finished, the results of the uninstallation appear.

Figure 2-13. Uninstallation Completion Window (Result)



Click the [Exit] button to complete the uninstallation.

Caution If all products are uninstalled, the CubeSuite Uninstaller will be also uninstalled automatically.

2.3 Using the License Management Feature

The license management feature registers and manages newly acquired licenses for CubeSuite and related tools on the host machine by starting the [CubeSuite License Manager window](#) using the method below.

- From the Windows the [Start] menu, select [Programs] >> [NEC Electronics CubeSuite] >> [CubeSuite License Manager]
- Upon installation, in the License registration window, click the [License Manager...] button

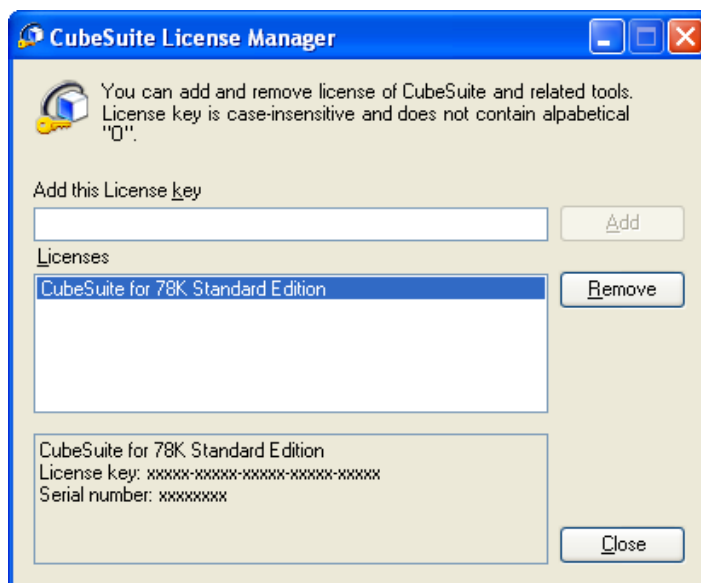
The following functionality will be restricted if a license is not registered.

- Update Manager will not be available.
- Below are the maximum code sizes that can be developed.
 - 78K0 microcontroller: 32 Kbytes
 - 78K0R microcontroller: 64 Kbytes
 - V850 microcontroller: 128 Kbytes
- The Tool Support Center Contact Information feature will not be available.

2.3.1 Show license information

When the [CubeSuite License Manager window](#) starts, valid registered licenses area shown in the [Licenses] area.

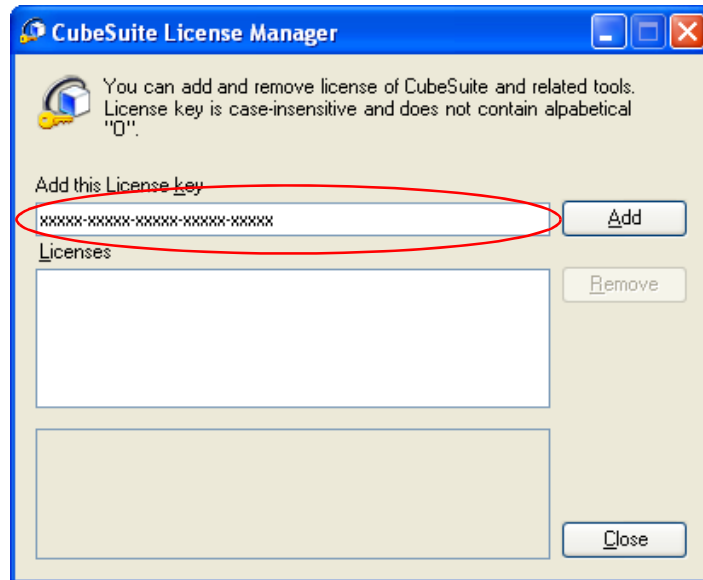
Figure 2-14. CubeSuite License Manager Window



2.3.2 Add a license

In the [CubeSuite License Manager window](#), in the [Add this License key] text box, enter a license key, then click the [Add] button to add the license.

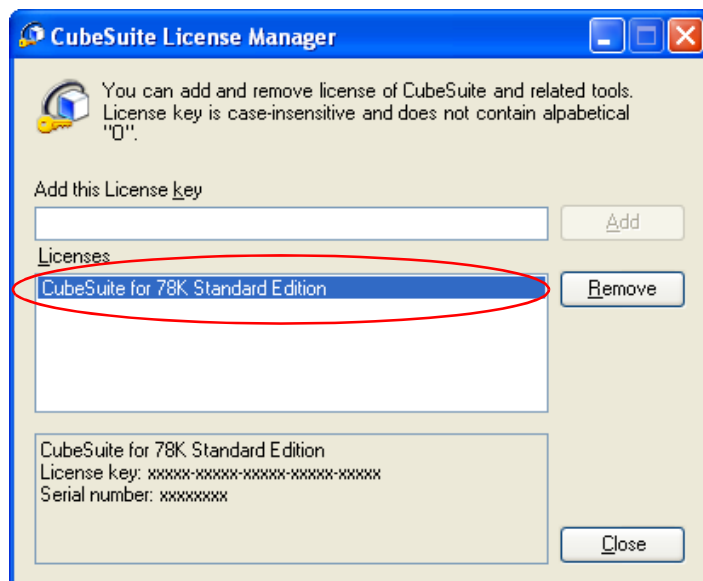
Figure 2-15. CubeSuite License Manager Window



2.3.3 Delete a license

In the [CubeSuite License Manager window](#), select the license you wish to delete from the [Licenses] area, then click the [Remove] button to delete the license.

Figure 2-16. CubeSuite License Manager Window



2.4 Using the Update Feature

The update feature updates installed tools and documents to their latest versions.

It acquires information about the latest versions via the Internet, and then downloads and installs them.

There are two update methods: manual update, where you perform the update guided by information displays; and auto update, which automatically updates according to your settings.

- Cautions 1. When you use the update feature, the host machine must have a connection to the Internet.**
- 2. You must register the license to use the update feature.**
See "2.3 Using the License Management Feature" for how to register the license.

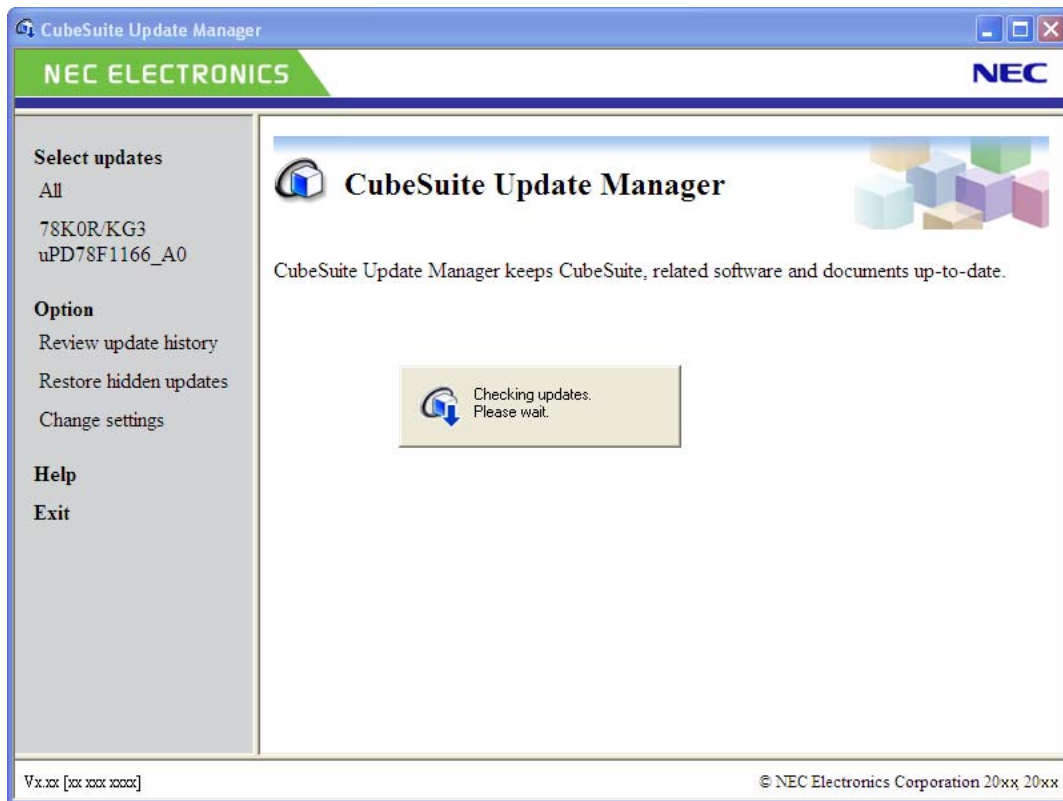
2.4.1 The manual update feature

This section describes the manual update.

(1) Launch Update Manager.

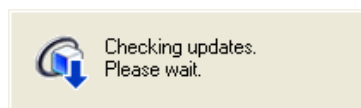
From the Windows the [Start] menu, select [Programs] >> [NEC Electronics CubeSuite] >> [CubeSuite Update Manager], or from the CubeSuite menu, select [Help] >> [Check for Updates...]. The [CubeSuite Update Manager window](#) launches.

Figure 2-17. CubeSuite Update Manager Window



Remark The [Checking for Updates dialog box](#) appears as the manager checks for update information.

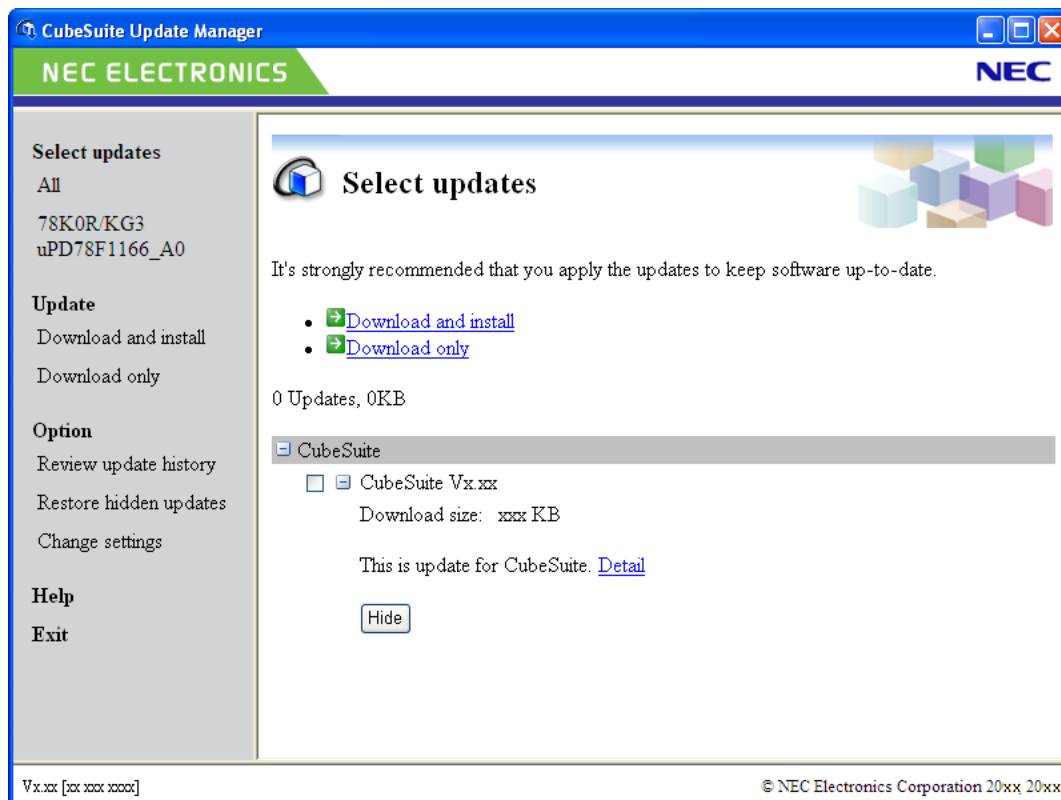
Figure 2-18. Checking for Updates Dialog Box



(2) Select update items.

A list of appropriate update items appears on the [Select updates page](#).

Figure 2-19. Select updates Page



Select the check boxes of updates and click [Download and install] or [Download only] in the update selection area.

- When [Download and install] is clicked: (3)
- When [Download only] is clicked: (5)

Remarks 1. In the menu area, select a microcontroller to only display update items for the selected microcontroller.

If you launched the [CubeSuite Update Manager window](#) by selecting [Check for Updates...] from the [Help] menu of CubeSuite, then update items for the microcontroller of the project currently open in CubeSuite is automatically selected.

2. If Update Manager failed to obtain the information, then the [Error page](#) appears.

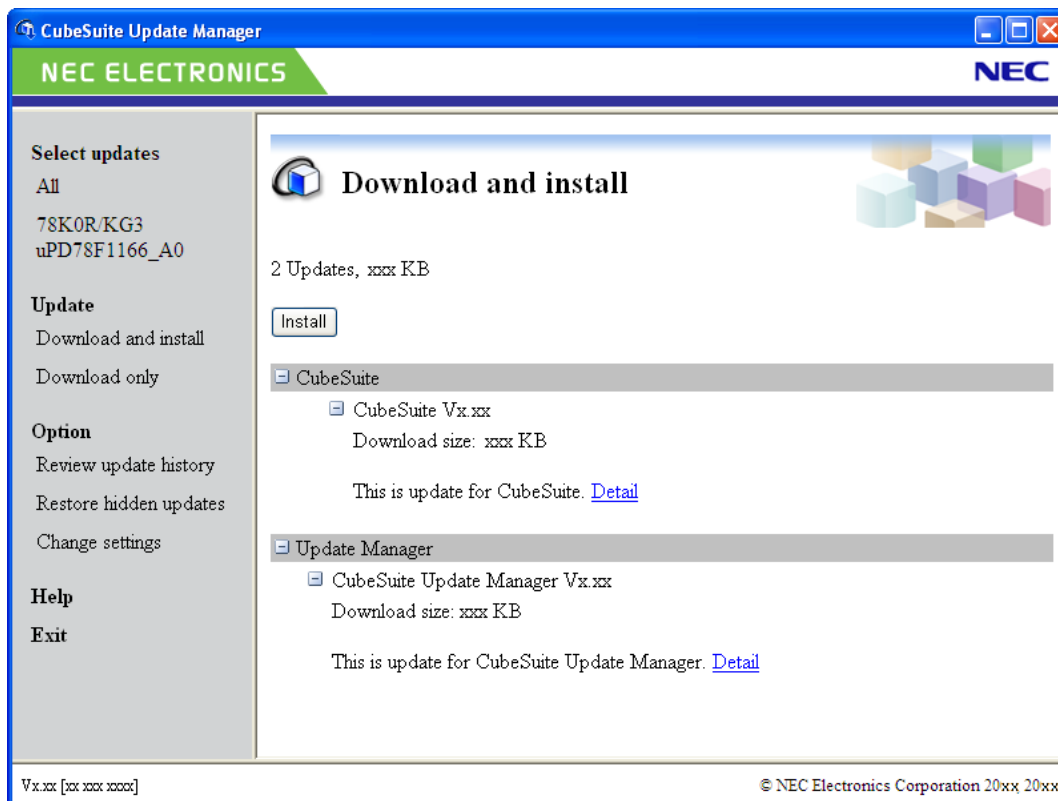
(3) Download and install.

Update items selected on the [Select updates page](#) appears on the [Download and install page](#).

Remark If there are the updates that must be installed together with the updates selected on the [Select updates page](#), a Message dialog box appears.

Click the [Yes] button on the Message dialog box to select the required updates.

Figure 2-20. Download and install Page



Check the information, and click the [Install] button.

(4) Confirm the download and install progress.

Update items are downloaded, and then they are installed. A progress bar on the **Update in progress dialog box** displays the download and install progress.

The download and install progress of each tool appear in the [Update status] area.

Figure 2-21. Update in progress Dialog Box (Downloading)

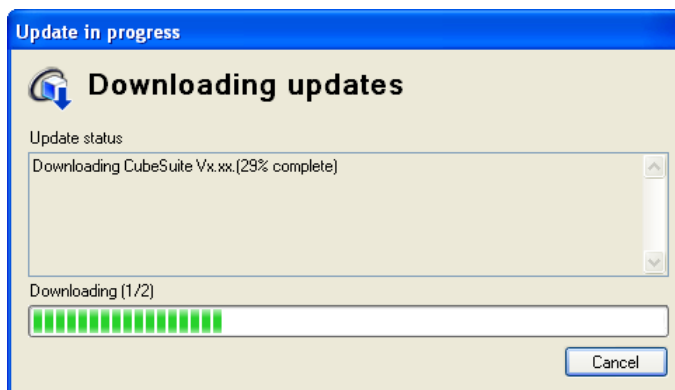
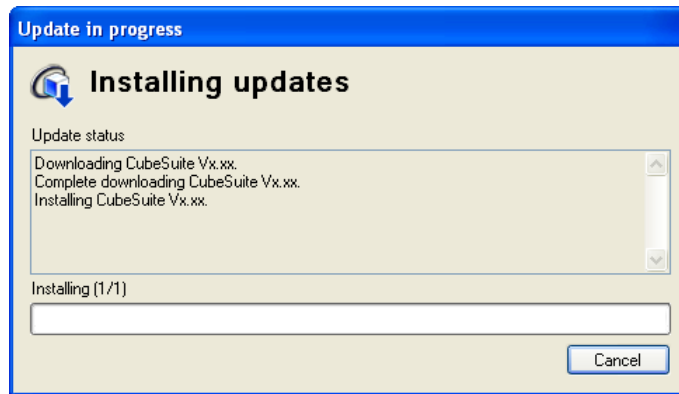


Figure 2-22. Update in progress Dialog Box (Installing)



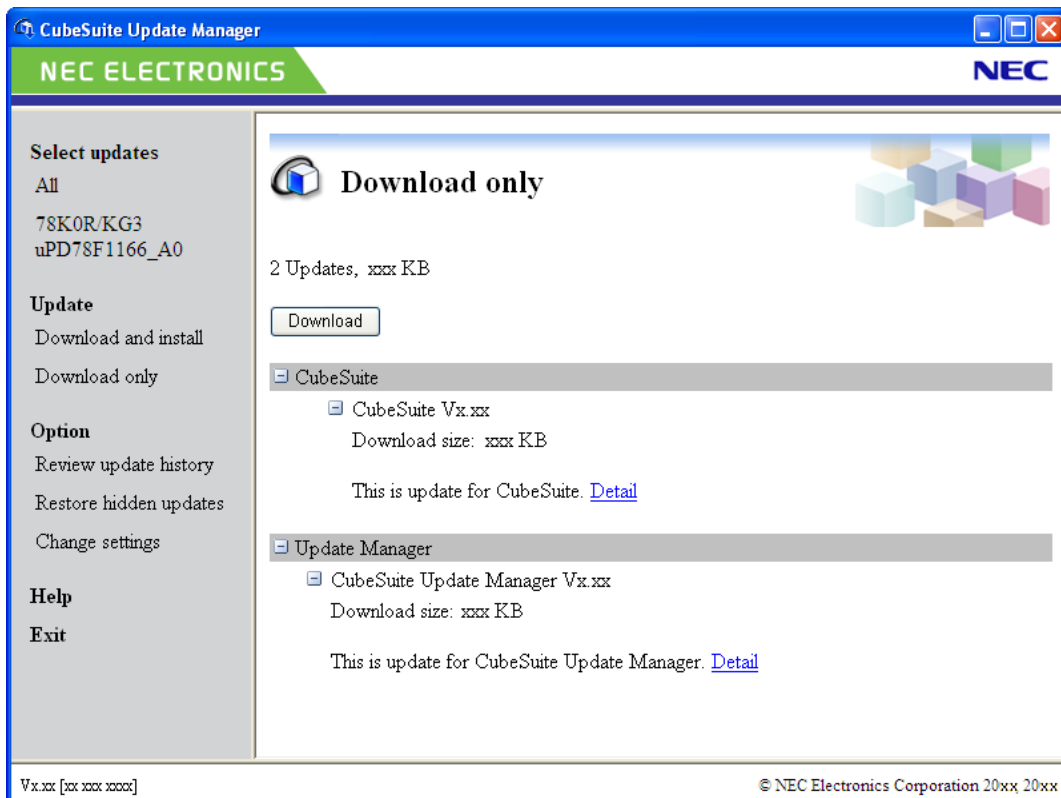
- Cautions**
1. You must have administrator privileges to perform installation.
 2. If a emulator USB driver is installed on Windows Vista or Windows 7, the warning dialog box may appear.

Go to (7).

(5) Download only.

Update items selected on the [Select updates page](#) appears on the [Download only page](#).

Figure 2-23. Download only Page



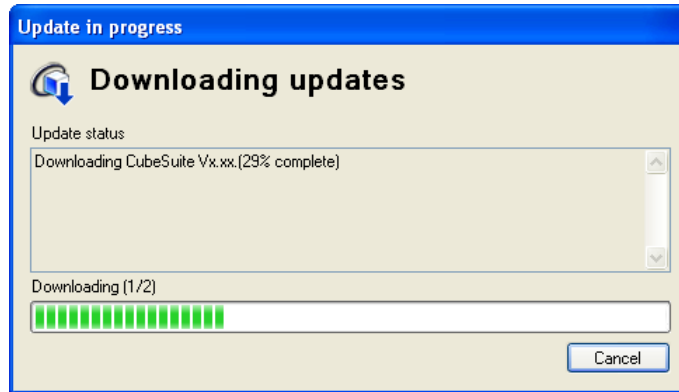
Check the information, and click the [Download] button.

(6) Confirm the download progress.

Update items are downloaded. A progress bar on the [Update in progress dialog box](#) displays the download progress.

The download progress of each tool appear in the [Update status] area.

Figure 2-24. Update in progress Dialog Box

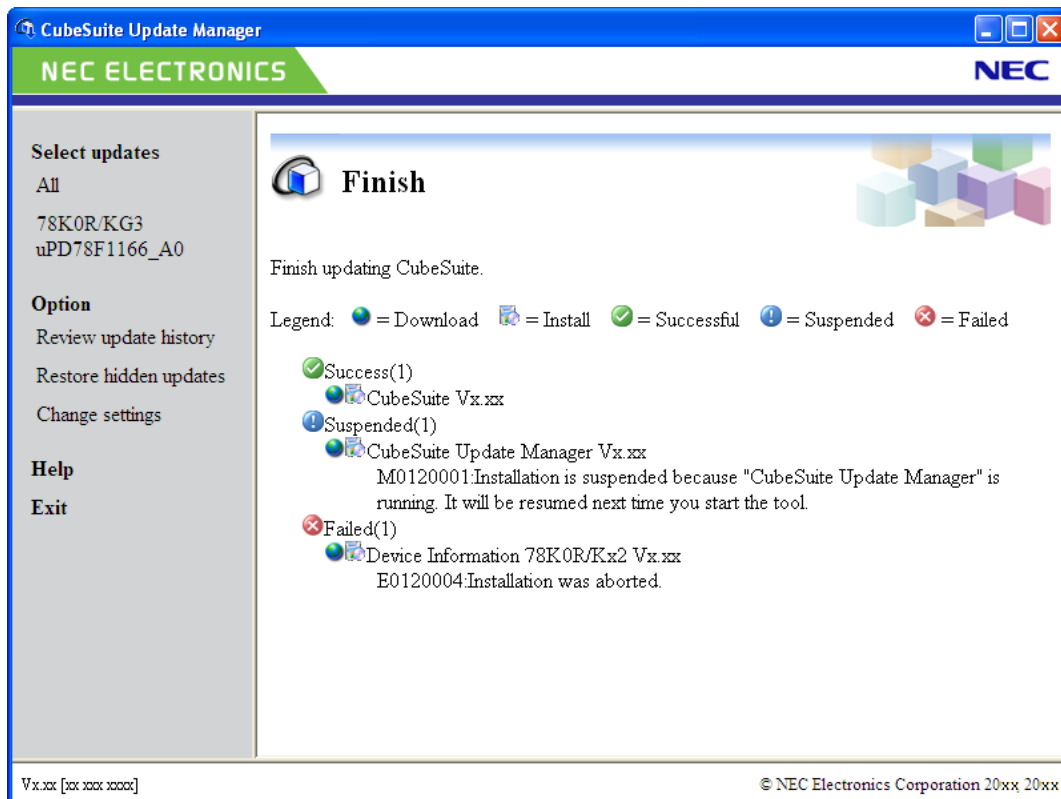


Remark See "2.4.5 [Displaying the update history](#)" for instructions on installing download update items later.

(7) Display results of download and installation.




When all update tasks are finished, the results of the update execution will appear on the [Finish page](#).

Figure 2-25. Finish Page





Click the [Exit] to complete updating.

Remarks 1. The results of the download and/or installation are displayed via.

	Displays number of successful updates and update titles.
	Displays number of canceled updates, update titles, and messages.
	Displays number of failed updates, update titles, and messages.

2. The icons below are displayed in the head of update titles.

	Updates are downloaded.
	Updates are installed.

2.4.2 The auto update feature

This section describes the auto update.

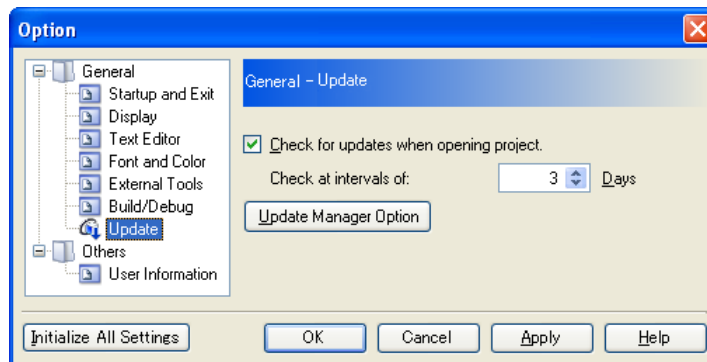
The auto update feature performs some of the tasks of the manual update feature automatically when a CubeSuite project is opened.


(1) Auto checking function

This function performs the check for updates when a CubeSuite project is opened.

This function is executed when: in the [Option dialog box](#), in the [\[General - Update\] category](#), the [\[Check for updates when opening project.\]](#) checkbox is selected, and the conditions set in [\[Check at intervals of\]](#) are met.

Figure 2-26. Option Dialog Box ([General - Update] Category)



Remarks 1. This function performs the check for updates in parallel with the CubeSuite process. While the check is ongoing, the  icon appears in the [Task Tray](#).

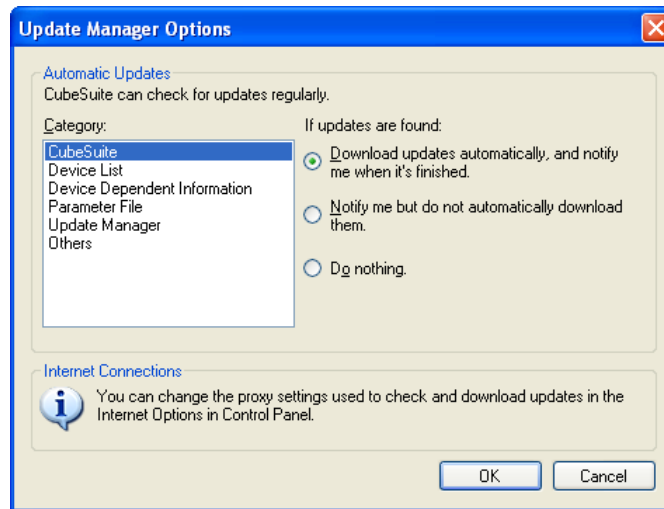
2. If the check does not find any updates, the auto checking function exits.


(2) Auto download function

The auto download function checks if the conditions for updating in (1) are met, and then downloads any updates that were found.



This function is executed for each category when: in the [Update Manager Options dialog box](#), [\[Download updates automatically, and notify me when it's finished.\]](#) is selected.

Figure 2-27. Update Manager Options Dialog Box



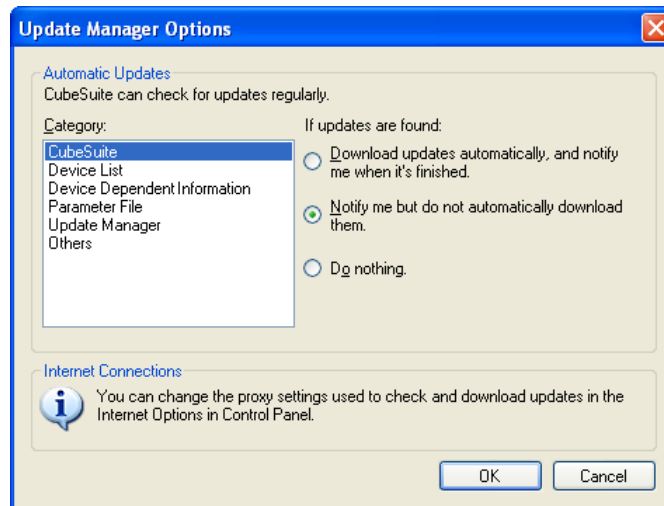
Remark During the download, the  icon appears in the [Task Tray](#).

(3) The update notification function

After downloading the target update in (2), the update notification feature displays an icon  notifying of the update in the [Task Tray](#). If any of the updates that were found for the first time during this auto check, the  icon appears.

This function is executed without performing the download when: in the [Update Manager Options dialog box](#), [Notify me but do not automatically download them.] is selected. This can be specified for each category.

Figure 2-28. Update Manager Options Dialog Box





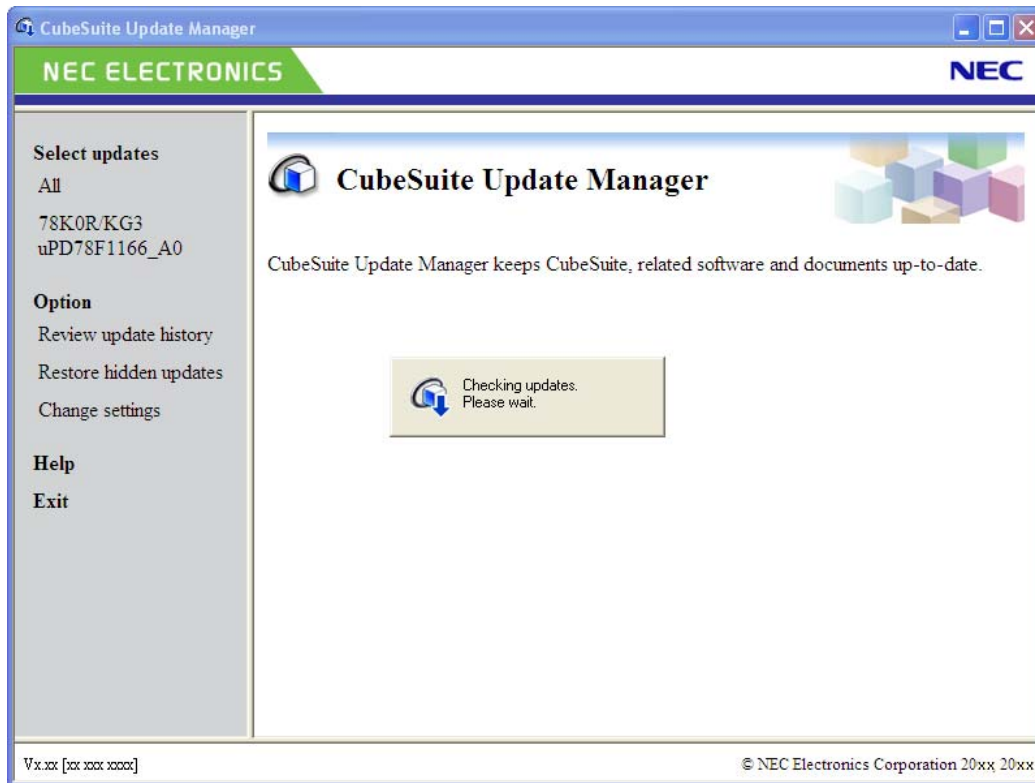
Remark Click the  or  icon to display the [CubeSuite Update Manager window](#). Perform the installation in accordance with “[2.4.1 The manual update feature](#)”, step (3).

Figure 2-29. CubeSuite Update Manager Window



2.4.3 Canceling an update

From the [Update in progress dialog box](#), click the [Cancel] button to cancel the update. If the update was canceled, then the [Finish page](#) will show that the update was canceled.

When performing an update, it is not possible to update currently running related tools or open files. For this reason, if a related tool is running then the installation will be postponed, and the [Finish page](#) will indicate that the installation was suspended.

2.4.4 Resuming an update

When Update Manager or a tool related to CubeSuite supporting the update function is started, it checks whether there are suspended installations.

If there are suspended installations, then if a related tool is running, a Message dialog box appears. After closing the tool, click the [Retry] button.

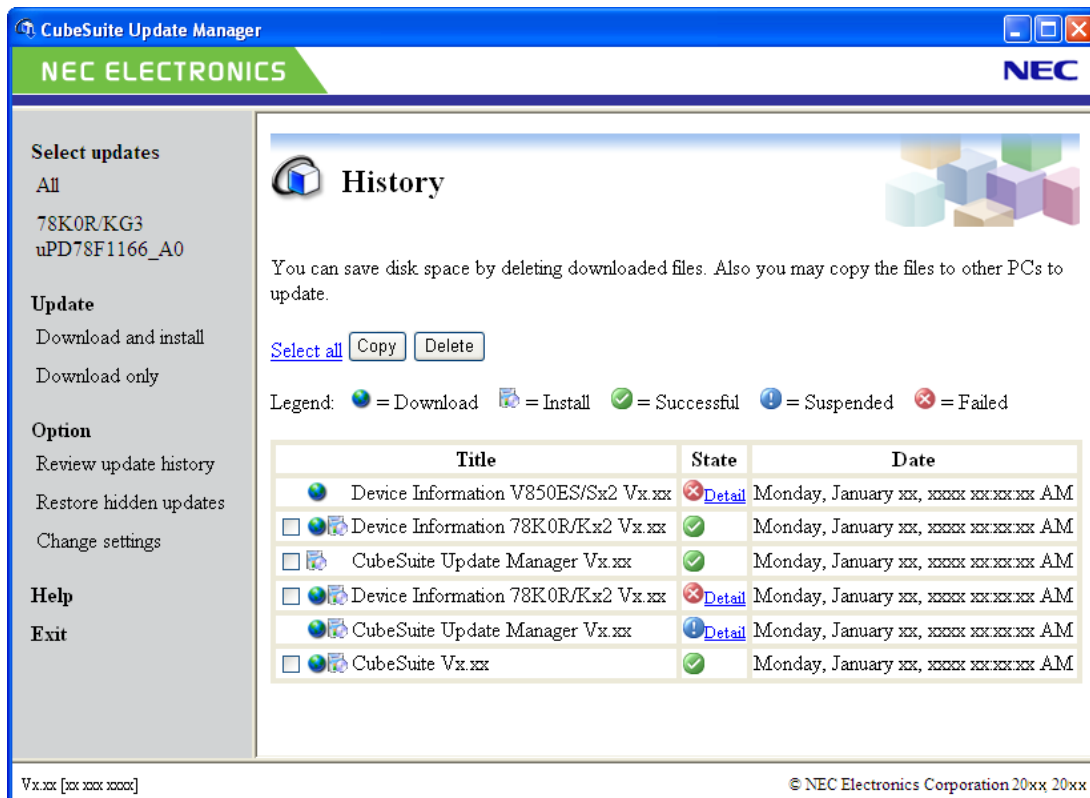
If a related tool is not running, a Message dialog box appears. Click the [Yes] button. Update Manager will perform the update. After the update is performed, the [CubeSuite Update Manager window](#) closes, and the tool that was shut down is restarted.

2.4.5 Displaying the update history

In the menu area of the CubeSuite Update Manager window, select [Review update history]. The History page appears.

From this page, you can install download files (including uploading to another PC) or delete them.

Figure 2-30. History Page



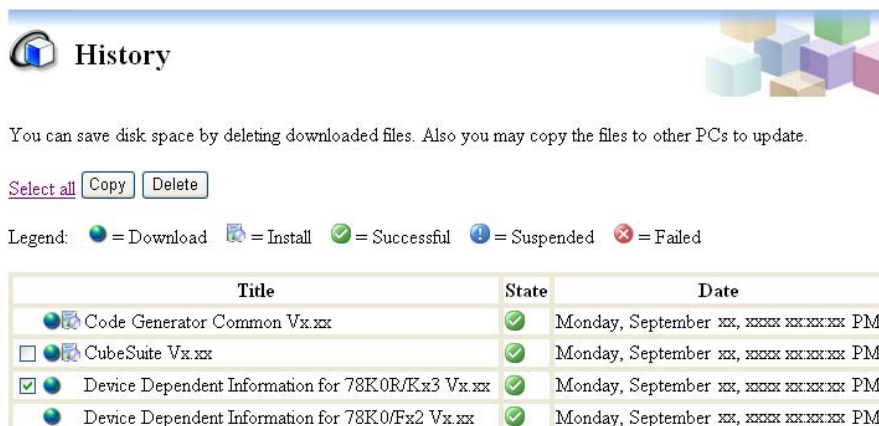
(1) Installing download files

This section describes how to install download files.

(a) Select download files.

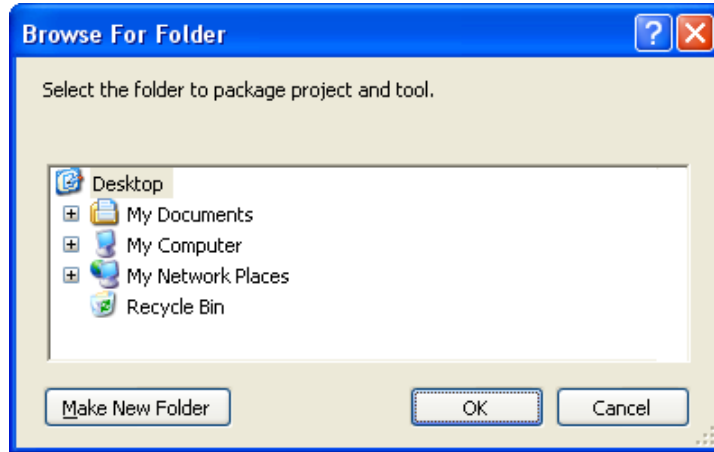
Select the check boxes of download files in the History page, and click the [Copy] button.

Figure 2-31. History Page



- (b) **Select a copy destination folder.**
The [Browse For Folder](#) dialog box appears.

Figure 2-32. Browse For Folder Dialog Box



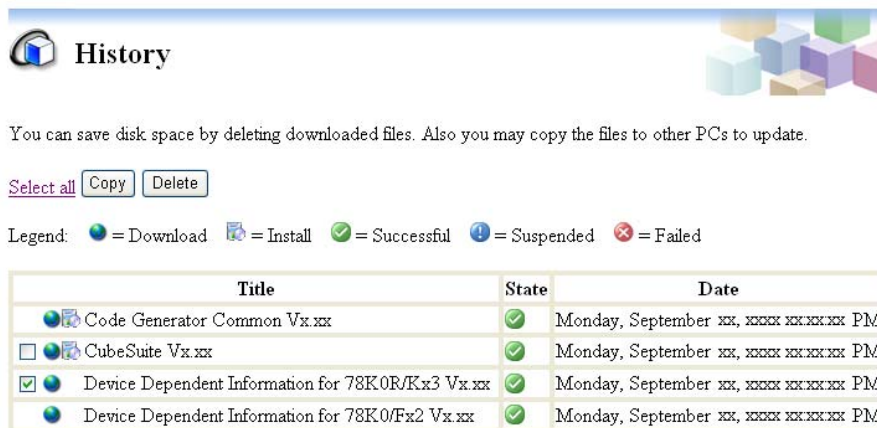
Select a copy destination folder, and click the [OK] button.

- (c) **Confirm the copy destination folder and install.**
The download files are copied to the specified folder.
Open the folder in Explorer, and install from the download files.

(2) Deleting download files

- (a) **Select download files.**
Select the check boxes of download files in the [History](#) page, and click the [Delete] button.

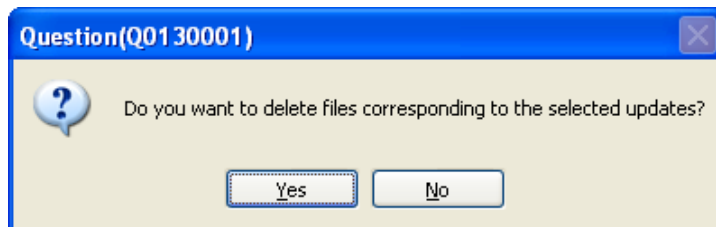
Figure 2-33. History Page



(a) Confirm whether you want to delete download files and delete them.

A Message dialog box asks whether you want to delete download files.

Figure 2-34. Message Dialog Box



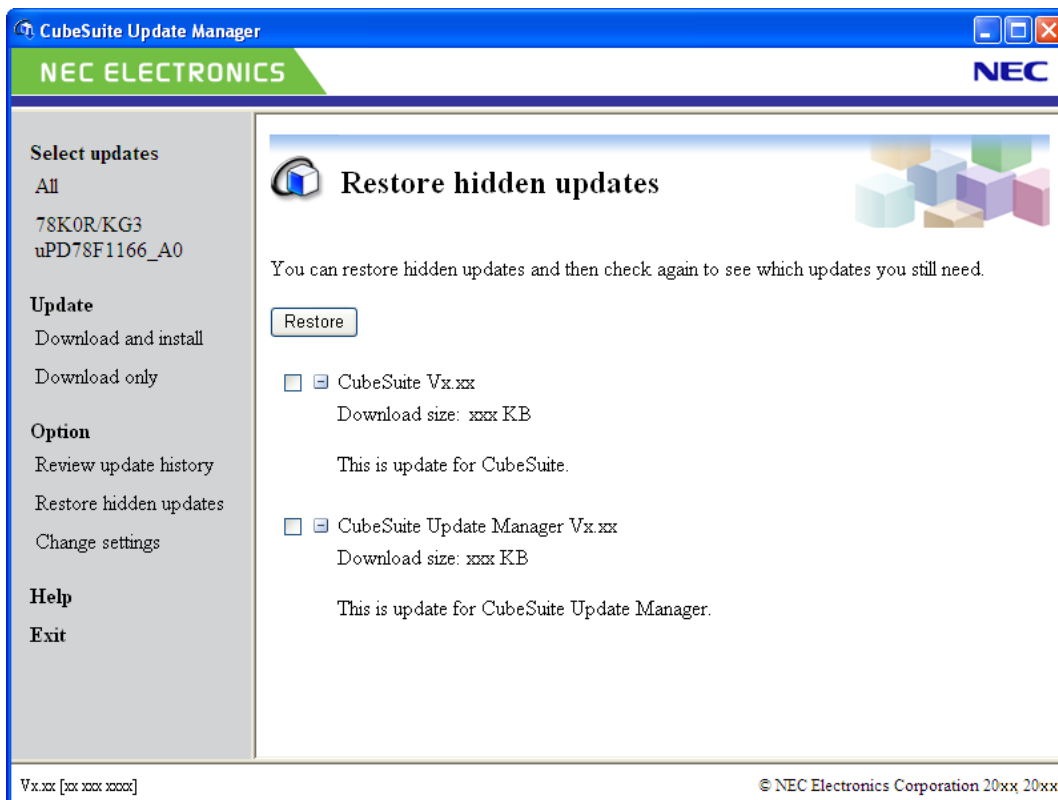
If you click the [Yes] button, the download files are deleted.

2.4.6 Restoring hidden updates

In the menu area of the [CubeSuite Update Manager window](#), select [Restore hidden updates] to display the [Restore hidden updates page](#).

From this page, you can display the updates that were hidden on the [Select updates page](#).

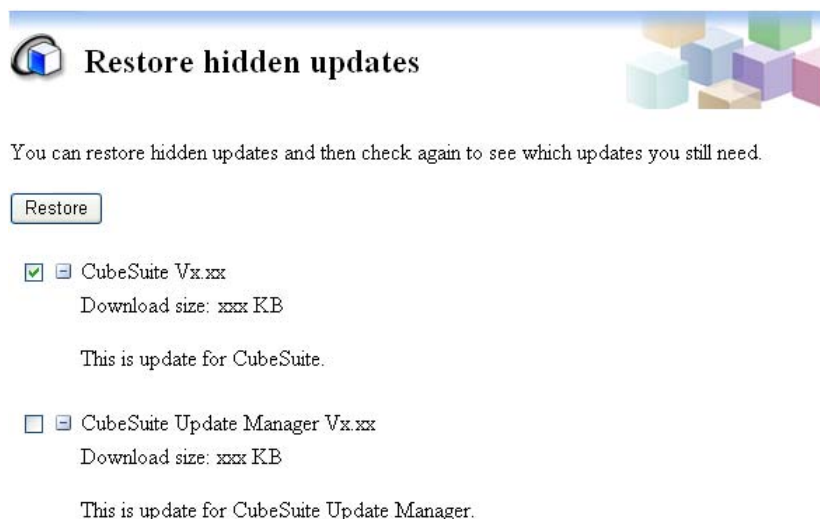
Figure 2-35. Restore hidden updates Page



(a) Select updates.

Select the check boxes of updates in the [Restore hidden updates page](#), and click the [Restore] button.

Figure 2-36. Restore hidden updates Page



(b) Confirm that hidden updates are restored.

Confirm that the selected updates become invisible on the [Restore hidden updates page](#) and become visible on the [Select updates page](#).

Figure 2-37. Restore hidden updates Page

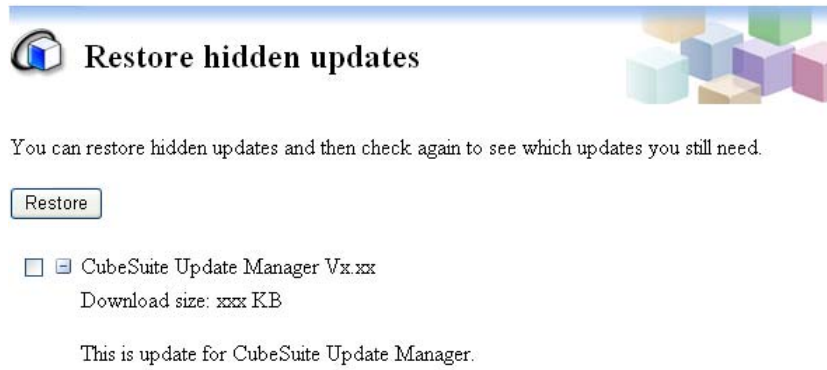
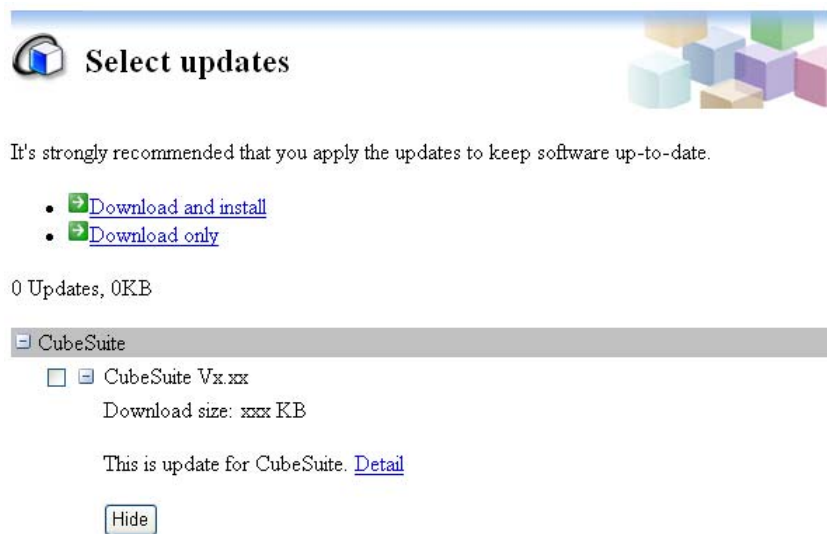


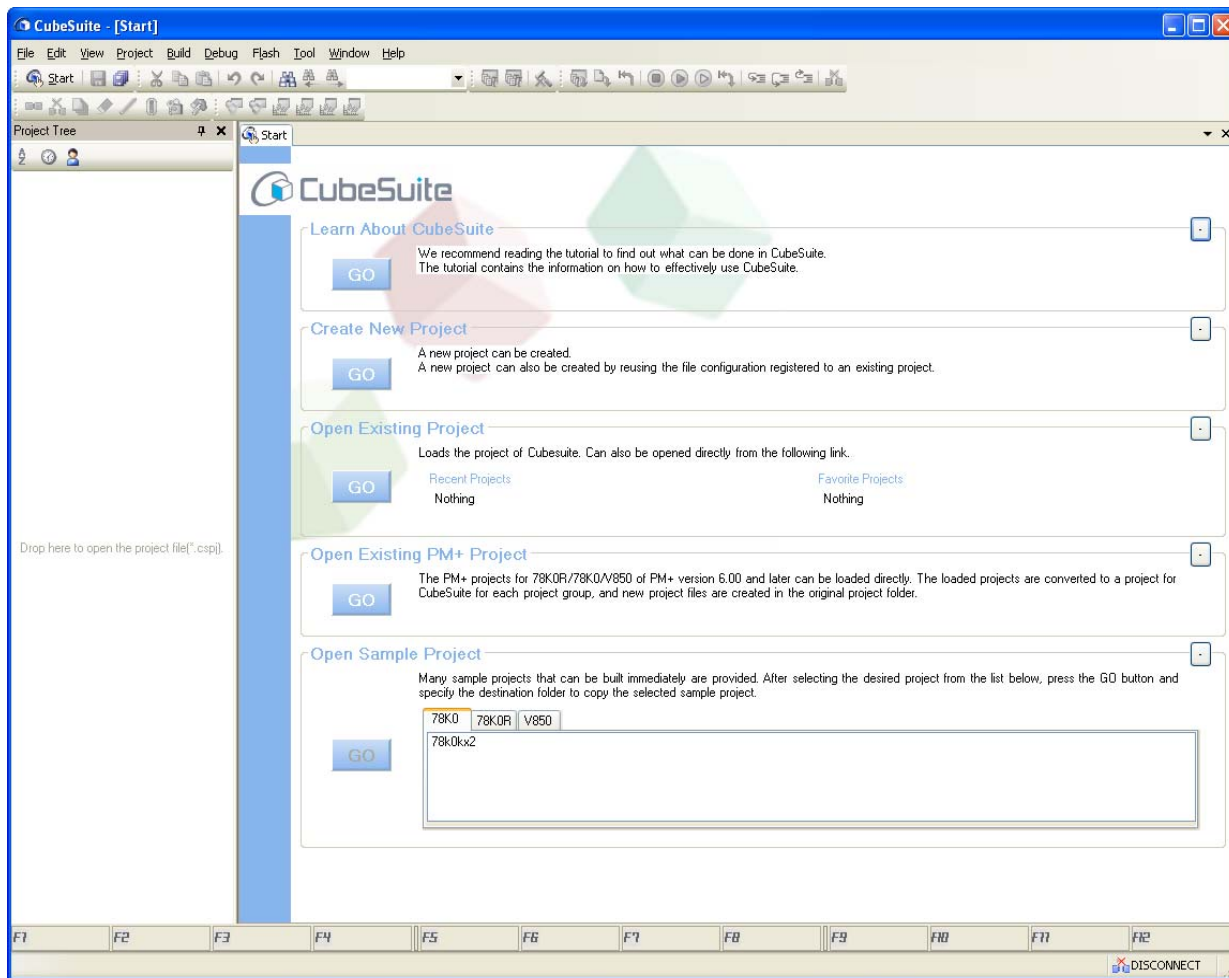
Figure 2-38. Select updates Page



2.5 Start CubeSuite

Select Windows [Start] menu >> [Programs] >> [NEC Electronics CubeSuite] >> [CubeSuite], CubeSuite is started.

Figure 2-39. Main Window (Start Panel)



2.6 Create a Project

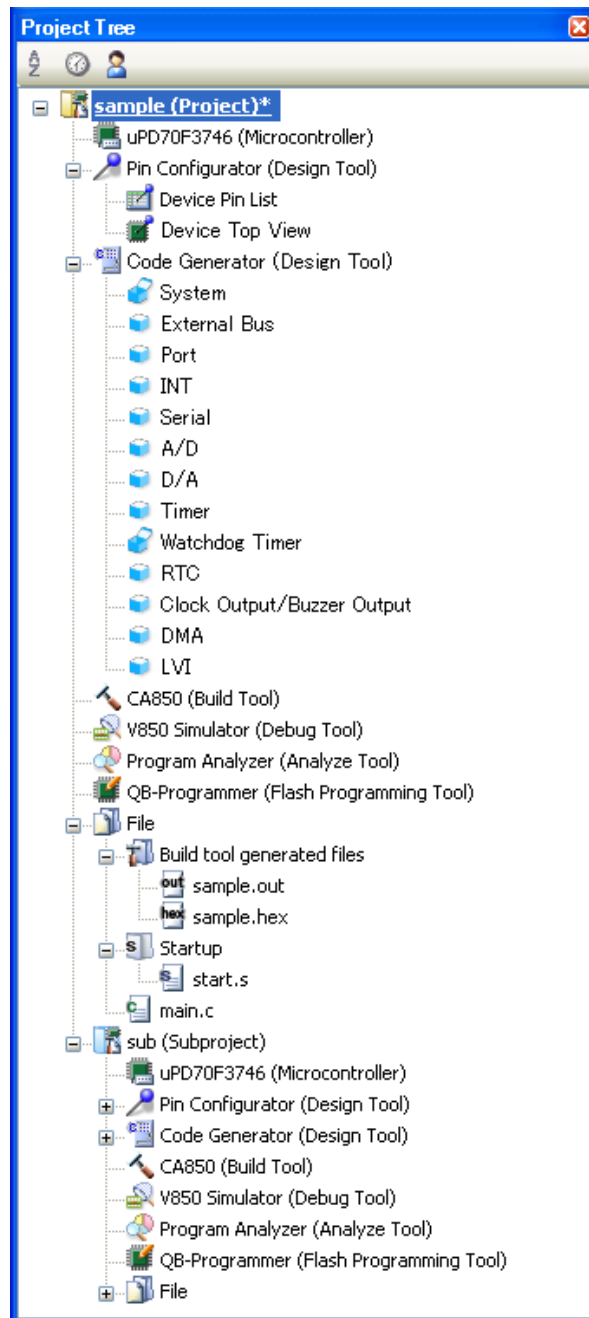
A project is managed by CubeSuite as the unit for application system development.

CubeSuite saves settings information used in the project such as the microcontroller, build tool, and source files, to the project file (*.cspj) and references it.

(1) Project tree organization and detailed settings

The project's settings are made on the [Project Tree](#) panel.

Figure 2-40. Project Tree Panel



On the project tree, project components are displayed as the nodes below in a tree view.

If you select each component (node or file), its detailed information (properties) is displayed in the [Property panel](#) and you can change its settings.

You can also make project settings from the context menu for each component (node or file).

Node	Explanation
<i>Project name</i> (Project) (Hereafter referred to as "Project node")	The project's name.
<i>Microcontroller name</i> (Microcontroller) (Hereafter referred to as "Microcontroller node")	The microcontroller used in the project.
<i>Design Tool name</i> (Design Tool) (Hereafter referred to as "Design Tool node")	The design tool (pin configurator, code generator, etc.) used. Note that Code Generator (Design Tool) node is not shown when the project type is a debug-dedicated project ^{Note} .
<i>Build tool name</i> (Build tool) (Hereafter referred to as "Build Tool node")	The build tool (compiler, assembler, etc.) used. When the project type is a debug-dedicated project, ^{Note} "None" is shown as <i>build tool name</i> .
<i>Debug tool name</i> (Debug Tool) (Hereafter referred to as "Debug Tool node")	The debug tool (in-circuit emulator, simulator, etc.) used.
Program Analyzer (Analyze Tool) (Hereafter referred to as "Analyze Tool node")	The analyze tool used. Note that this node is not shown when the project type is a debug-dedicated project ^{Note} .
QB-Programmer (Flash Programming Tool) (Hereafter referred to as "Flash Programming Tool node")	The flash programming tool used.
File (Hereafter referred to as "File node")	Files registered to the project are displayed directly below the File node.
Download files (Hereafter referred to as "Download files node")	This is a node for adding download files to the project. Note that this node is shown only when the project type is a debug-dedicated project ^{Note} .
Build tool generated files (Hereafter referred to as "Build tool generated files node")	This node is created during a build. Files created by the build tools are displayed directly below the node (except for object files). Note that this node is not shown when the project type is a debug-dedicated project ^{Note} .
Startup (Hereafter referred to as "Startup node")	This is a node for adding other than standard startup files to the project. This node is always shown under the File node. Note that this node is not shown when the project type is a debug-dedicated project ^{Note} .
<i>Category name</i> (Hereafter referred to as "Category node")	These user-defined categories are used to classify files into modules.
<i>Subproject name</i> (Subproject) (Hereafter referred to as "Subproject node")	Subprojects added to the project. For subprojects, see "(2) Projects and subprojects".

Note See "APPENDIX E USING AN EXTERNAL BUILD TOOL" for details on a debug-dedicated project.

- Remarks 1.** Only the tools corresponding to the microcontroller in use are shown.
- 2.** When more than one components are selected, only the tab that is common to all the components is displayed.
When more than one files are selected and their common properties are different, that field is left blank.

(2) Projects and subprojects

Projects can have subprojects added to the level beneath them.

The subproject's settings information is saved to a subproject file (*.cssp).

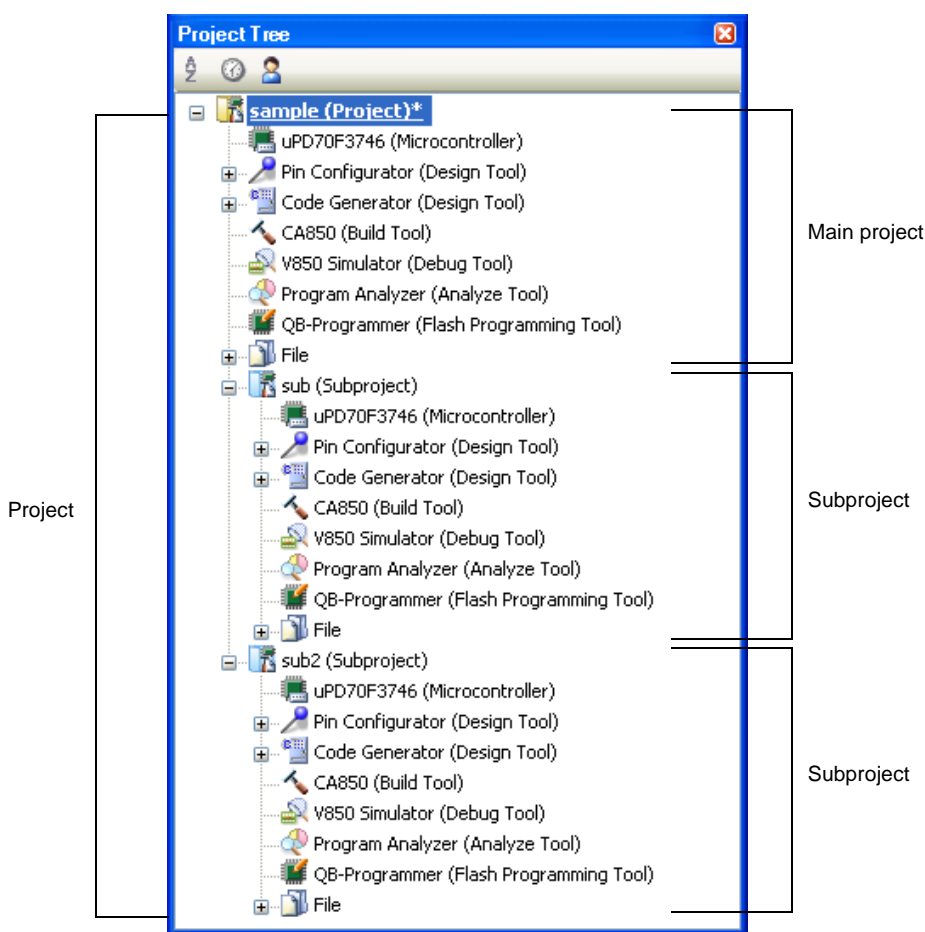
Subprojects, for example, are used in the following ways.

- When also creating a project to create library files used in the project, create a project to create library files as a subproject.
- When developing the same application system for different microcontrollers, create the projects that differ for the microcontroller as subprojects.

When subprojects are added, this manual call the project the "main project", as opposed to its subprojects.

"Project" is a common term for the main project and subprojects.

Figure 2-41. Project Tree Panel (When Subprojects Added)




In addition, project settings and the settings for subprojects added to a project are independent and have no effect on each other. When making the same settings between the main project and subprojects, or between differing subprojects, select multiple nodes to set on the project tree, and make the settings with the [Property panel](#).

Caution A subproject cannot be added to another subproject.

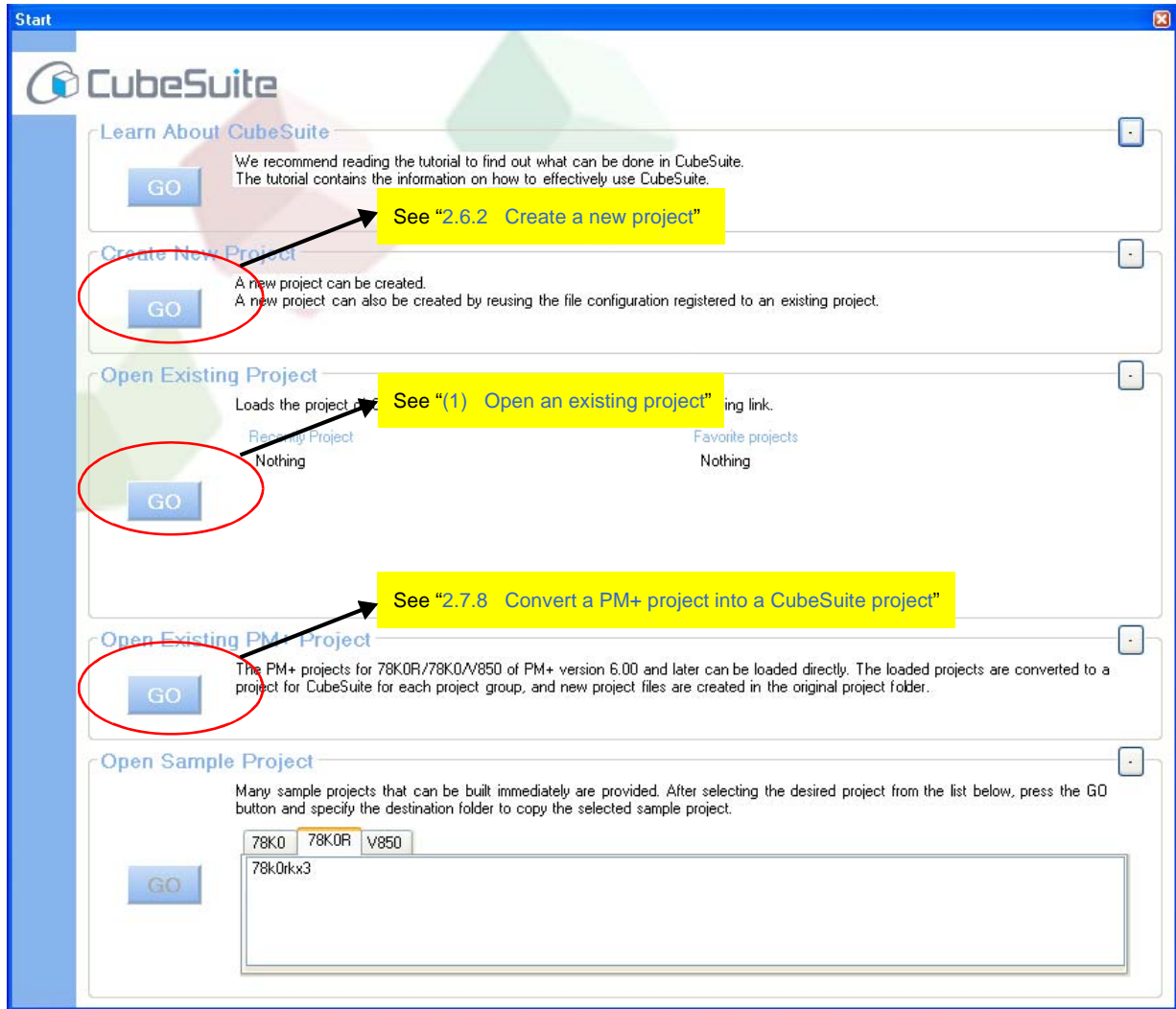
Remark See "2.6.3 Add a new subproject" and "2.7.2 Add an existing subproject", for how to add subprojects.

2.6.1 Start a project

On the tool bar, click  Start to open the Start panel. You can click the buttons on the panel to create a new project or open an existing one.

Note that the Start panel opens automatically the first time that CubeSuite starts.

Figure 2-42. Start Panel

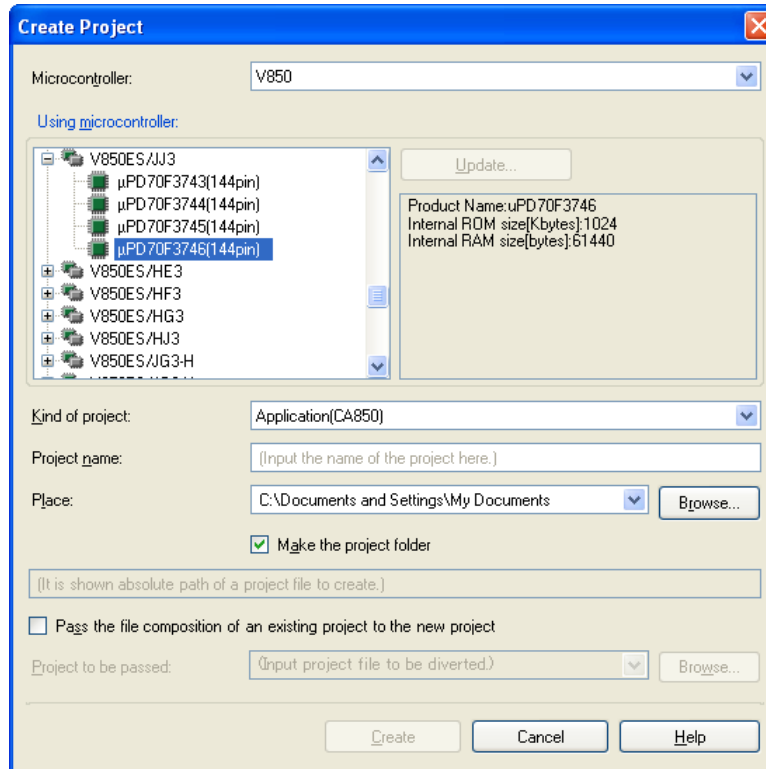


2.6.2 Create a new project

Create a new project.

From the [Project] menu, select [Create New Project...], the [Create Project dialog box](#) will open.

Figure 2-43. Create Project Dialog Box (When First Started)



Set the items in the order below.

(1) Select the microcontroller type

Select the microcontroller type to use in the project on [Microcontroller].

You can select the item below.

- V850
- 78K0R
- 78K0

(2) Select the microcontroller

Select the microcontroller to use in the project on the [Using microcontroller] area.

If you register the license of CubeSuite, the [Update...] button becomes enabled.

If your microcontroller is not in the [Using microcontroller] area, click the [Update...] button.

You can open the [CubeSuite Update Manager window](#), and check for microcontroller information updates via the network.

(3) Select the project type

Select the project type to create on [Kind of project].

You can select the item below.

- Application(CA850/CX/CA78K0/CA78K0R)

Select this to generate the ROMization module file [V850], load module file, and hex file, by using the build tool provided by CubeSuite.

- Library(CA850/CX/CA78K0/CA78K0R)

Select this to generate a library file for a user library, by using the build tool provided by CubeSuite.

- Debug Only

Select this to debug a load module file or hex file generated with a build tool other than the one provided by CubeSuite.

For details on how to create and use this project type (debug-dedicated project), see "[APPENDIX E USING AN EXTERNAL BUILD TOOL](#)".

(4) Specify the project name and location to create the project file

Specify the name of the project and the location to create the project file in [Project name] and [Place].

If you don't create a folder with the project name under the specified location, clear the [Make the project folder] check box.

Caution When directly entering the location to create the project file, enter it as an absolute path.

(5) Specify the reuse of the file structure of an existing project

When creating a project that reuses the file structure of an existing project, check [Pass the file composition of an existing project to the new project] and specify the location of the project filename to reuse in [Project to be passed].

Caution You cannot specify a PM+ project file.

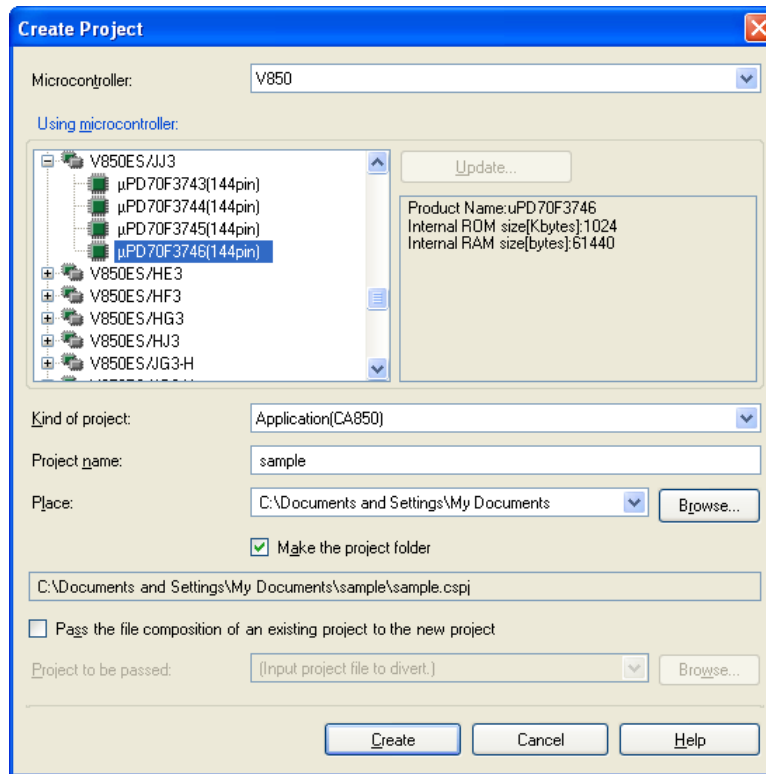
If you wish to copy an existing PM+ project, open the project in CubeSuite, then save it as a CubeSuite project (see "[2.7.8 Convert a PM+ project into a CubeSuite project](#)" for details). Next, specify the saved project file in this area.

Remarks 1. When the version of the build tool used in the source project is different from the version of the build tool in the project to be created, it is automatically diverted (the case that "Debug Only" is specified with [Kind of project] is excluded).

2. You can create a project with CX as the build tool by reusing the file structure of a project with CA850 as the build tool (see "[2.7.7 Convert a CA850 project into a CX project](#)").

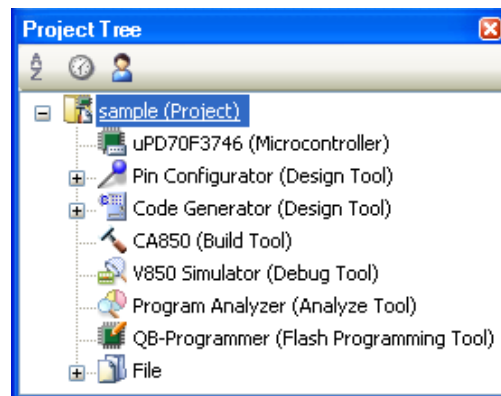
An image of the dialog box after setting the items is shown below.

Figure 2-44. Create Project Dialog Box (After Setting Items)



When you click the [Create] button, the project file is created in the location specified in (4) and the structure of the created project is displayed as a tree in the [Project Tree](#) panel.

Figure 2-45. Project Tree Panel (After Creating a New Project)



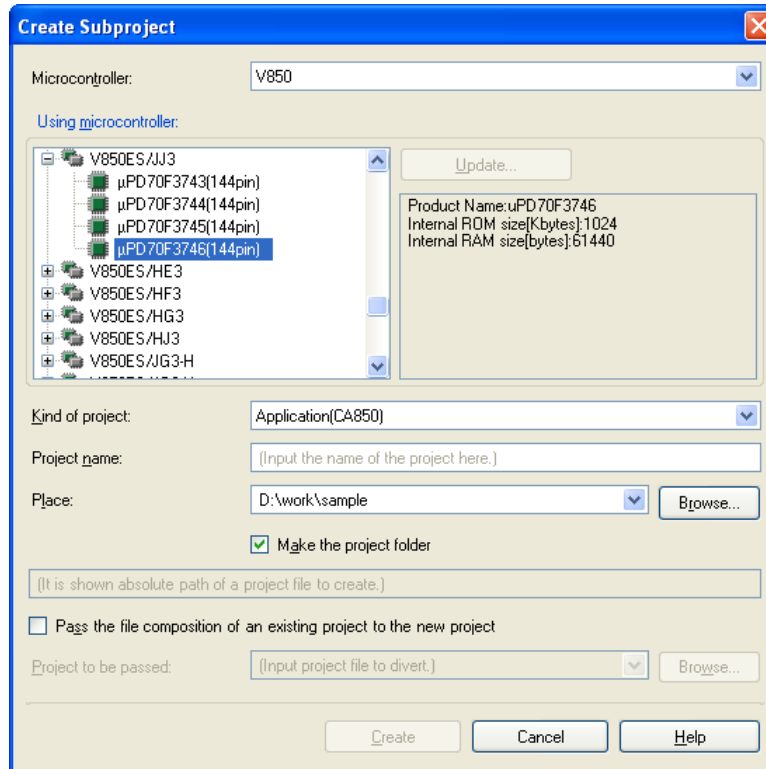
Remark After creating a project, you must add target files to the project in order to perform building or debugging. For details on how to add these files, see the following.

- When "Application" or "Library" is selected on [Kind of project]
 - > See the "CubeSuite Build"
- When "Debug Only" is selected on [Kind of project]
 - > See "E.3 Add a File to a Project"

2.6.3 Add a new subproject

Select the Project node on the project tree and if you select [Add] >> [Add New Subproject...] on the context menu, the [Create Project dialog box](#) will open.

Figure 2-46. Create Project Dialog Box (When Adding a New Subproject)

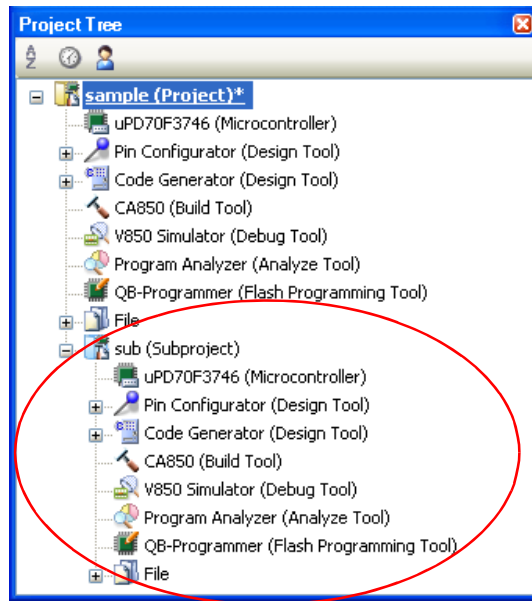


After setting each item on the dialog box, click the [Create] button (For more on the settings for each item, see ["2.6.2 Create a new project"](#)).

Remark You can create a subproject with CX as the build tool by reusing the file structure of a subproject with CA850 as the build tool (see ["2.7.7 Convert a CA850 project into a CX project"](#)).

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

Figure 2-47. Project Tree Panel (After Adding a Subproject)



2.7 Manipulate a Project

This section describes how to manipulate a project.

2.7.1 Open a project

Use the following method to open a project.

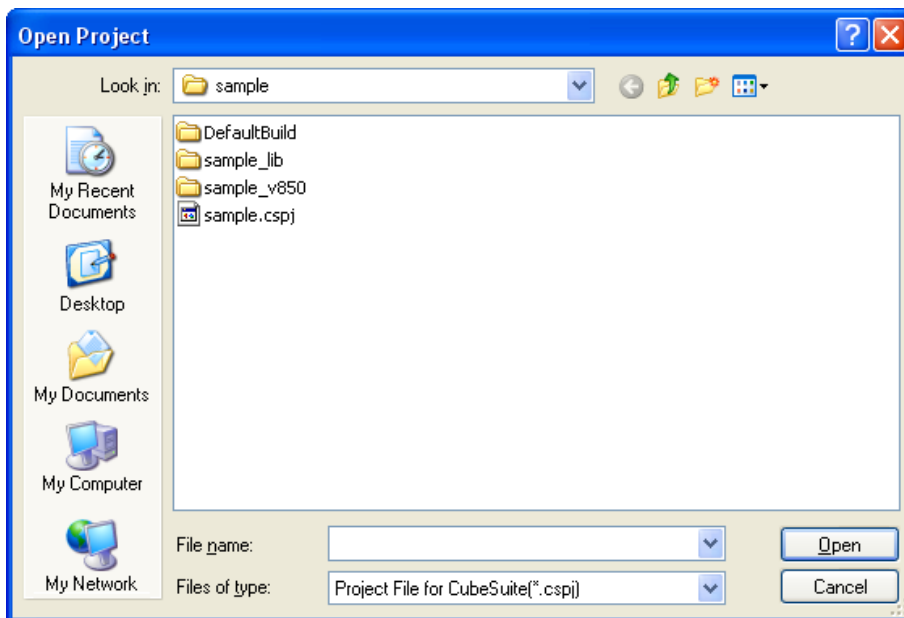
- [Open an existing project](#)
- [Open a recently used project](#)
- [Open a project from the favorites menu](#)

(1) Open an existing project

Existing projects are opened by specifying the project file.

From the [Project] menu, select [Open Project...], the [Open Project dialog box](#) will open.

Figure 2-48. Open Project Dialog Box



On the dialog box, specify the project file and click the [Open] button.

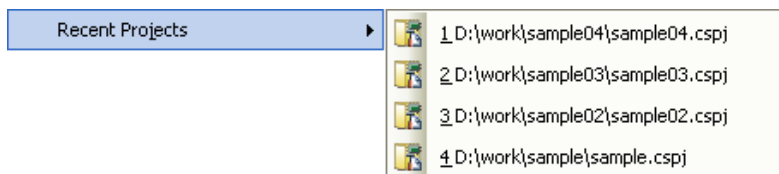
Remark When CubeSuite is not running, you can start CubeSuite and load a project by double-clicking on that project in Explorer.

(2) Open a recently used project

You can directly open the most recently used projects (from the most recent to the fourth most recent) from the menu.

From the [File] menu, select [Recent Projects], the path of the recently used projects will display in a cascading menu in order from most recent to fourth most recent. Select the project you wish to open.

Figure 2-49. [Recent Projects] Item

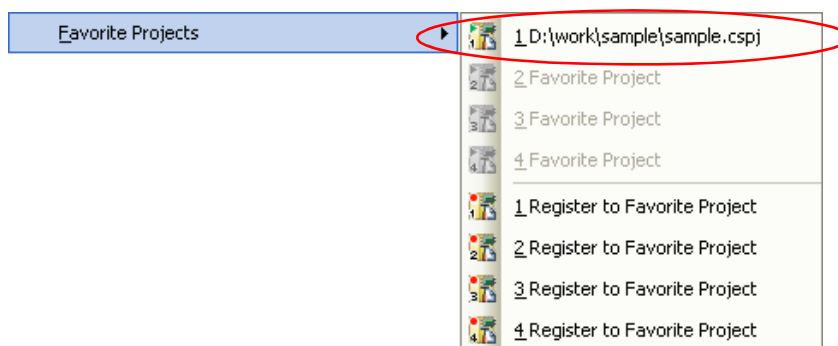


(3) Open a project from the favorites menu

Open a project registered on the favorites menu.

From the [Project] menu, select [Favorites Projects], the path of the projects registered on the favorites menu is displayed as a cascading menu. Select the project you wish to open.

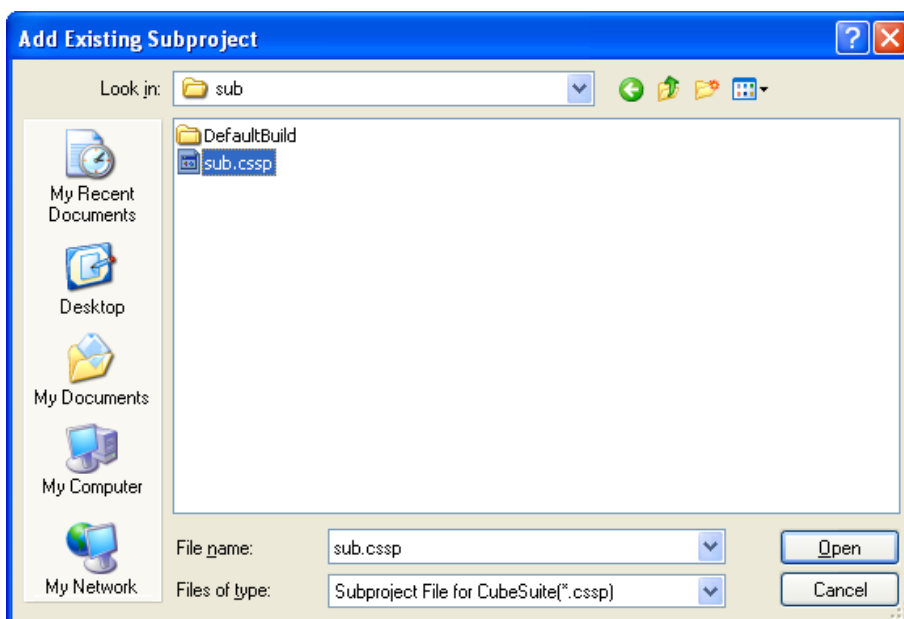
Figure 2-50. [Favorites Projects] Item



2.7.2 Add an existing subproject

Select the Project node on the project tree and if you select [Add] >> [Add Subproject...] on the context menu, the [Add Existing Subproject dialog box](#) will open.

Figure 2-51. Add Existing Subproject Dialog Box



On the dialog box, specify the subproject file of the subproject to add and click the [Open] button.

2.7.3 Project is added to the favorites menu

You can add the currently open project to the menu as a "favorite project" (up to four projects).

From the [Project] menu, select [Favorites Projects] >> [1 - 4 Register Favorites Project], the path of the currently open project is registered under the [Project] menu >> [Favorites Projects].

Figure 2-52. [Register to Favorites Project] Item

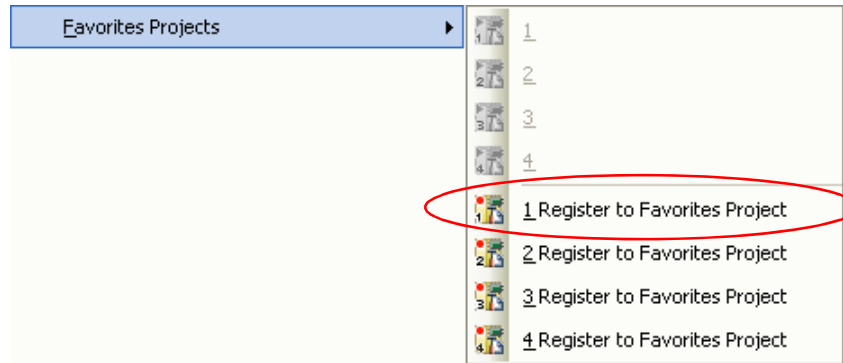
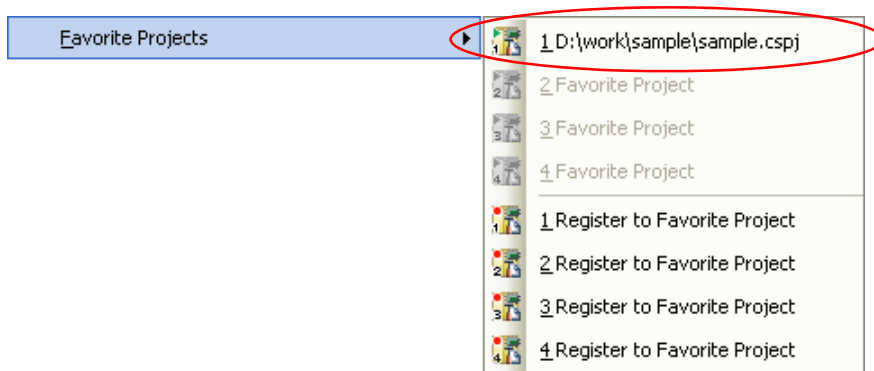


Figure 2-53. [Favorites Projects] Item (After Registering a Project)

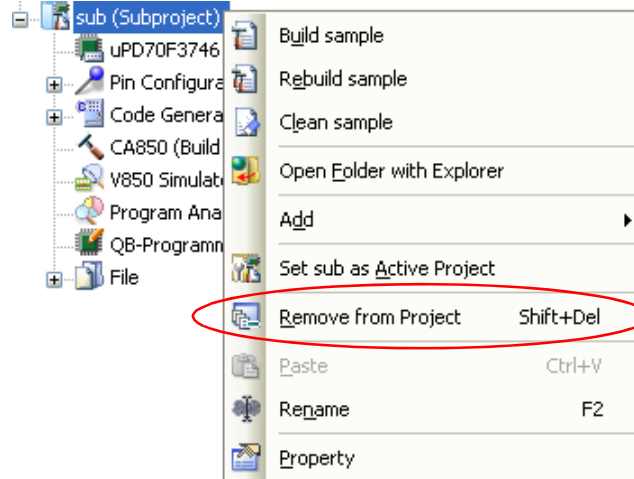


2.7.4 Remove a subproject from the project

To remove a subproject registered to a project from that project, select the Subproject node on the project tree, and select [Remove from Project] on the context menu.

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

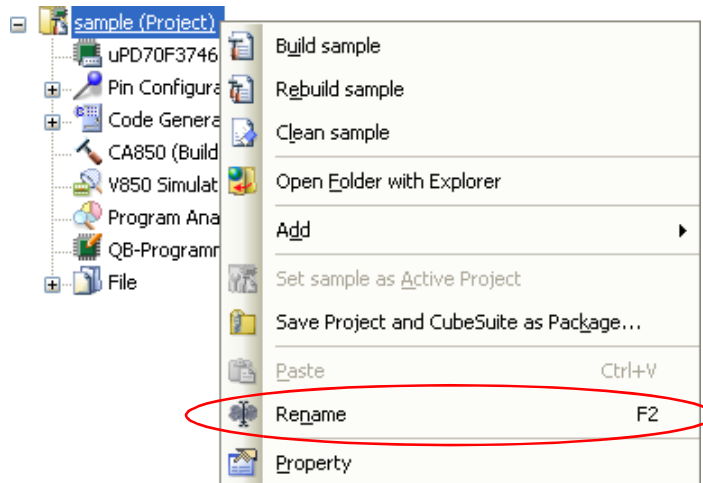
Figure 2-54. [Remove from Project] Item



2.7.5 Change the project name

You can change the name of the project (main project or subproject) on the project tree. Select the Project node or Subproject node and select [Rename] on the context menu.

Figure 2-55. [Rename] Item (For a Project)



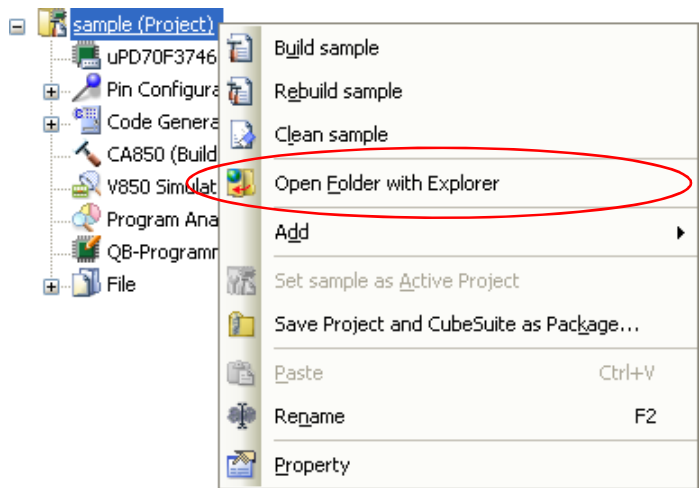
Remark After changing the project name, when you save the project, the actual name of the project file is also changed.

2.7.6 Open a project folder in Explorer

You can open the folder where the project file for a project (main project or subproject) is saved from the project tree in Explorer.

Select the Project node or Subproject node and select [Open Folder with Explorer] on the context menu.

Figure 2-56. [Open Folder with Explorer] Item (For a Project)



Remark When you select [Open Folder with Explorer] from a file's context menu, the folder the file is saved in is opened in Explorer.

2.7.7 Convert a CA850 project into a CX project

You can convert a CubeSuite project with CA850 as the build tool into a project with CX as the build tool by creating the project reusing the file structure of a project.

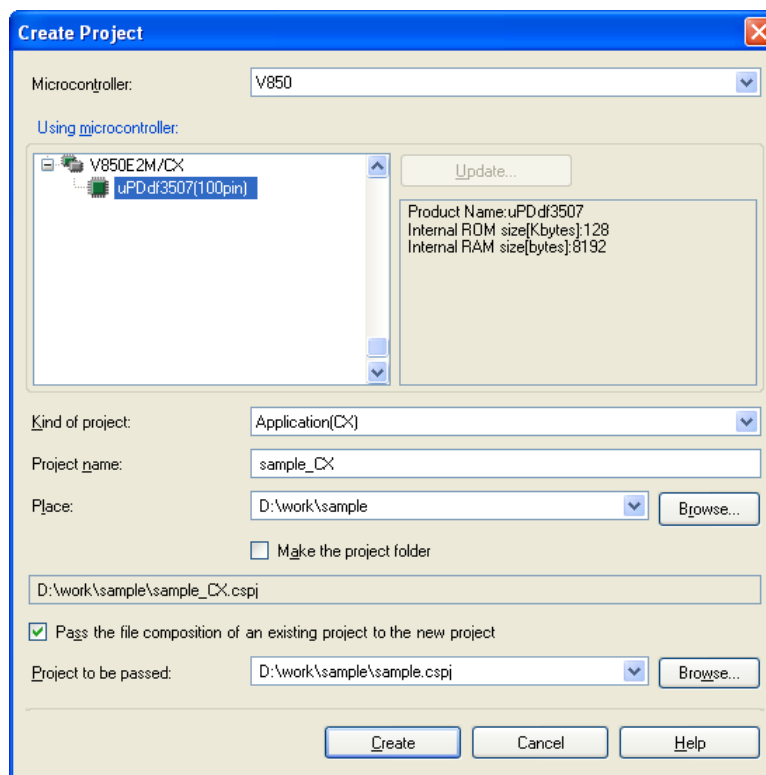
When creating the project, it is not necessary to code build-tool dependent source, because the source files are converted for the CX.

The properties of the build tool are also maintained, and converted for use with the CX.

Caution If you wish to reuse an existing PM+ project, open the project in CubeSuite, then save it as a CubeSuite project (see "2.7.8 Convert a PM+ project into a CubeSuite project").

First, from the [Project] menu, select [Create New Project...], the [Create Project dialog box](#) will open.

Figure 2-57. Create Project Dialog Box



Set the items in the order below and click the [Create] button.

(1) Select the microcontroller type

Select "V850" on [Microcontroller].

(2) Select the microcontroller

Select the microcontroller to use in the project on [Using microcontroller].

(3) Select the project type

Under [Kind of project], select "Application(CX)" or "Library(CX)", in accordance with the source project.

(4) Specify the project name and location to create the project file

Specify the name of the project and the location to create the project file in [Project name] and [Place].
 If you don't create a folder with the project name under the specified location, clear the [Make the project folder] check box.

(5) Specify the reuse of a CA850 project

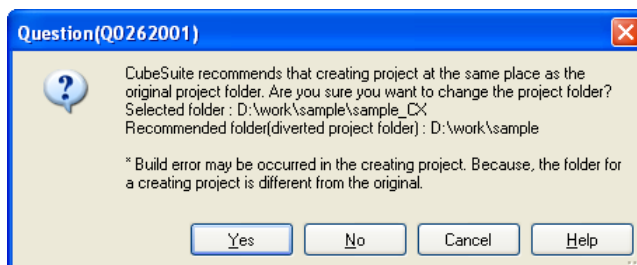
Check [Pass the file composition of an existing project to the new project] and specify the location of the project filename to reuse in [Project to be passed].

Caution We recommend using the same project folder as the source project, in order to avoid breaking the relationship between the project folder and source files.

If the project folder is different from that of the source project, the following message dialog box will appear if you click the [Create] button.

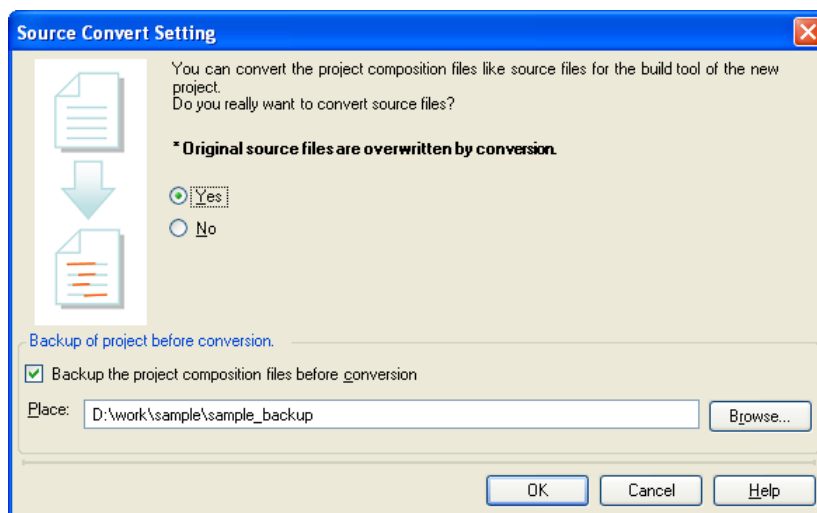
To continue the process, click the [Yes] button.

Figure 2-58. Message Dialog Box



Click the [Create] button. The Source Convert Setting dialog box [CX] appears.

Figure 2-59. Source Convert Setting Dialog Box



Select [Yes] to perform conversion on the source files.

To save a backup of the entire project (including source files), select the [Backup the project composition files before conversion] check box, and specify a location in which to save the backup.

Click the [OK] button to convert the source files and create the CX project.

- Remark** The result that a CX project is created by reusing a CA850 project (the versions of the IDE and compiler package, and conversion information of options) is output to a file as project divert information.
- The project divert information file name is "ProjectDivertInformation*n*.txt" (*n* = 2 to 100).
n is not added normally. It is added if the file to be created already exist.
 - The project divert information file is output for each created project (subproject).
 - The project divert information file is output to the project folder of the project (subproject).
 - The project divert information file is added to the File node of the project (subproject) on the [Project Tree panel](#).

The output format of the project divert information file is shown below.

```
(1)Time and date on which a project was created

(2)<CubeSuite IDE(Integrated Development Environment Framework) version>
    CubeSuite IDE: Version of IDE of CA850 project -> Version of IDE of CX project

(3)<Compiler package version>
    CA850: Version -> CX: Version

(4)<Options not to use(Build mode)>
    Command name(Tab name of build tool property)
    Option
    :

(5)<Options to change(Build mode)>
    Command name(Tab name of build tool property)
    Option of CA850 project -> Option of CX project
    :
```

Number	Description
(1)	Time and date on which a project was created The time and date on which a CX project was created by reusing a CA850 project is output using format "ddd, mmmm dd, yyyy hh:mm:ss AM/PM".
(2)	CubeSuite IDE(Integrated Development Environment Framework) version The version of IDE of a CA850 project and the version of IDE of a CX project are output.
(3)	Compiler package version The compiler package used in a CA850 project and the version, and the compiler package used in a CX project and the version are output. The version of CX is the latest version in the compiler packages which are installed in the CubeSuite environment.

Number	Description
(4)	<p>Options not to use(<i>Build mode</i>)</p> <p>If an option that has been set in a CA850 project and is not used in a CX project exists, the information is output for each build mode in the format shown below.</p> <p style="padding-left: 40px;"><i>Command name (Tab name of build tool property)</i></p> <p style="padding-left: 40px;"><i>Option</i></p> <p style="padding-left: 80px;">:</p> <ul style="list-style-type: none"> - This item is output only when the corresponding option exists. - CA850 options are converted into CX options which have the same function. The option that has the same function and different name is not output. - Build modes are output in the following order: "DefaultBuild", user-created build mode ("DefaultBuild" is the build mode that CubeSuite provides by default). <p>See "CubeSuite Build" for detail about a build mode.</p>
(5)	<p>Options to change(<i>Build mode</i>)</p> <p>If an option that has been set in a CA850 project and has been changed to use in a CX project exists (in the case that the function is same as CA850, but the parameter does not exist in CX, so it is changed to other one, and the like), the information is output for each build mode in the format shown below.</p> <p style="padding-left: 40px;"><i>Command name (Tab name of build tool property)</i></p> <p style="padding-left: 40px;"><i>Option of CA850 project -> Option of CX project</i></p> <p style="padding-left: 80px;">:</p> <ul style="list-style-type: none"> - This item is output only when the corresponding option exists. - CA850 options are converted into CX options which have the same function. The option that has the same function and different name is not output. - Build modes are output in the following order: "DefaultBuild", user-created build mode ("DefaultBuild" is the build mode that CubeSuite provides by default). <p>See "CubeSuite Build" for detail about a build mode.</p>

2.7.8 Convert a PM+ project into a CubeSuite project

You can convert a PM+ project into a CubeSuite project automatically by selecting [Open Project...] from the [Project] with CubeSuite.

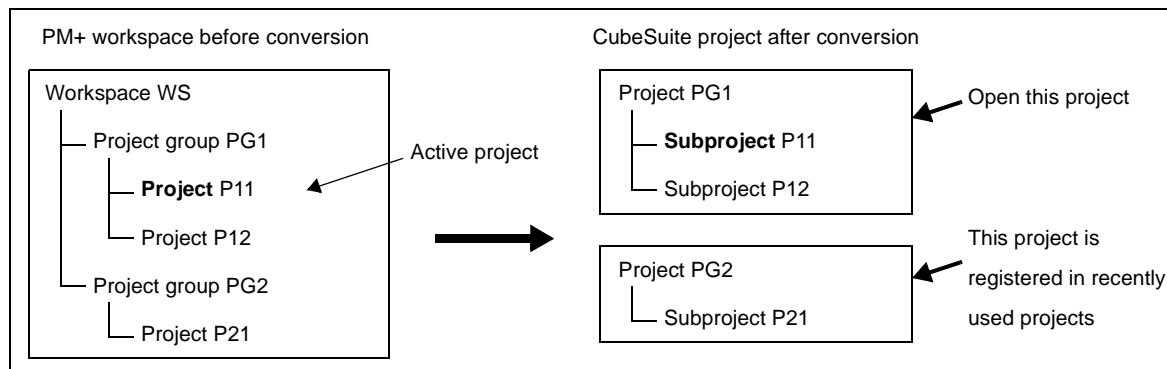
The result of conversion is output to a file as project convert information.

(1) When opening from a PM+ workspace file (*.prw)

Files are converted to CubeSuite projects according to the rules below.

- The PM+ workspace is not converted.
- PM+ project groups are divided into projects with the same name as the project group. After conversion, the project file name will be "*project group name.cspj*".
- PM+ projects are converted to subprojects with the same name as the original project. After conversion, the subproject file name will be "*project name.cssp*".
- After conversion, the first subproject shown in the Project Tree will be the active project.

Figure 2-60. PM+ Project Conversion Image



- Remarks 1.** After conversion, CubeSuite opens the project that was converted from the PM+ project group that includes the active project. Projects other than this project are registered in recently used projects. However, the number of projects that exceeds four is not registered.
2. The project settings after conversion, of those in the PM+ project group, are set to the same settings as the project listed first in the workspace file. In addition, files are not registered.
 3. If you wish to organize each of the projects into a single project after conversion, add each project as a subproject to a single project.
 4. When loading a workspace that includes projects without build tools specified, build tools are set according to the microcontroller.

- Cautions 1.** Only projects that can be loaded by PM+ and built normally can be loaded into CubeSuite.
2. If there is already a subproject file in the same folder and with the same file name (excluding the file extension) as the project file, then the project file will not be saved correctly. Change the name of the main project or the subproject on the project tree.

(2) When opening from a PM+ project file (*.prj)

Files are converted to CubeSuite projects according to the rules below.

- PM+ projects are converted to projects with the same name as the original project. After conversion, the project file name will be "*project name.cspj*".

Remark When reading a project without build tools specified, build tools are set according to the microcontroller.

Caution Only projects that can be loaded by PM+ and built normally can be loaded into CubeSuite.

- Remarks 1.** The target for conversion is the file structure information of the PM+ project file and the build mode/build options information.
2. When build options have been added from the version of the target build tools in the PM+ project to the version of the build tools after conversion, the added options are set to the build tools' default values after conversion.
 3. The project files after conversion are created in the folder the PM+ project file was placed in. If there is already a project file with the same name, "*_number_*" (*number*: 1, 2, ...) will be added to the file name.
 4. After conversion, the build mode will have the same name as that of a PM+ project.
Any characters that are not allowed in a CubeSuite folder name, however (\, /, :, *, ?, ", <, >, |) will be replaced by underscores ("_").
If there is another build mode with the same name after conversion, then the "*_n_*" (*n* = 1, 2, ...) will be appended to the build mode name.
 5. The following PM+ and build-tool versions are supported for conversion: PM+ V6.30 and CC78K0 V4.00/RA78K0 V4.00 or higher [78K0]; PM+ V6.20 and CC78K0R V1.00/RA78K0R V1.00 or higher [78K0R]; and PM+ V6.00 and CA850 V3.00 or higher [V850]. The latest version of the compiler package installed on your computer is set as the version of your build tool.
 6. The result that a PM+ project is converted into a CubeSuite project is output to a file as project convert information.
 - The project convert information file name is "ProjectConvertInformation_*projectname*.txt".
 - The project convert information file is output for each converted project (subproject).
 - The project convert information file is output to the project folder of the project (subproject).
 - The project convert information file is displayed the File node of the project (subproject) on the [Project Tree panel](#).

The output format of the project convert information file is shown below.

```
(1)Time and date on which a project was converted

(2)<IDE version>
    CubeSuite IDE: Version

(3)<Compiler package version>
    Compiler package used in PM+ project: Version -> Compiler package used in
    CubeSuite project: Version

(4)<Options not to use(Build mode)>
    Tool name of PM+(Tab name of build tool property in CubeSuite)
    Option
    :

(5)<Options to change(Build mode)>
    Tool name of PM+(Tab name of build tool property in CubeSuite)
    Option of PM+ project -> Option of CubeSuite project
    :
```

Number	Description
(1)	<p>Time and date on which a project was converted</p> <p>The time and date on which a PM+ project was converted into a CubeSuite project is output using format "<i>dddd, mmmm dd, yyyy hh:mm:ss AM/PM</i>".</p>
(2)	<p>IDE version</p> <p>The version of CubeSuite is output.</p>
(3)	<p>Compiler package version</p> <p>The compiler package used in a PM+ project and the version, and the compiler package used in a CubeSuite project and the version are output.</p> <p>The version of the compiler package used in a CubeSuite project is the latest version in the compiler packages which are installed in the CubeSuite environment.</p>
(4)	<p>Options not to use(<i>Build mode</i>)</p> <p>If an option that has been set in a PM+ project and is not used in a CubeSuite project (option that has been deleted by upgrading the compiler package, and PM+ option) exists, the information is output for each build mode in the format shown below.</p> <p style="text-align: center;"><i>Tool name(Tab name of build tool property in CubeSuite)</i></p> <p style="text-align: center;"><i>Option</i></p> <p style="text-align: center;">:</p> <ul style="list-style-type: none"> - Build modes are output in the following order: "Debug Build", "Release Build", user-created build mode ("Debug Build" and "Release Build" are the build modes that PM+ provides by default. They differ depending on whether the debug information output option is set or not.). - "Other Options" is output if <i>Tab name of build tool property in CubeSuite</i> for <i>Tool name</i> does not exist.
(5)	<p>Options to change(<i>Build mode</i>)</p> <p>If an option that has been set in a PM+ project and has been changed to use in a CubeSuite project (option that the range the parameter can be specified has been changed, and option that has been changed by upgrading the compiler package) exists, the information is output for each build mode in the format shown below.</p> <p style="text-align: center;"><i>Tool name(Tab name of build tool property in CubeSuite)</i></p> <p style="text-align: center;"><i>Option of PM+ project -> Option of CubeSuite project</i></p> <p style="text-align: center;">:</p> <ul style="list-style-type: none"> - Build modes are output in the following order: "Debug Build", "Release Build", user-created build mode ("Debug Build" and "Release Build" are the build modes that PM+ provides by default. They differ depending on whether the debug information output option is set or not.). - "Other Options" is output if <i>Tab name of build tool property in CubeSuite</i> for <i>Tool name</i> does not exist.

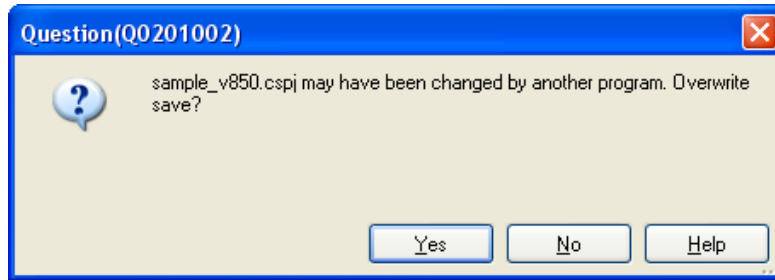
2.8 Save the Project File

The project's settings information is saved to the project file (*.cspj).

Select the [File] or [Project] menu >> [Save Project].

When there is a change in the project, the message dialog box below will open.

Figure 2-61. Message Dialog Box



To continue with the operation, click the [Yes] button on the dialog box.

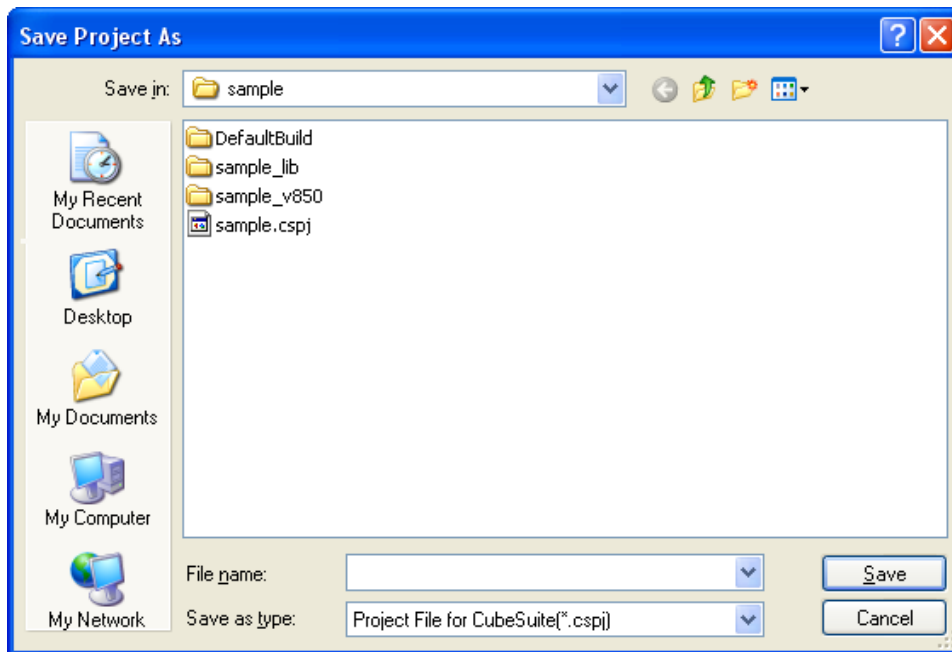
The project file is overwritten with the current settings information.

2.8.1 Save the project file with a different name

You can save the project file with a different name.

From the [File] or [Project] menu, select [Save Project As...], the [Save Project As dialog box](#) will open.

Figure 2-62. Save Project As Dialog Box



Specify the save folder and separate project filename (*.cspj) on the dialog box. The project file is saved in the specified folder with the filename.

Cautions 1. The files registered to the project are the same as those in the original project (the files registered to the project are not copied).

2. If there is already a subproject file in the same folder and with the same file name (excluding the file extension) as the project file, then the project file will not be saved correctly.
Change the name of the main project or the subproject.

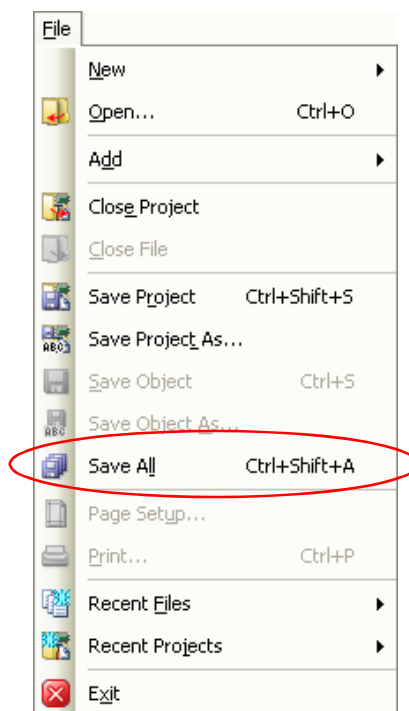
Remark If you save the project file with a different name from the original project file, and then modify one or both of them, then the project files will have different contents.

2.8.2 Save all files

You can save the project file and all the files being edited.

From the [File] or [Project] menu, select [Save All], the project file is overwritten with the current settings information and all files being edited are saved.

Figure 2-63. [Save All] Item

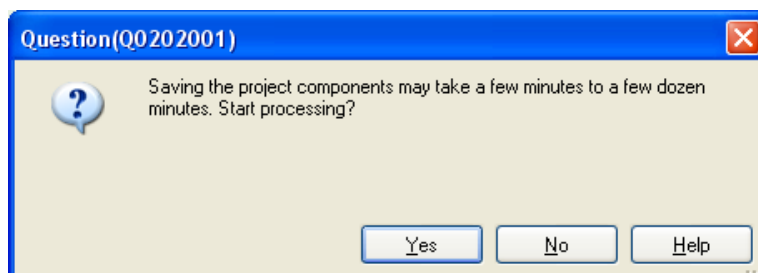


2.8.3 Pack and save the project and CubeSuite

You can copy the product suite (IDE, language tools, microcontroller information, etc.) and project set (also includes subprojects if they exist) to a specified folder and save it organized into a single folder.

Select the Project node on the project tree and if you select [Save Project and CubeSuite as Package] on the context menu, the message dialog box below will open.

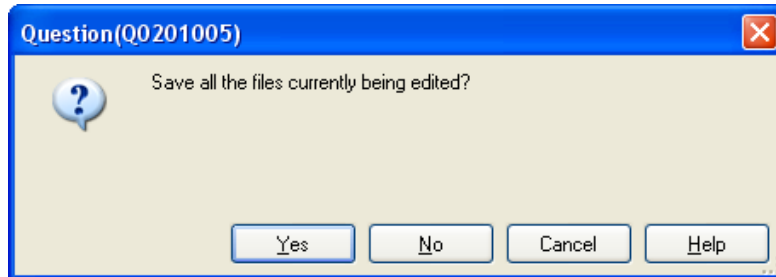
Figure 2-64. Message Dialog Box



To continue with the operation, click the [Yes] button on the dialog box.

When using external tools such as an external editor and files are being edited, the message dialog box below will open.

Figure 2-65. Message Dialog Box

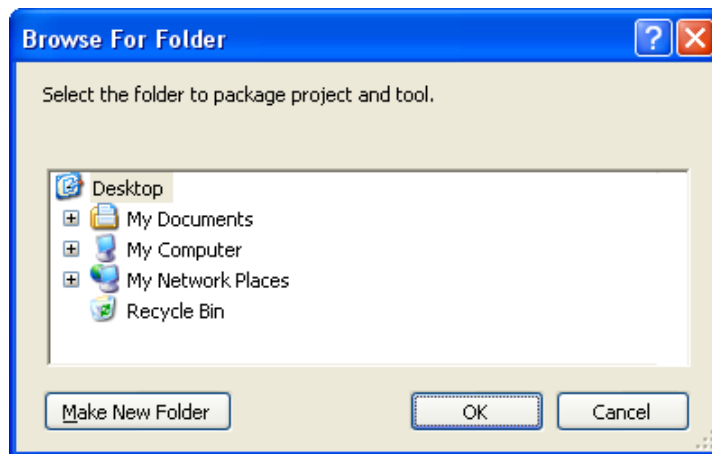


Remark This dialog box only appears if in the [Option dialog box](#), in the [\[General - Text Editor\]](#) category, you selected the [\[Use external text editor\]](#) property.

To continue with the operation, click the [OK] button on the dialog box.

The [Browse For Folder dialog box](#) will open.

Figure 2-66. Browse For Folder Dialog Box

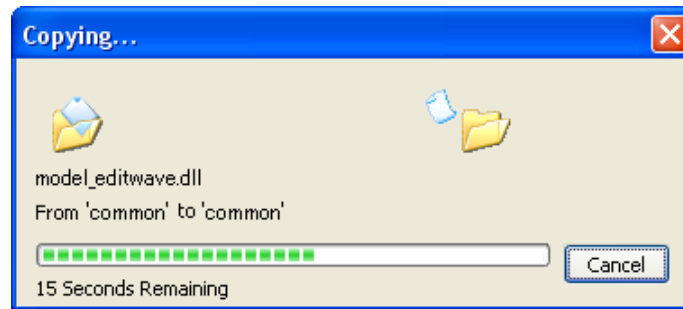


Specify the save location folder on the dialog box.

Caution Specify a folder other than the project folder to save to.

If you click the [OK] button, the copy process will begin. The dialog box below is displayed during the copying process.

Figure 2-67. Copying



The save location folder structure is shown below.

Save location

— readme.txt	: Explanation of the structure of
— errorlog.txt	: A record of the files that
— <i>Project name</i>	: Project folder
— <i>Install folder</i>	: The folder CubeSuite is
— Copy_Files	: The copy location folder for files not placed in the project
— copylog.txt	: A record of the copied files
— 1	: Storage folder 1 for copied files ^{Note}
— 2	: Storage folder 2 for copied files ^{Note}
— :	

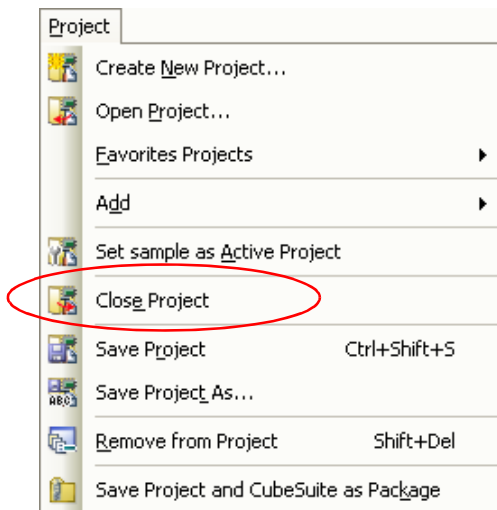
Note The digits in the folder name are adjusted by the required number of folders. For example, for 2 digits, the name is "01", "02", and so on.

- Cautions**
1. Information on the start menu is not a subject of the save. When using the save location tools, follow the content of readme.txt in the save folder.
 2. Information on the tool installation is not a subject of the save. To uninstall the save location tools, delete the entire save folder.
 3. Custom settings made in the [Option dialog box](#) and [User Setting dialog box](#) are not saved. When you use tools from the saved file, the environment will be configured to the default settings.
 4. Tools in the save folder cannot be updated.
 5. When saving, we recommend the files registered to the project be placed inside the project folder when possible. Files not placed inside the project folder are copied to the Copy_files folder when saving, so when using the save location project, you must re-register the files to the project.
 6. Emulator drivers are not saved. If you use a pack on another computer, you must install the drivers separately.

2.8.4 Close a project

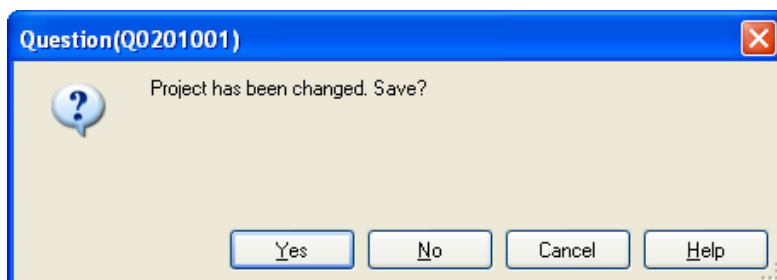
To close a project, select the [File] or [Project] menu >> [Close Project].

Figure 2-68. [Close Project] Item



When the open project or changed files are not saved, a Message dialog box is displayed.

Figure 2-69. Message Dialog Box



Click the [Yes] button to save, click the [No] button to not save.

2.9 Changing the Window Layout

This section describes how to change the CubeSuite window layout.


2.9.1 Automatically hide panels

Panels support the auto-hide feature.

When the auto-hide feature is in use, unused panels are minimized on the edge of the frame, making it possible to display more information at one time.

(1) Enabling the auto-hide feature

To enable the auto-hide feature, click on the panel to hide to select it.

Right click on the panel's title bar, and select [Auto hide] or click on the panel's title bar .


When an auto-hidden panel loses the focus, a minimized icon and its panel name appear on the edge of the frame as a tab.

To display the auto-hidden panel, move the mouse cursor over the tab. The panel opens from the tab, and becomes available.

When the panel loses the focus again, the panel will be minimized as the tab.

(2) Disabling the auto-hide feature

To disable the auto-hide feature, click on the panel to hide to select it.

Right click on the panel's title bar, and select [Auto hide] or click on the panel's title bar .

2.9.2 Floating a window

To float a window, click on its title bar and move it.

You can also double click the title bar of the window you wish to float, or right click and select [Floating].

Figure 2-70. Context Menu












2.9.3 Docking windows

You can dock a floating window to the main window.

Click and drag the title bar of the window you wish to dock. Docking indicators appears automatically in the center, top, bottom, left, and right of the main window.

When the mouse pointer moves over one of the docking indicators, part of the window background becomes blue. If you release the mouse pointer at this point, the window will be docked in the blue area.



Selecting the indicators allows the window to be placed freely, as shown below.

	Places in the upper part of CubeSuite
	Places in the left part of CubeSuite
	Places in the right part of CubeSuite
	Places in the lower part of CubeSuite
Above 	Places in the upper part of the target panel
Left of 	Places in the left part of the target panel
Right of 	Places in the right part of the target panel
Below 	Places in the bottom part of the target panel
Center of 	Places in the target panel as a tab

You can also double click the title bar of the window or right click and select [Floating]. This docks the window at its former docking position.

You can adjust the size of windows docked to the main window by clicking and dragging the splitters.

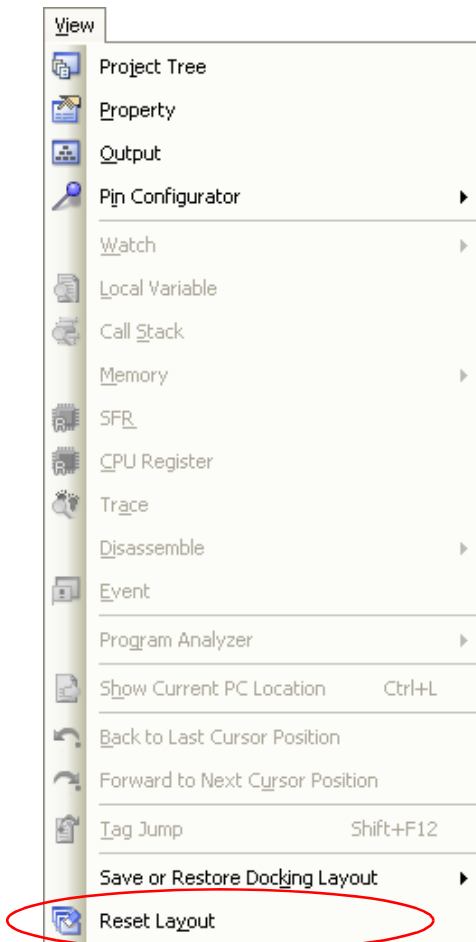
2.9.4 Displaying multiple panels

Click on a panel you wish to view, and drag it over another panel you wish to view simultaneously. As described in "2.9.3 Docking windows", a docking indicator appears. Move the mouse pointer to , and release the mouse over the location where you wish to  place the panel (left, right, top, or bottom).

2.9.5 Resetting the window layout

From the [Main window's](#) [View] menu, select [Reset Layout]. The window layout is returned to its initial state.

Figure 2-71. [Reset Layout] Item



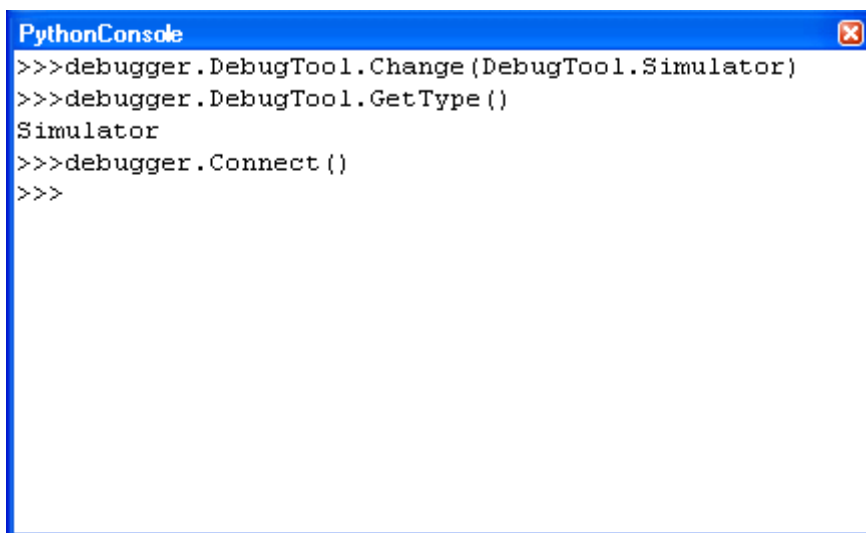
2.10 Execute Python Functions

CubeSuite enables the execution of IronPython functions and control statements, and CubeSuite Python functions (see “[F.3 CubeSuite Python Functions](#)”) added for controlling CubeSuite via command input method.

Select [Python Console] from the [View] menu. The [Python Console panel](#) opens.

You can control CubeSuite and the debugging tool by executing Python functions and control statements in the panel.

Figure 2-72. Python Console Panel



```
PythonConsole
>>>debugger.DebugTool.Change(DebugTool.Simulator)
>>>debugger.DebugTool.GetType()
Simulator
>>>debugger.Connect()
>>>
```

Remark See "[APPENDIX F Python CONSOLE/Python FUNCTIONS](#)" for details about the Python console and Python functions.

2.11 Manipulate CubeSuite on the Command Line

You can launch CubeSuite from the command line (e.g. the Windows command prompt), and control it via command input without displaying the [Main window](#).

This section describes the control of CubeSuite when launched from the Windows command prompt. From the command prompt, execute CubeSuite.exe or Cube-SuiteW.exe, located in the CubeSuite installation folder.

(1) When executing CubeSuite.exe

If you execute CubeSuite.exe, you can launch CubeSuite, load plugins, and execute builds without displaying the [Main window](#).

The format to specify on the command line is shown below.

```
CubeSuite.exeΔ[[[/bb|/br|/bcb|/bcΔ[build-mode-name]]]Δ[[/npΔplug-in-name[, plug-in-name, ...]]]Δ[/npall]Δ[plug-in-optionΔ[plug-in-parameter, ...]]Δproject-file-name]
```

- Δ: One or more spaces
- []: Can be omitted
- |: When options are separated by pipeline characters ("|"), any one of the options can be specified
- ...: Pattern in proceeding [] can be repeated

Each option is described below.

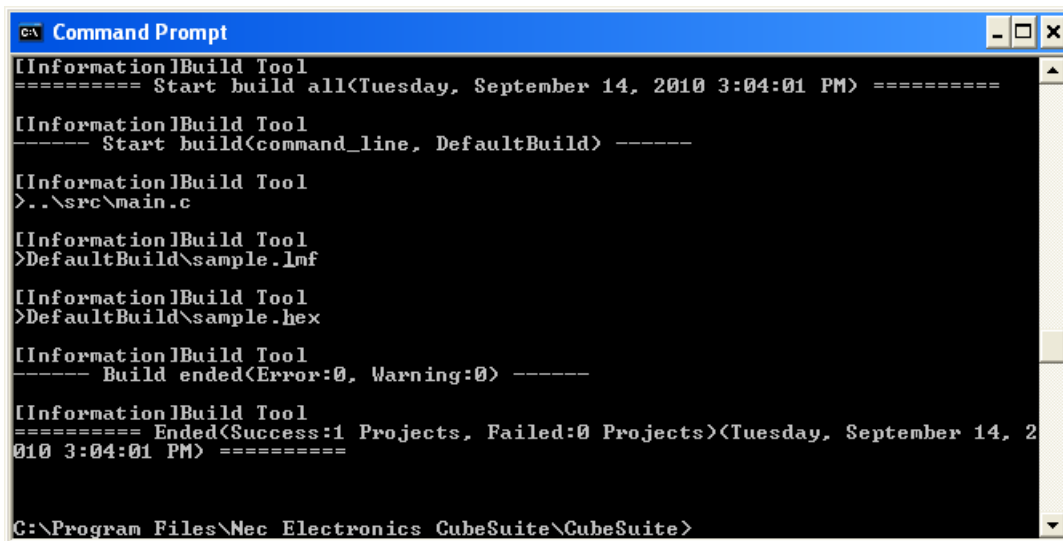
Option	Description
None	Launch CubeSuite without displaying the Main window, and exit without performing any actions.
/bbΔ[build-mode-name]	Execute a build. Launch without displaying the Main window, build all the projects included in specified <i>project-file-name</i> , with the build mode specified by <i>build-mode-name</i> , and then exit. If a project does not have the build mode specified by <i>build-mode-name</i> , then the build mode is copied based on the DefaultBuild, and the build is performed. If the build mode specified by <i>build-mode-name</i> is not defined in the project specified by <i>project-file-name</i> , then an error will be displayed, and processing will end. If <i>build-mode-name</i> is omitted, then the build will use the DefaultBuild. If <i>project-file-name</i> is omitted, then an error will be displayed, and processing will end.
/brΔ[build-mode-name]	Execute a rebuild. Launch without displaying the Main window, build all the projects included in specified <i>project-file-name</i> , with the build mode specified by <i>build-mode-name</i> , and then exit. If a project does not have the build mode specified by <i>build-mode-name</i> , then the build mode is copied based on the DefaultBuild, and the build is performed. If the build mode specified by <i>build-mode-name</i> is not defined in the project specified by <i>project-file-name</i> , then an error will be displayed, and processing will end. If <i>build-mode-name</i> is omitted, then the build will use the DefaultBuild. If <i>project-file-name</i> is omitted, then an error will be displayed, and processing will end.

Option	Description
<code>/bcbΔ[build-mode-name]</code>	<p>Perform a clean and then execute a build.</p> <p>Launch without displaying the Main window, build all the projects included in specified <i>project-file-name</i>, with the build mode specified by <i>build-mode-name</i>, and then exit.</p> <p>If a project does not have the build mode specified by <i>build-mode-name</i>, then the build mode is copied based on the DefaultBuild, and the build is performed.</p> <p>If the build mode specified by <i>build-mode-name</i> is not defined in the project specified by <i>project-file-name</i>, then an error will be displayed, and processing will end.</p> <p>If <i>build-mode-name</i> is omitted, then the build will use the DefaultBuild.</p> <p>If <i>project-file-name</i> is omitted, then an error will be displayed, and processing will end.</p>
<code>/bcΔ[build-mode-name]</code>	<p>Perform a clean.</p> <p>Launch without displaying the Main window, build all the projects included in specified <i>project-file-name</i>, with the build mode specified by <i>build-mode-name</i>, and then exit.</p> <p>If a project does not have the build mode specified by <i>build-mode-name</i>, then the build mode is copied based on the DefaultBuild, and the build is performed.</p> <p>If the build mode specified by <i>build-mode-name</i> is not defined in the project specified by <i>project-file-name</i>, then an error will be displayed, and processing will end.</p> <p>If <i>build-mode-name</i> is omitted, then the build will use the DefaultBuild.</p> <p>If <i>project-file-name</i> is omitted, then an error will be displayed, and processing will end.</p>
<code>/npΔplug-in-name[, plug-in-name, ...]</code>	<p>Start CubeSuite without displaying the Main window, and without loading the specified plugins in the Plugins folder.</p> <p>In <i>plug-in-name</i>, specify the name of the folder in which each DLL file is saved. Note that folder names are case-insensitive.</p> <p>You can specify multiple plugin names by separating them with commas.</p> <p>You can also specify the "/np" option itself multiple times. Each DLL will not be loaded.</p> <p>If the plugin specified by this option does not exist, it will be ignored.</p> <p>If <i>project-file-name</i> is omitted, then an error will be displayed, and processing will end.</p>
<code>/npall</code>	<p>Start CubeSuite without displaying the Main window, and without loading any of the specified plugins in the Plugins folder.</p> <p>If this is specified together with the "/np" option, then this option will take precedence.</p> <p>If <i>project-file-name</i> is omitted, then an error will be displayed, and processing will end.</p>
<code>plug-in-optionΔ[plug-in-parameter, ...]</code>	Specify an option for the plug-in (see "(3) Plug-in Options").
<code>project-file-name</code>	Start CubeSuite without displaying the Main window, with the specified project file loaded.

Remark While CubeSuite is running, press the [Ctrl] + [C] keys to forcibly terminate CubeSuite.

The results of execution are output to the command prompt.

Figure 2-73. Command Prompt (If Build Is Executed with CubeSuite.exe)



(2) When executing CubeSuiteW.exe

Execute CubeSuiteW.exe to launch CubeSuite, displaying the [Main window](#) and with plugins loaded, in the same manner as when launching it from the [Start] menu.

The format to specify on the command line is shown below.

```
CubeSuiteW.exeΔ[[/npΔplug-in-name[,plug-in-name,...]]Δ[/npall]Δ[plug-in-optionΔ[plug-in-parameter,...]] [Δproject-file-name]
```

- Δ: One or more spaces
- []: Can be omitted
- ...: Pattern in proceeding [] can be repeated

Each option is described below.

Option	Description
None	Start CubeSuite with the Main window displayed, in the same manner as when launching it from the [Start] menu.
/npΔplug-in-name[,plug-in-name,...]	Start CubeSuite displaying the Main window, without loading the specified plugins in the Plugins folder. In <i>plug-in-name</i> , specify the name of the folder in which the DLL file is saved. Note that folder names are case-insensitive. You can specify multiple plug-in names by separating them with commas. You can also specify the "/np" option itself multiple times. Each DLL will not be loaded. If the plugin specified by this option does not exist, it will be ignored.
/npall	Start CubeSuite displaying the Main window, without loading any of the specified plug-ins in the Plugins folder. If an invalid option is specified, it will be ignored.
plug-in-optionΔ[plug-in-parameter,...]	Specify an option for the plugin (see "(3) Plug-in Options").
project-file-name	Start CubeSuite displaying the Main window, with the specified project file loaded.

(3) Plug-in Options

Below are the plug-in options that can be specified.

Option	Description
<code>/psΔscript-file-name</code>	This is the Python Console plugin option. After loading the project file in CubeSuite, run the code in <i>script-file-name</i> . If an unnecessary parameter is specified, and the script file cannot be loaded, then an error will be displayed, and the script file will not be executed.

A sample script file is shown below.

```
debugger.Connect ()
debugger.Download.LoadModule (r"C:\project\test\DefaultBuild\test.out")
debugger.Register.GetValue ("pc")
breakpoint = BreakCondition ()
breakpoint.Address = "func"
debugger.Breakpoint.Set (breakpoint)
debugger.Go (GoOption.WaitBreak)
debugger.Register.GetValue ("pc")
```


APPENDIX A WINDOW REFERENCE

This section describes the windows, panels, and dialog boxes related to installation, updates, license settings, and starting CubeSuite.

A.1 Description

Below is a list of the windows, panels, and dialog boxes related to installation, updates, license settings, and starting CubeSuite.

Table A-1. Window/Panel/Dialog Box List

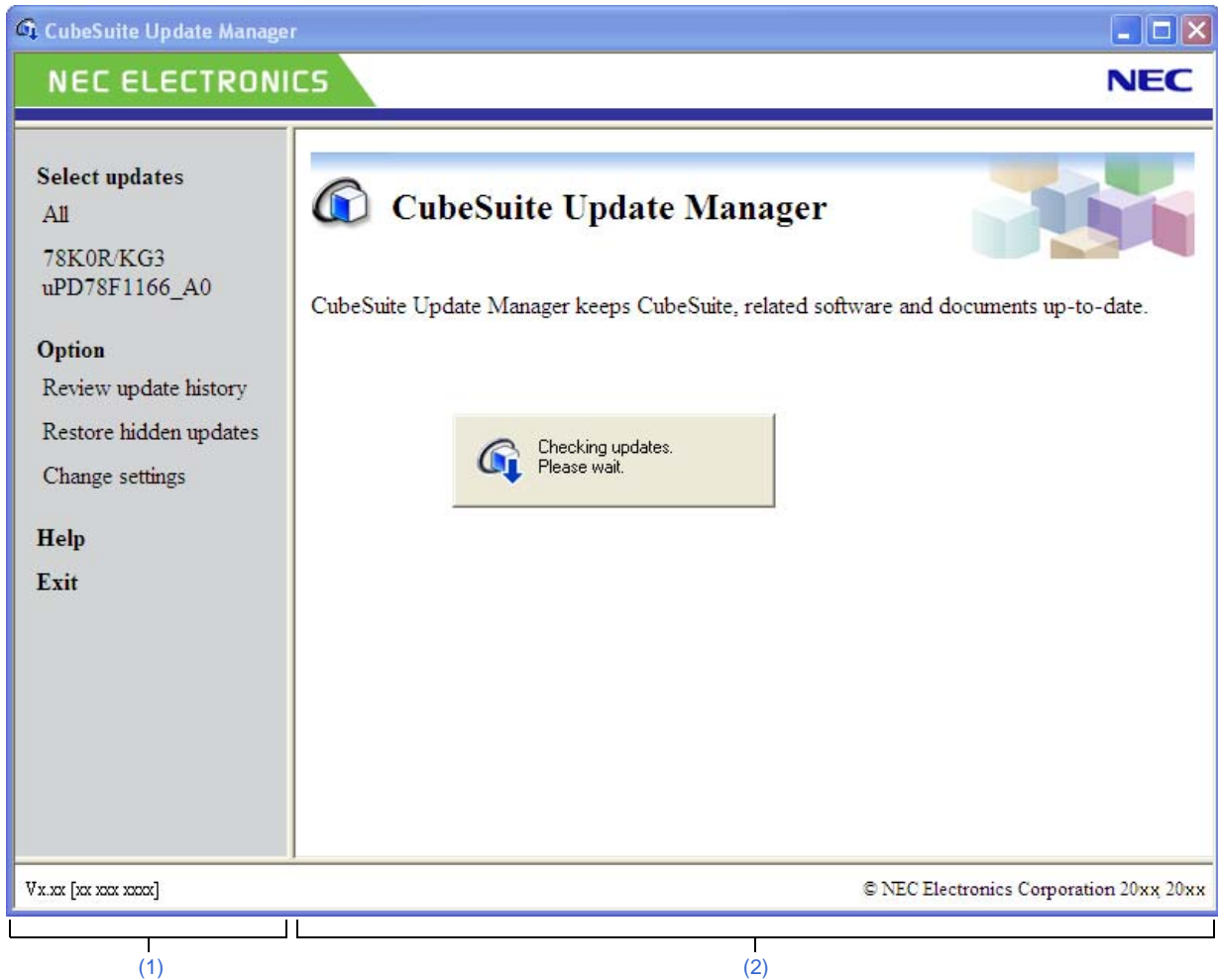
Window/Panel/Dialog Box Name	Function Description
CubeSuite Update Manager window	This window displays the status of and controls the update function.
Checking for Updates dialog box	This dialog box appears while the application is checking for updates.
Update in progress dialog box	This dialog box displays the progress of update download and installation.
Update Manager Options dialog box	This dialog box displays and changes the Update Manager options.
Task Tray	When the application is checking for or downloading updates in the background, an icon appears in the Windows task tray.
CubeSuite License Manager window	This window displays a list of licenses, and adds and deletes licenses.
Main window	This is the start-up window that opens when CubeSuite is launched.
Start panel	This panel allows you to easily open a tutorial, access (create/open) a project, or load a sample project.
Create Project dialog box	This dialog box is used to create new projects or subprojects.
Source Convert Setting dialog box [CX]	This dialog box configures the composition files of the source project (the source files and the like) to convert them for the build tool of the project to be created.
Project Tree panel	This panel is used to display components of the microcontroller, build tool, and source file of the project in tree view.
Property panel	In this panel, the detailed information on the node that is selected in the Project Tree panel is displayed categorized. Also, the settings of the selected node can be changed.
Editor panel	This panel is used to display and edit text files and source files.
Output panel	The message that is output from the build tool/debug tool/each plug-in or the result of the Whole Search with the Search and Replace dialog box is displayed.
Add File dialog box	This dialog box is used to create a new file and add it to the project.
Add Folder and File dialog box	This dialog box is used to add existing files and folder hierarchies to the project.
Character String Input dialog box	This dialog box is used to input and edit characters in one line.
Text Edit dialog box	This dialog box is used to input and edit texts in multiple lines.
Build Mode Settings dialog box	This dialog box is used to add and delete build modes and configure the current build mode in batch.
Batch Build dialog box	This dialog box is used to do build, rebuild and clean process in batch with the build mode that the project has.
Go to the Location dialog box	This dialog box is used to move the caret to the designated location.

Window/Panel/Dialog Box Name	Function Description
Save Settings dialog box	This dialog box is used to set the encoding and newline code of the file that is being edited on the Editor panel .
Search and Replace dialog box	This dialog box is used to search and replace the designated characters.
Progress Status dialog box	This dialog box is used to display how the process has been progressed when the time consuming process is taken place.
Option dialog box	This dialog box is used to configure the CubeSuite environment.
User Setting dialog box	This dialog box allows you to customize toolbars and menus displayed in the Main window .
New Toolbar dialog box	This dialog box is used to create a new toolbar to appear in the Main window .
Rename Toolbar dialog box	This dialog box is used to edit the name of a toolbar created by the user.
Customize Keyboard dialog box	This dialog box is used to assign shortcut keys to the various commands.
Rearrange Commands dialog box	This dialog box allows you to change the arrangement (including addition and deletion) of menu items and buttons in the Main window .
Version Information dialog box	This dialog box is used to display versions of CubeSuite and each plug-in product.
Contact Information for Technical Support dialog box	This dialog box is used to display necessary information to contact to Tool Support Center.
One Point Advice dialog box	This dialog box is used to display tips for using CubeSuite.
Other &Windows... dialog box	This dialog box is used to select one of the divide panels shown in the Main window to activate or close.
Open Project dialog box	This dialog box is used to open an existing project or select the project file to designate the project to divert when creating a new project.
Add Existing Subproject dialog box	This dialog box is used to select subprojects for adding existing subprojects to projects.
Browse For Folder dialog box	This dialog box is used to select the folder or file output destination (e.g. source code or report file) for the caller of this dialog box.
Save Project As dialog box	This dialog box is used to save project files as different names.
Save As dialog box	This dialog box is used to save the editing file or contents of each panel to a file with a name.
Select Program dialog box	This dialog box is used to select the executable file of an external tool.
Select External Text Editor dialog box	This dialog box is used to select the executable file of an external text editor.
Python Console panel	This panel is used to use IronPython to control CubeSuite and the debug tool via the command input method.
CubeSuite Uninstaller window	This window is used to specify one or more installed CubeSuite products to uninstall at once.

CubeSuite Update Manager window

This window displays the status of and controls the update function.



Figure A-1. CubeSuite Update Manager Window



The following items are explained here.

- [How to open]
- [Description of each area]

[How to open]

- From the Windows [Start] menu, select [Programs] >> [NEC Electronics CubeSuite] >> [Update Manager].
- From the CubeSuite [Help] menu, select [Check for Updates...].
- In the Windows task tray, click the  or  icon.

[Description of each area]

(1) Menu area

This area displays a menu for changing the appearance and settings of the main area. Select an item to perform the corresponding action.

All	Displays the updates.
<i>Microcontroller Name</i>	Displays the updates corresponding to that microcontroller. (Displays up to five microcontrollers for which the specify microcontroller function was used, most recent first.)
Download and install	Downloads the selected updates, and after the downloads are complete, install them.
Download only	Downloads the selected updates only.
Review update history	Displays the update history.
Restore hidden updates	Displays the Restore hidden updates page.
Change settings	Displays the Update Manager Options dialog box .
Help	Displays help.
Exit	Exits Update Manager.

(2) Main area

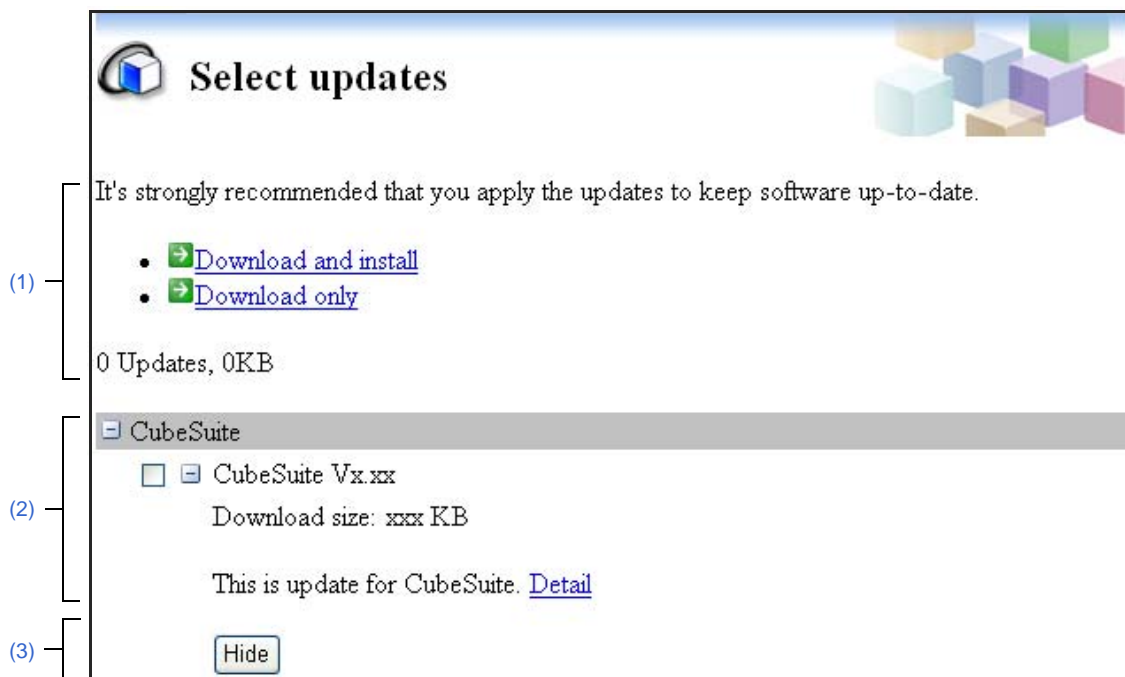
Switch the display to one of the following pages, in accordance with the active function.

- [Select updates page](#)
- [Download and install page](#)
- [Download only page](#)
- [Finish page](#)
- [History page](#)
- [Restore hidden updates page](#)
- [Error page](#)

Select updates page

This page displays a list of available updates. Select updates to download and install.

Figure A-2. Select updates Page



The following items are explained here.

- [How to open]
- [Description of each area]

[How to open]

- From the CubeSuite Update Manager window menu area, click [All].
- From the CubeSuite Update Manager window menu area, click [Microcontroller Name].

[Description of each area]

(1) Select updates area

If a microcontroller was specified, the microcontroller name appears.

Download and install	Downloads the selected updates, and after the downloads are complete, install them.
Download only	Downloads the selected updates only.

(2) Update selection area

Displays a list of items to update.

The following information is displayed for each update.

Category	Displays the category of the update. Updates with the same category are displayed together.
Title	Displays the title of the update.
Download size	Displays the download size of the update.
Summary	Displays a summary of the update information.
Detail	This appears if there is detailed information about the update. Selecting it will display details in a browser.

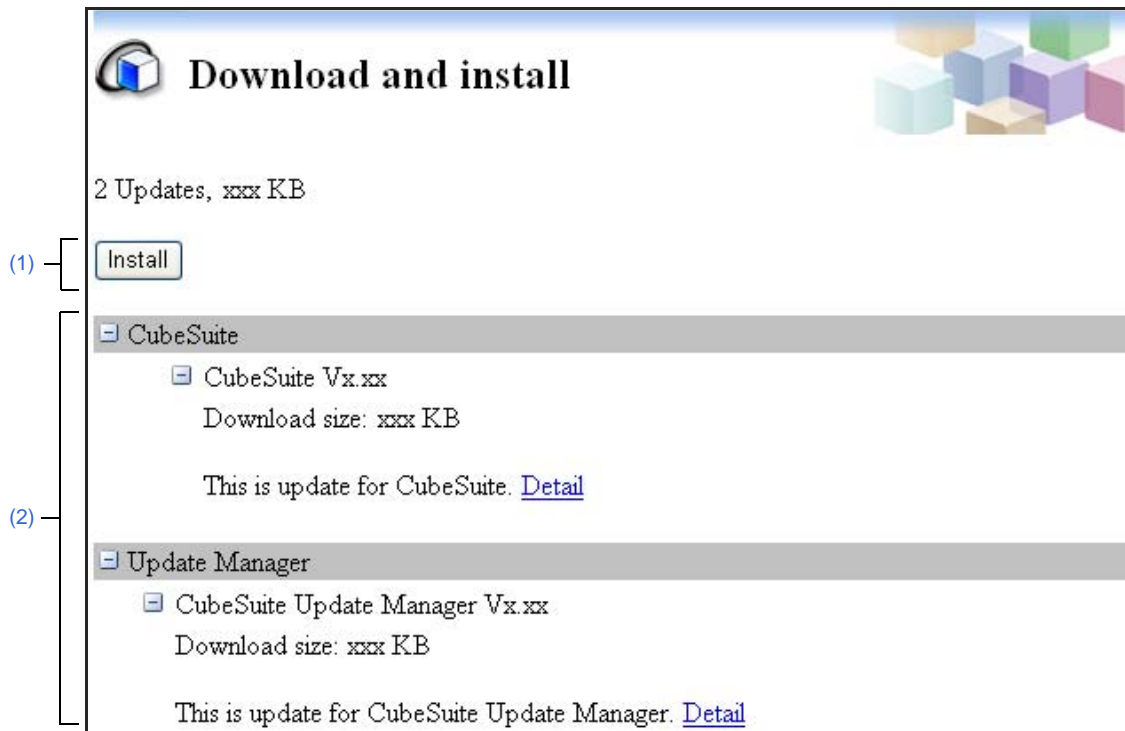
(3) Button [Hide]

If an update is selected in the Update Selection area, it is completely hidden.

Download and install page

This page displays a list of selected updates, and starts the download and installation process.

Figure A-3. Download and install Page



The following items are explained here.

- [How to open]
- [Description of each area]

[How to open]

- From the [CubeSuite Update Manager window](#) menu area, click [Download and install].
- From the [Select updates page](#) main area, click [Download and install].

[Description of each area]

(1) Button [Install]

Begins the download and installation process.
This button is disabled if no updates are selected.

Caution If a driver is updated on Windows Vista, the Driver Warning dialog box may appear during installation. If you choose to cancel the installation in the warning dialog box, the driver installation will fail.

(2) Update Selection area

Displays a list of items to update.

The following information is displayed for each update.

Category	Displays the category of the update. Updates with the same category are displayed together.
Title	Displays the title of the update.
Download size	Displays the download size of the update.
Summary	Displays a summary of the update information.
Detail	This appears if there is detailed information about the update. Selecting it will display details in a browser.

Download only page

This page displays a list of selected updates, and starts the download process.

Figure A-4. Download only Page



The following items are explained here.

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

[How to open]

- From the [CubeSuite Update Manager window](#) menu area, click [Download only].
- From the [Select updates page](#) main area, click [Download only].

[Description of each area]

(1) Button [Download]

Begins the download.

This button is disabled if no updates are selected.

(2) Update Selection area

Displays a list of items to update.

The following information is displayed for each update.

Category	Displays the category of the update. Updates with the same category are displayed together.
Title	Displays the title of the update.
Download size	Displays the download size of the update.
Summary	Displays a summary of the update information.
Detail	This appears if there is detailed information about the update. Selecting it will display details in a browser.

Finish page

This page displays lists of successful, canceled, and failed updates.

Figure A-5. Finish Page



The following items are explained here.

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

[How to open]

- Opens automatically when the update completes.

[Description of each area]

(1) Update Results area

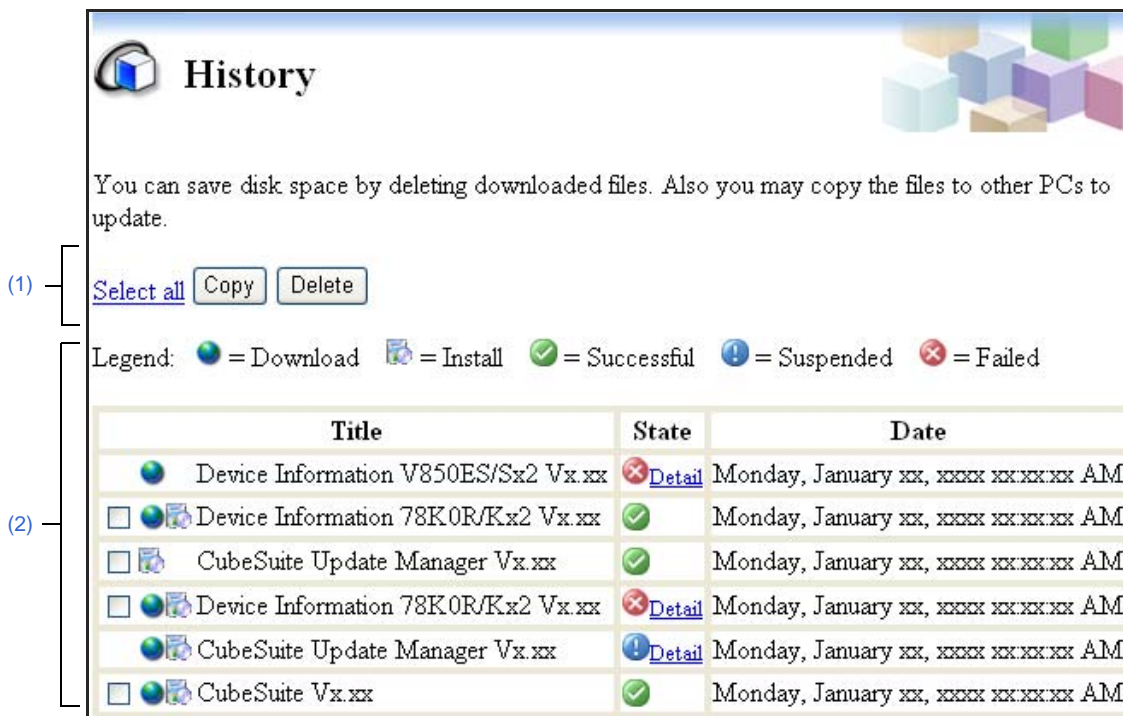
Displays a list of update results.

	Displays number of successful updates and update titles.
	Displays number of canceled updates, update titles, and messages.
	Displays number of failed updates, update titles, and messages.

History page

Use this page to display the history of updates performed, and to copy or delete update files.

Figure A-6. History Page



The following items are explained here.

- [How to open]
- [Description of each area]

[How to open]

- From the [CubeSuite Update Manager window](#) menu area, click [Review update history].

[Description of each area]

(1) Button area

Click [Select all] to select all the check boxes in the list.

Button	Function
Copy	Copies the selected updates to the specified folder.
Delete	Deletes the selected updates.

(2) Update History Display area

Displays a history of updates performed.

In the status column of the list, click [Detail] to display a message dialog box with a message corresponding to the results of the selected update.

Restore hidden updates page

Use this page to display a list of updates hidden in the [Select updates page](#), restore the visibility of the selected updates, and enable that update to be checked and installed again.

Figure A-7. Restore hidden updates Page



The following items are explained here.

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

[How to open]

- From the [CubeSuite Update Manager window](#) menu area, click [Restore hidden updates].

[Description of each area]

(1) Button [Restore]

The selected updates become visible, enabling them to be re-checked and installed.

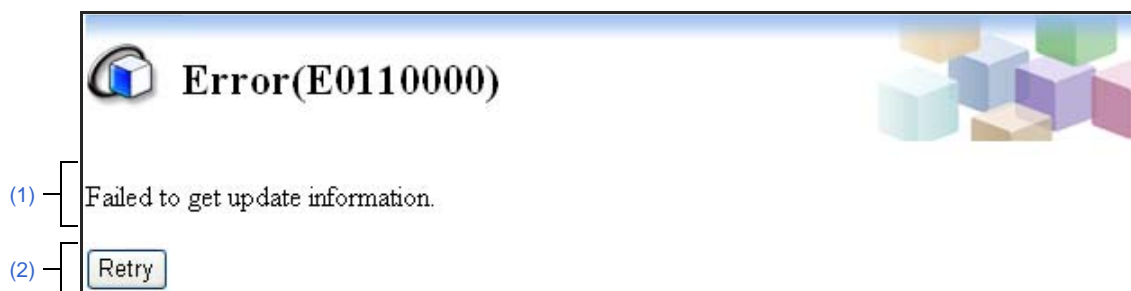
(2) Hidden Updates Display area

Displays a list of updates that were hidden via the [Select updates page](#).

Error page

This page appears when acquisition of update information fails.

Figure A-8. Error Page



The following items are explained here.

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

[How to open]

- Opens automatically when acquisition of update information fails.

[Description of each area]

(1) Message area

Displays a message that the acquisition of update information fails.

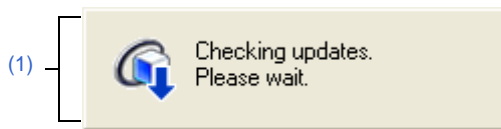
(2) Button [Retry]

Re-acquires update information.

Checking for Updates dialog box

This dialog box appears while the application is checking for updates.

Figure A-9. Checking for Updates Dialog Box



The following items are explained here.

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

[How to open]

- Opens automatically while checking for updates.

[Description of each area]

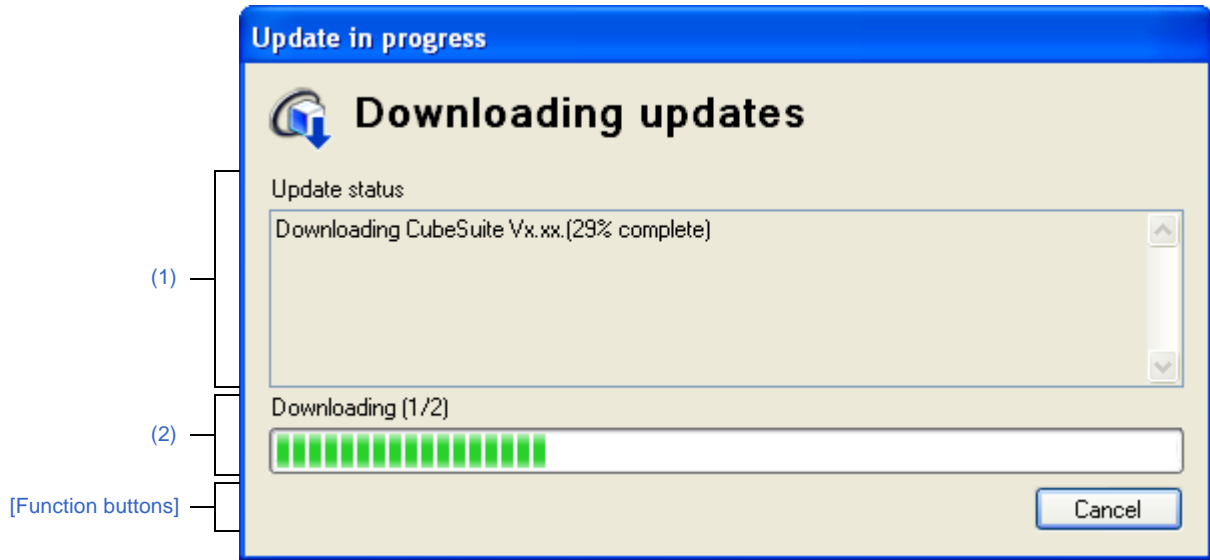
(1) Checking for Updates message area

This area displays messages while checking for updates.
When the update check ends, it closes automatically.

Update in progress dialog box

This dialog box displays the progress of update download and installation.

Figure A-10. Update in progress Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- Opens automatically while downloading and installing updates.

[Description of each area]

(1) [Update status] area

This area displays the current status of downloads and installs.
It automatically closes when the updates are downloaded and installed.

(2) [Downloading/Installing] area

Displays the proportion of the size of target downloads that is complete.
Displays the proportion of the number of target updates installed that is complete.

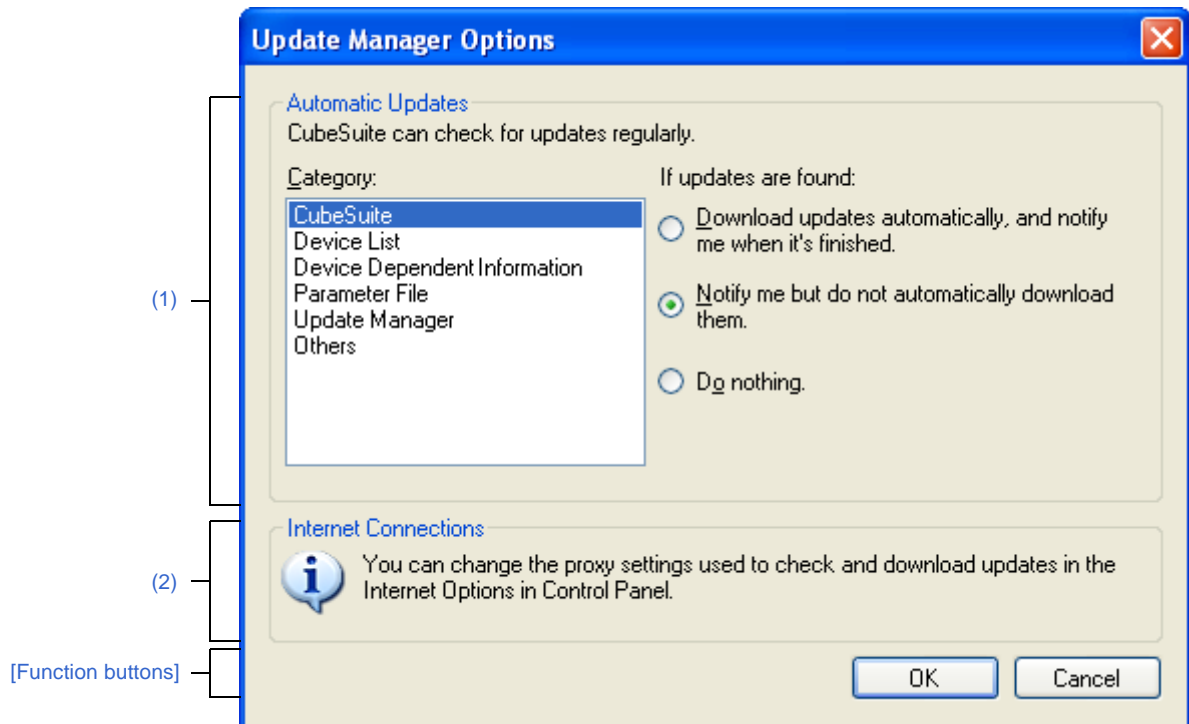
[Function buttons]

Button	Function
Cancel	Cancels the update.

Update Manager Options dialog box

This dialog box displays and changes the Update Manager options.

Figure A-11. Update Manager Options Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [CubeSuite Update Manager window](#) menu area, click [Change settings].

[Description of each area]

(1) [Automatic Updates] area

(a) [Category]

Select the category of for which to configure the behavior for found updates.

(b) [If updates are found]

Set the behavior for found updates.

Select the action to perform for the selected category.

- [Download updates automatically, and notify me when it's finished.]

If corresponding updates are found during periodic checks for updates, download them without asking for confirmation, and notify the user when the download is complete.

- [Notify me but do not automatically download them.]
If corresponding updates are found during periodic checks for updates, notify the user.
- [Do nothing.]
If corresponding updates are found during periodic checks for updates, do not notify the user.

(2) [Internet Connections] area

Describes the Internet connection.

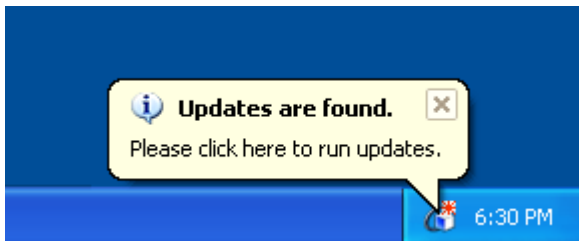
[Function buttons]

Button	Function
OK	Finalizes the settings, reflects them, and closes the dialog box.
Cancel	Cancels any changes made to the settings, and closes the dialog box.

Task Tray

When the application is checking for or downloading updates in the background, an icon appears in the Windows task tray.

Figure A-12. Sample Task Tray Display



The following items are explained here.

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




[How to open]

- Appears automatically when the application is checking for or downloading updates in the background.

[Description of each area]

(1) Task tray

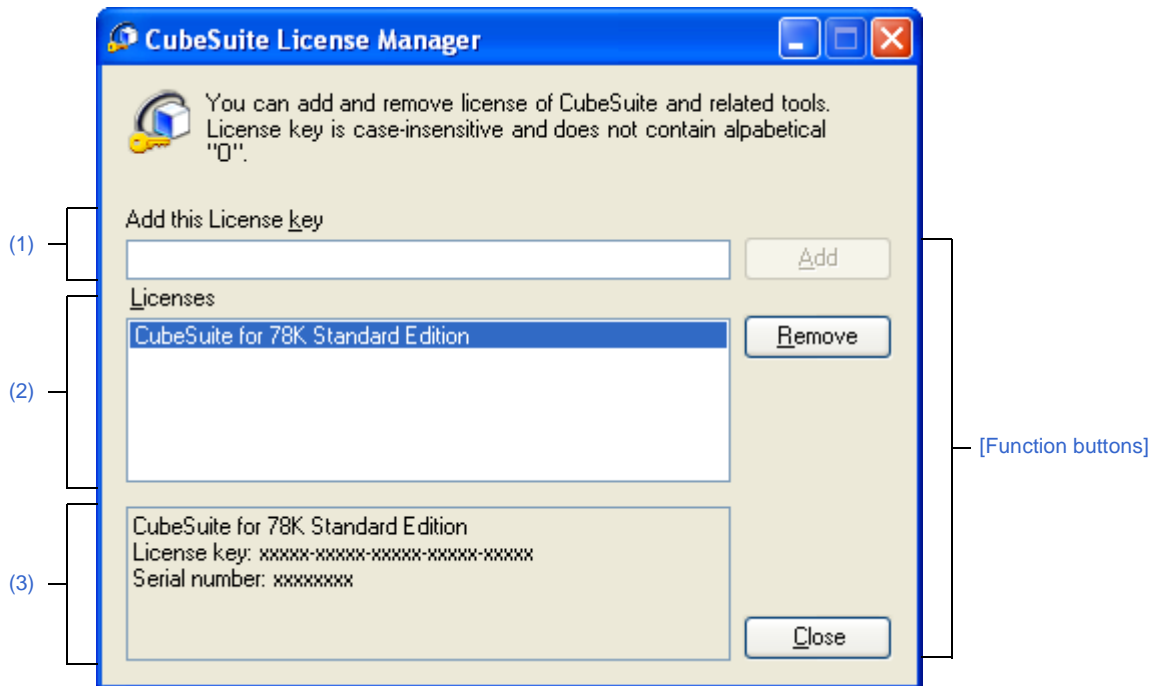
The following icons appear when the application is checking for or downloading updates in the background.

	This icon indicates that the application is checking for updates.
	This icon indicates that a download is in progress. The progress is shown as a percentage (%).
	This is an update notification icon. It also displays the details of the notification with the  icon. Click it to display the CubeSuite Update Manager window .

CubeSuite License Manager window

This window displays a list of licenses, and adds and deletes licenses.

Figure A-13. CubeSuite License Manager Window



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the Windows [Start] menu, select [Programs] >> [NEC Electronics CubeSuite] >> [CubeSuite License Manager].
- Upon installation, in the License registration window, click [License Manager...].

[Description of each area]

(1) [Add this License key] area

Enter a license key in this area.

(2) [Licenses] area

This area displays a list of valid and registered licenses.

(3) License Information area

When a license is selected in the License area, this area displays details about the selected license.

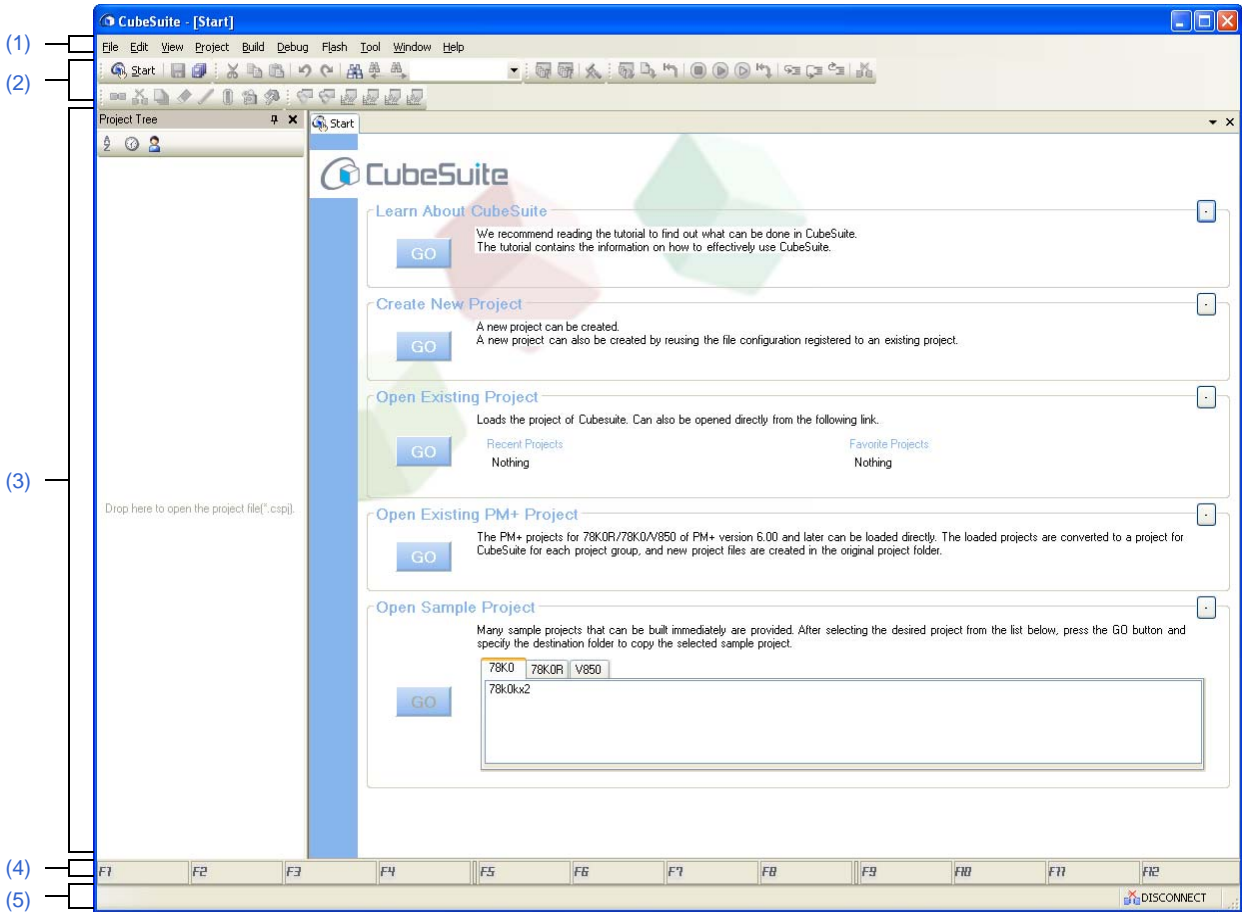
[Function buttons]

Button	Function
Add	Adds the entered license.
Remove	Deletes the selected license.
Close	Closes the License Manager.

Main window

This is the start-up window that opens when CubeSuite is launched.
 In this window, you can control the user program execution and open panels.

Figure A-14. Main Window



The following items are explained here.

- [How to open]
- [Description of each area]

[How to open]

- Select Windows [Start] >> [Programs] >> [NEC Electronics CubeSuite] >> [CubeSuite].

[Description of each area]

(1) Menu bar

This displays common menus.

Contents of each menu can be customized in [User Setting dialog box](#).

(a) [File]

The [File] menu displays file-related menu commands.

New	Display a cascading menu of items to create.
Create New Project...	Closes the current project, and open the Create Project dialog box in order to create a new project. If the currently open project or its files have been modified, then the program will ask if you would like to save your changes.
Open...	Opens the Open File dialog box for opening files and projects.
Add	Displays a cascading menu of items to add.
Add Subproject...	Opens the Add Existing Subproject dialog box to add an existing subproject to the project.
Add New Subproject...	Opens the Create Project dialog box for adding a new subproject to the project.
Add File...	Opens the Add Existing File dialog box, and add the selected file(s) to the project.
Add New File...	Opens the Add File dialog box , create a file with the selected type, and add it to the project. A file extension is assigned to the new file, and it is opened by the application.
Close Project	Closes the currently open project. If the currently open project or its files have been modified, then the program will ask if you would like to save your changes.
Close File	Closes the Editor Panel currently in focus. If the file has been modified, then the program will ask if you would like to save your changes.
Save Project	Saves the settings of the currently open project to a project file.
Save Project As...	Opens the Save Project As dialog box in order to save the settings of the currently open project to a project file with a different name.
Save <i>Target</i>	Saves the file that is currently in focus.
<i>Target</i> Save Option...	Opens the Save Settings dialog box to set the encoding and newline code to use for the file being edited in the Editor panel.
Save <i>Target</i> As...	Opens the <i>Save Target</i> As dialog box in order to save the contents of the file currently in focus with a different name.
Save All	Saves all files being updated in the Editor panel and the project.
Page Setup...	Opens the Page Setup dialog box provided by Windows for printing.
Print....	Opens the Print dialog box provided by Windows in order to print the contents of the active Editor panel.

Recent Files	Displays a list of recently used files in a cascading menu to open those files.
1 Recent File	Uses this item to open the most recently used file.
2 Recent File	Uses this item to open the second most recently used file.
3 Recent File	Uses this item to open the third most recently used file.
4 Recent File	Uses this item to open the fourth most recently used file.
Recent Projects	Displays a list of recently used projects in a cascading menu to open those projects.
1 Recent Project	Uses this item to open the most recently used project.
2 Recent Project	Uses this item to open the second most recently used project.
3 Recent Project	Uses this item to open the third most recently used project.
4 Recent Project	Uses this item to open the fourth most recently used project.
Exit	Exits the application. If there are unsaved source files, or main or subproject files, then a Message dialog box will ask if you want to save them.

(b) [Edit]

The [Edit] menu displays editing-related menu commands.

Undo	Undoes the last action.
Redo	Redoes an undone action.
Cut	Cuts the selection and copy it into the clipboard.
Copy	Copies the selection into the clipboard.
Paste	Pastes the contents of the clipboard.
Delete	Deletes the selection.
Select All	Selects all items.
Rename	Changes the name of the selected item.
Find...	Opens the Search and Replace dialog box , or the Trace Search dialog box if the Trace panel has focus, or the Memory Search dialog box if the Memory panel has focus, and perform the specified search.
Replace...	Opens the Search and Replace dialog box and replace the specified string with another string.
Move To...	Opens the Go to the Location dialog box , and move to the specified location.

(c) [View]

The [View] menu displays panel and other view-related menu commands.

Project Tree	Shows the Project Tree panel , and move the focus to it.
Property	Shows the Property panel , and move the focus to it. The property selected in the Project Tree panel is shown.
Output	Shows the Output panel , and move the focus to it.
Debug Manager	Shows the Debug Manager panel.

Watch	Displays a cascading menu for opening a Watch panel.
Watch1	Shows the Watch 1 panel.
Watch2	Shows the Watch 2 panel.
Watch3	Shows the Watch 3 panel.
Watch4	Shows the Watch 4 panel.
Local Variable	Shows the Local Variables panel.
Call Stack	Shows the Call Stack panel.
Memory	Displays a cascading menu for opening a Memory panel.
Memory1	Shows the Memory 1 panel.
Memory2	Shows the Memory 2 panel.
Memory3	Shows the Memory 3 panel.
Memory4	Shows the Memory 4 panel.
SFR [78K0][78K0R] / IOR [V850]	Shows the SFR panel [78K0][78K0R]/IOR panel [V850].
CPU Register	Shows the CPU Register panel.
Trace	Shows the Trace panel.[IECUBE][Simulator]
Disassemble	Displays a cascading menu for opening a Disassemble panel.
Disassemble1	Shows the Disassemble 1 panel.
Disassemble2	Shows the Disassemble 2 panel.
Disassemble3	Shows the Disassemble 3 panel.
Disassemble4	Shows the Disassemble 4 panel.
Event	Shows the Events panel.
Show Current PC Location	Shows the current PC location in the Editor panel. If there is no source information or source file at the PC location, it is shown in a Disassemble panel. If the Editor or Disassemble panel to show the location is already visible, then the focus will move to that panel.
Forward to Next Cursor Position	Goes to the move destination.
Back to Last Cursor Position	Returns to the last location before moving to the defined location.
Tag Jump	Jumps to the caret line in the editor indicated by the message (file, line, and column).

Save or Restore Docking Layout	Displays commands relating to saving and restoring the docking layout in a cascading menu.
Save Layout 1	Saves the docking layout in slot 1. This layout can be restored via the [Restore Layout 1].
Save Layout 2	Saves the docking layout in slot 2. This layout can be restored via the [Restore Layout 2].
Save Layout 3	Saves the docking layout in slot 3. This layout can be restored via the [Restore Layout 3].
Save Layout 4	Saves the docking layout in slot 4. This layout can be restored via the [Restore Layout 4].
Restore Layout 1	Restores a saved docking layout. This command restores the layout saved via the [Save Layout 1]. If no layout has been saved to this slot, the layout from the first startup is restored.
Restore Layout 2	Restores a saved docking layout. This command restores the layout saved via the [Save Layout 2]. If no layout has been saved to this slot, the layout from the first startup is restored.
Restore Layout 3	Restores a saved docking layout. This command restores the layout saved via the [Save Layout 3]. If no layout has been saved to this slot, the layout from the first startup is restored.
Restore Layout 4	Restores a saved docking layout. This command restores the layout saved via the [Save Layout 4]. If no layout has been saved to this slot, the layout from the first startup is restored.
Reset Layout	Returns all settings relating to the layout of the Main window docking panels to their default values.
Python Console	Shows the Python Console panel .

Remark The layout is saved and restored in each of the following three states.

- Before loading a project
- When a project is loaded and before a debug tool is connected
- When a project is loaded and a debug tool is being connected

(d) [Project]

[Project] menu shows menu items to operate the project.

Create New Project...	Closes the current project and opens the Create Project dialog box to create a new project. Changes the current project or file to the new one. If they have not saved, confirm the user whether to save them.
Open Project...	Closes the current project and opens the Open Project dialog box to open the existing project. Changes the current project or file. If they are not saved, confirm the user whether to save them.

Favorites Projects	Shows the cascading menu to open and add to your favorite project.
1 <i>Favorites Projects</i>	Selects [Favorites Projects] >> [1 Register Favorites Project] to open the added project. If no project is added, "Favorite Project" is shown.
2 <i>Favorites Projects</i>	Selects [Favorites Projects] >> [2 Register Favorites Project] to open the added project. If no project is added, "Favorite Project" is shown.
3 <i>Favorites Projects</i>	Selects [Favorites Projects] >> [3 Register Favorites Project] to open the added project. If no project is added, "Favorite Project" is shown.
4 <i>Favorites Projects</i>	Selects [Favorites Projects] >> [4 Register Favorites Project] to open the added project. If no project is added, "Favorite Project" is shown.
1 <i>Register to Favorites Project</i>	The current project path is added to [1 <i>path</i> in [Favorites Projects]].
2 <i>Register to Favorites Project</i>	The current project path is added to [2 <i>path</i> in [Favorites Projects]].
3 <i>Register to Favorites Project</i>	The current project path is added to [3 <i>path</i> in [Favorites Projects]].
4 <i>Register to Favorites Project</i>	The current project path is added to [4 <i>path</i> in [Favorites Projects]].
Add	Shows the cascading menu to add a project to a subproject.
Add Subproject....	Opens the Add Existing Subproject dialog box to add an existing subproject to a project.
Add New Subproject...	Opens the Create Project dialog box to add a new subproject to a project.
Add File...	Opens the Add Existing File dialog box to add the selected file to a project.
Add New File...	Opens the Add File dialog box to create a file with the selected file type and add to the file to a project. The added file can be opened with the application corresponds to the file extension.
Add New Category	Adds a new Category node to the root of the File node. This allows the category name to be changed. The default category name is "New category". The new category name can be changed to the same name as the existing Category node. Note that this menu is disabled when the build tool is in operation.
Set <i>Project Active Project</i>	Sets the selected project or subproject to an active project.
Close Project	Closes the current project. Changes the current project or file to the new one. If they have not saved, confirm the user whether to save them.
Save Project	Saves the configuration information of the current project to the project file.
Save Project As...	Opens the Save Project As dialog box to save the configuration information of the current project to the project file with another name.
Remove from Project	Removes the selected project or subproject from the project. The subproject files or the file themselves are not deleted from the file system.
Save Project and CubeSuite as Package	Saves a set of the CubeSuite and the project by copying them in a folder.

(e) [Build]

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

Note that only menu items that are displayed when the target project type is a debug-dedicated project (see "[APPENDIX E USING AN EXTERNAL BUILD TOOL](#)") are shown here.

See the "CubeSuite Build" for the information about building for a project other than a debug-dedicated project.

Build Project	Runs a build of the project (see " E.5.1 Run a build "). A build of a subproject is also run when it is added in the project. Note that this menu is disabled when the build tool is in operation.
Rebuild Project	Runs a rebuild of the project (see " E.5.2 Run a rebuild "). A rebuild of a subproject is also run when it is added in the project. Note that this menu is disabled when the build tool is in operation.
Clean Project	Runs a clean of the project (see " E.5.3 Run a clean "). A clean of a subproject is also run when it is added in the project. Note that this menu is disabled when the build tool is in operation.
Rapid build	Toggles the rapid build function (see " E.5.4 Run a rapid build ") between enabled (default) and disabled.
Build <i>active project</i>	Runs a build of the active project (see " E.5.1 Run a build "). If the active project is the main project, a build of its subproject is not run. Note that this menu is disabled when the build tool is in operation.
Rebuild <i>active project</i>	Runs a rebuild of the active project (see " E.5.2 Run a rebuild "). If the active project is the main project, a rebuild of its subproject is not run. Note that this menu is disabled when the build tool is in operation.
Clean <i>active project</i>	Runs a clean of the active project (see " E.5.3 Run a clean "). If the active project is the main project, a clean of its subproject is not run. Note that this menu is disabled when the build tool is in operation.
Stop Build	Cancels the build, rebuild, batch build and clean operation.
Build Mode Settings...	Opens the Build Mode Settings dialog box to modify and add to the build mode.
Batch Build...	Opens the Batch Build dialog box to run a batch build (see " E.5.5 Run a batch build ").
Build Option List	Lists the currently set build options in the Output panel .

(f) [Debug]

See the "CubeSuite Debug" for information about debugging.

(g) [Flash]

See the "CubeSuite Programming" for information about flash programming.

(h) [Tool]

The [Tool] menu displays tool-related menu commands.

Options for <i>Product name</i> ...	Opens the Option dialog box for <i>Product name</i> .
Customize...	Opens the User Setting dialog box .

(i) [Window]

[Window] menu shows menu items to operate the window.

Divide Panel	Divides the active Editor panel horizontally. Only the active Editor panel can be divide. Other panels will not be divide. A panel can be divide a maximum of two times.
Cancel Panel Dividing	Undoes the divide view of the Editor panel.
Close All Divided Panels	Closes all divided panels.
1 <i>Tab name of panel</i>	Moves the focus to 1 <i>tab name of panel</i> .
2 <i>Tab name of panel</i>	Moves the focus to 2 <i>tab name of panel</i> .
3 <i>Tab name of panel</i>	Moves the focus to 3 <i>tab name of panel</i> .
4 <i>Tab name of panel</i>	Moves the focus to 4 <i>tab name of panel</i> .
5 <i>Tab name of panel</i>	Moves the focus to 5 <i>tab name of panel</i> .
6 <i>Tab name of panel</i>	Moves the focus to 6 <i>tab name of panel</i> .
7 <i>Tab name of panel</i>	Moves the focus to 7 <i>tab name of panel</i> .
8 <i>Tab name of panel</i>	Moves the focus to 8 <i>tab name of panel</i> .
9 <i>Tab name of panel</i>	Moves the focus to 9 <i>tab name of panel</i> .
Other Windows...	If 10 or more divide panels are open, open the Other &Windows... dialog box .

(j) [Help]

The [Help] menu displays help-related menu commands.

Help for <i>Product name</i>	Opens the help for <i>product name</i> .
Open Help for <i>Target</i>	Opens the help corresponding to the window, panel, dialog box, error message, or other object that currently has focus.
One Point Advice...	Opens the One Point Advice dialog box .
Tutorial	Opens the help for <i>product name</i> .
Browse Renesas Electronics Microcontrollers Web	Opens our microcontroller website in the web browser.
Contact Info for Tech-support Center...	Shows the Contact Information for Technical Support dialog box .
Check for Updates...	Opens the CubeSuite Update Manager window , and begin checking for updates.
About <i>Product name</i> ...	Opens the Version Information dialog box .












(2) Toolbar

The toolbar shows common command buttons. See the "CubeSuite Build" and "CubeSuite Debug" for buttons relating to building and debugging.

Buttons on the toolbar can be customized in [User Setting dialog box](#). You can also create a new toolbar in the same dialog box.




(a) Start & Save

In the Start & Save, the button group concerning the start button, the file operation, and the edit is displayed.

	Opens the Start panel .
	Saves the file that is currently in focus.
	Saves all files being updated in the Editor panel and the project.
	Cuts the selection and copy it into the clipboard.
	Copies the selection into the clipboard.
	Pastes the contents of the clipboard.
	Undoes the last action.
	Redoes an undone action.
	Opens the Search and Replace dialog box , or the Trace Search dialog box if the Trace panel has focus, or the Memory Search dialog box if the Memory panel has focus, and perform the specified search.
	Searches backward in the panel being searched, using the specified parameters.
	Searches forward in the panel being searched, using the specified parameters.

(b) Build toolbar

Build toolbar shows buttons used in build process.

	Runs a build of the project. A build of a subproject is also run when it is added in the project. Note that this button is disabled when the build tool is in operation.
	Runs a rebuild of the project. A rebuild of a subproject is also run when it is added in the project. Note that this button is disabled when the build tool is in operation.
	Cancels the build, rebuild, batch build and clean operation.

Remark For details on a build, rebuild, clean, or batch build, see the following.

- When the target project type is other than a debug-dedicated project: "CubeSuite Build"
- When the target project type is a debug-dedicated project: ["E.5 Run a Build"](#)

(3) Panel display area

The following panels are displayed in this area.

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

Please see each panel section for the details of the contents of the display.

(4) Function keys bar

Shows the function keys assigned to the currently active window, and the function keys available in the main window.

Hover the mouse cursor over a function key button to display a message about that button.

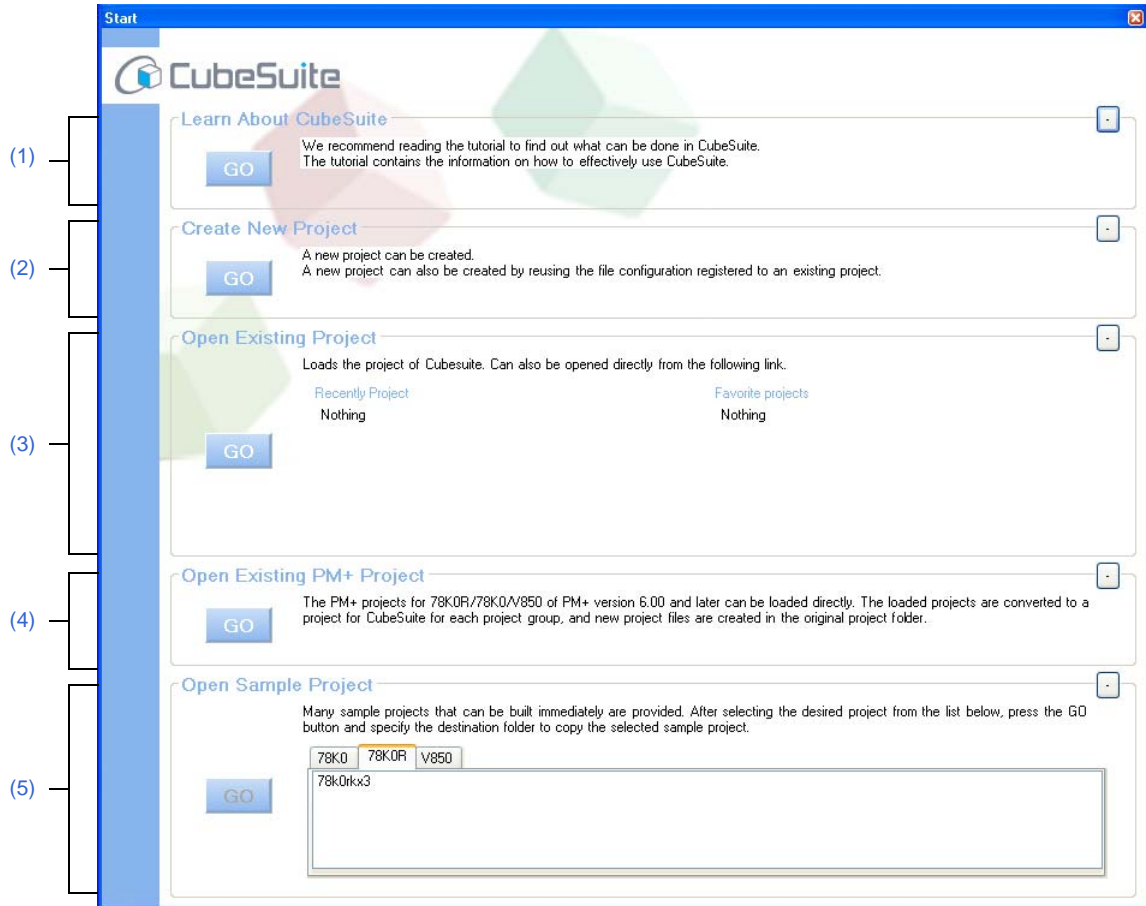
(5) Statusbar

Shows a brief explanation of the currently selected menu item and the various information necessary to debug, etc. See the "CubeSuite Debug" for details.

Start panel

This panel allows you to open a tutorial, access (create/open) a project, and load a sample project, with single-click.


Figure A-15. Start Panel



The following items are explained here.

- [How to open]
- [Description of each area]


[How to open]

- On the toolbar, click the  Start button.
- From the menu bar, select [Window] >> [Start].

[Description of each area]

(1) [Learn About CubeSuite] area

(a) [Learn About CubeSuite]

Click the  button to show a tutorial aimed at improving your understanding of CubeSuite.

(2) [Create New Project] area

(a) [Create New Project]

Click the  button to open the [Create Project dialog box](#), in order to create a new CubeSuite project file.

(3) [Open Existing Project] area


(a) [Recently Project]

This area displays up to four of the most recently used project file names (project names).

(b) [Favorite project]


This area displays up to four of the project file names (project names) in your favorites.

(c) [Open Existing Project]

Click the  button to open the [Open Project dialog box](#), in order to open an existing CubeSuite project file (with ".cspj" extension).

(4) [Open Existing PM+ Project] area


(a) [Open Existing PM+ Project]

Click the  button to open the [Open Project dialog box](#), in order to open an existing PM+ project file (with ".prw" extension).



(5) [Open Sample Project] area

(a) [Open Sample Project]

Load a sample project file (with ".cspj" extension) provided by CubeSuite.

Select the [*Microcontroller*] tab, then select a sample project from the list. Next, click the  button to open the [Browse For Folder dialog box](#) and specify a folder to which to copy the sample project. After copying the sample to the specified folder, the sample project is loaded.

(6) Button

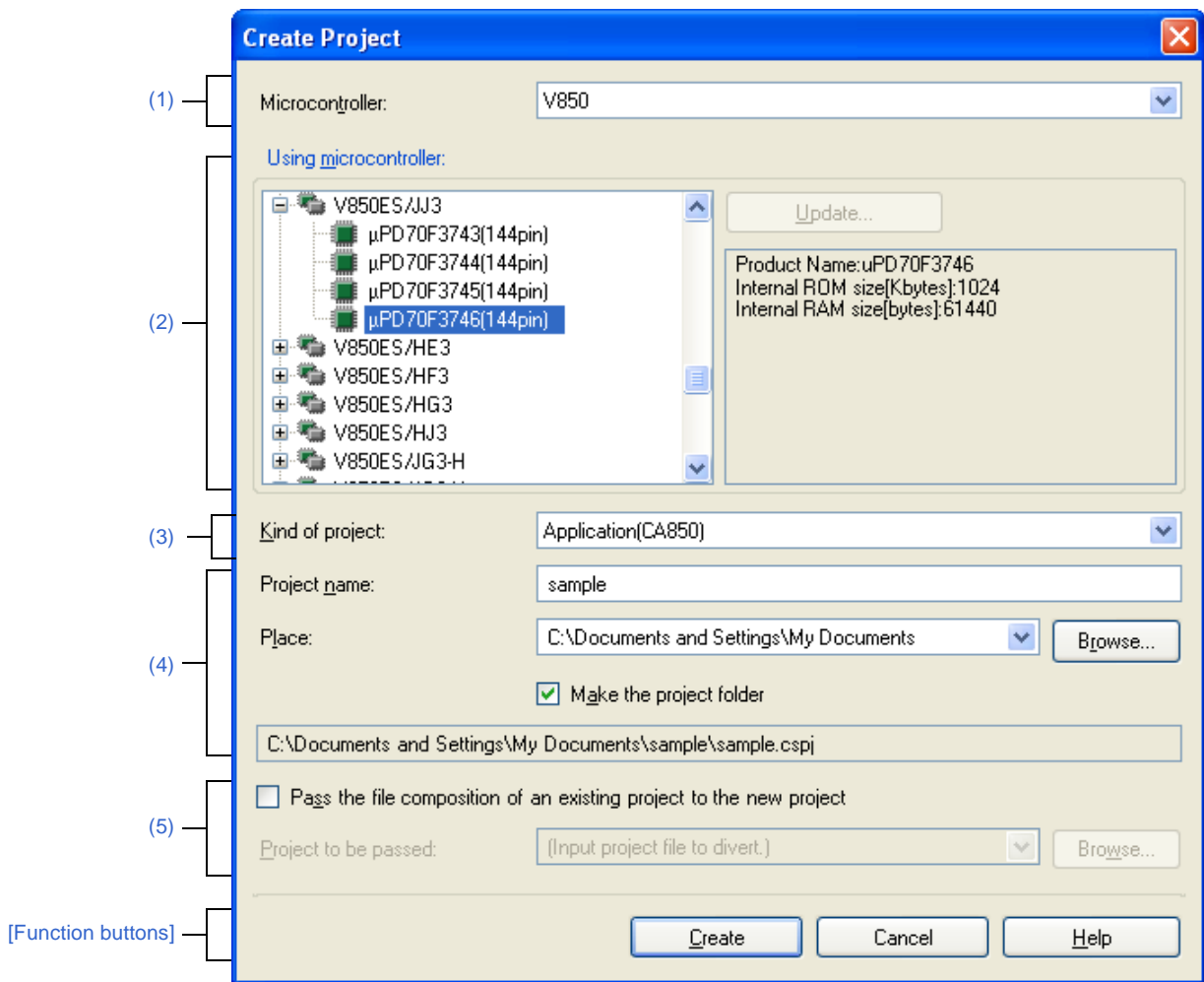
	Minimizes the area. Only the title text and minimize buttons will be shown.
	Restores a minimized area.

Create Project dialog box

This dialog box is used to create new projects or subprojects.

Caution A subproject cannot be added to another subproject.

Figure A-16. Create Project Dialog Box



Remark When a subproject is created, the title bar says "Create Subproject".

The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]**(1) When creating a main project**

- Click the [Start] button in the toolbar to open the [Start panel](#) and then click the [GO] button in the [Create New Project] area.
- From the [File] menu, select [New] >> [Create New Project...].
- From the [Project] menu, select [Create New Project...].

(2) When creating a subproject

- From the [Project] menu, select [Add] >> [Add New Subproject...].
- On the [Project Tree panel](#), select project or subproject, and then select [Add] >> [Add New Subproject...] from the context menu.

[Description of each area]**(1) [Microcontroller]**

Select the microcontroller type to use in the project.

You can select the item below.

- V850
- 78K0R
- 78K0

(2) [Using microcontroller] area**(a) List of microcontrollers**

Select a microcontroller to use in the project.

The microcontrollers that correspond to the microcontroller type selected in [\[Microcontroller\]](#) are listed and categorized by their nicknames.

The following information on the selected microcontroller that is used in the project is shown in the box at right.

- Product Name
- Internal ROM size [K bytes]
- Internal RAM size [K bytes]

(b) Button

Update...	<p>Displays the CubeSuite Update Manager window.</p> <p>You can use the CubeSuite Update Manager window to check for microcontroller information updates via the network.</p> <p>This button becomes enabled if you register the license of CubeSuite.</p>
-----------	--

Remark When the microcontroller to be used does not appear in the list, you can search for the microcontroller via the net work with the [CubeSuite Update Manager window](#) which is opened by clicking the [Update...] button.

Note that the [Update...] button is invalid if you have not registered the license of CubeSuite.

(3) [Kind of project]

Select the project type to create on [Kind of project].

You can select the item below.

- Application(CA850/CX/CA78K0/CA78K0R)

Select this to generate the ROMization module file [V850], load module file, and hex file, by using the build tool provided by CubeSuite.

- Library(CA850/CX/CA78K0/CA78K0R)

Select this to generate a library file for a user library, by using the build tool provided by CubeSuite.

- Debug Only

Select this to debug a load module file or hex file generated with a build tool other than the one provided by CubeSuite (i.e. creates a debug-dedicated project).

See "[APPENDIX E USING AN EXTERNAL BUILD TOOL](#)" for details on how to create and use the debug-dedicated project.

(4) Project Creation area

The project name to create and the location of the project file are designated.

(a) [Project name]

Directly enter the name of the project to create.

The entered project name is followed by the extension, ".cspj" and this forms the project file name.

Remark When the subproject is created, the entered project name is followed by the extension, ".cssp" and forms the subproject name.

(b) [Place]

Designate the location to create the project file by directly entering it with absolute path or selecting from the [Browse...] button.

The last designated location is shown in the box. "C:\Documents and Settings*User name*\My Documents" is shown by default when the project is created for the first time.

(c) Button

Browse...	<p>Opens the Browse For Folder dialog box.</p> <p>When a folder is selected, an absolute path of the folder is shown in [Place].</p>
-----------	--

(d) [Make the project folder]

Use this check box to select whether to create a folder with the project name under the location specified in [\[Place\]](#).



The check box is selected by default.

(e) Project file path

The full path of the project file (specified via [\[Project name\]](#) and [\[Place\]](#)) is shown.

Remarks 1. The number of characters that can be entered in [\[Project name\]](#) and [\[Place\]](#) is up to 259 both for the path name and file name together. When the input violates any restriction, the following messages are shown in the tooltip in [\[Project name\]](#).

Message	Description
File name with the path is too long. Shorten within 259 characters.	The file name with the path is more than 259 characters.
The designated path includes the folder that does not exist.	The path includes the folder that does not exist.
Invalid file name or path name. The characters, \, /, :, *, ?, ", <, >, , cannot be used.	The file name with the invalid path is designated. The characters, \, /, :, *, ?, ", <, >, , cannot be used for the folder name.

- When the project name and the path name is too long to be shown in each text area,  is displayed. The absolute path pops up when the mouse cursor is hovered over .

(5) Pass project area

Specify the name of the source project to pass, and the location in which to create the new project file.

(a) [Pass the file composition of an existing project to the new project]

Check this check box when reusing the file composition of the existing project (main or subproject) to create a new project.

The check box is unchecked by default.

The category of the file can be also reused.

(b) [Project to be passed]

Designate the name of the source project when diverting the file composition of the existing project to create a new project.

Designate the name of the source project by directly entering it with absolute path or selecting with the [Open Project dialog box](#) after pressing the [Browse...] button.

The last designated location is shown in the box. "C:\Documents and Settings\User name\My Documents" is the default location when the file is created for the first time.

This field is enabled only when the [\[Pass the file composition of an existing project to the new project\]](#) check box is checked.



Caution You cannot designate the PM+ project file.

(c) Button

Browse...	<p>Open Project dialog box appears.</p> <p>When a project file is selected, an absolute path of the project file is shown in [Project to be passed].</p>
-----------	--

- Remarks 1.** When the version of the build tool used in the source project is different from the version of the build tool in the project to be created, it is automatically diverted (except for a debug-dedicated project).
- When the build tool for the project to be created does not support a node of the source project, the node will be displayed as a normal Category node on the project tree (e.g., Startup node or Download files node etc.).
 - Up to 259 characters can be entered in [\[Project to be passed\]](#). When the input violates any restriction, the following messages are shown.

Message	Description
File name with the path is too long. Shorten within 259 characters.	The file name with the path is more than 259 characters.
The designated path includes the folder that does not exist.	The path includes the folder that does not exist.
Invalid file name or path name. The characters, \, /, :, *, ?, ", <, >, , cannot be used.	The file name with the invalid path is designated. The characters, \, /, :, *, ?, ", <, >, , cannot be used for the folder name.

- When the path name is too long to be shown in the text area,  is displayed. The absolute path pops up when the mouse cursor is hovered over .

[Function buttons]

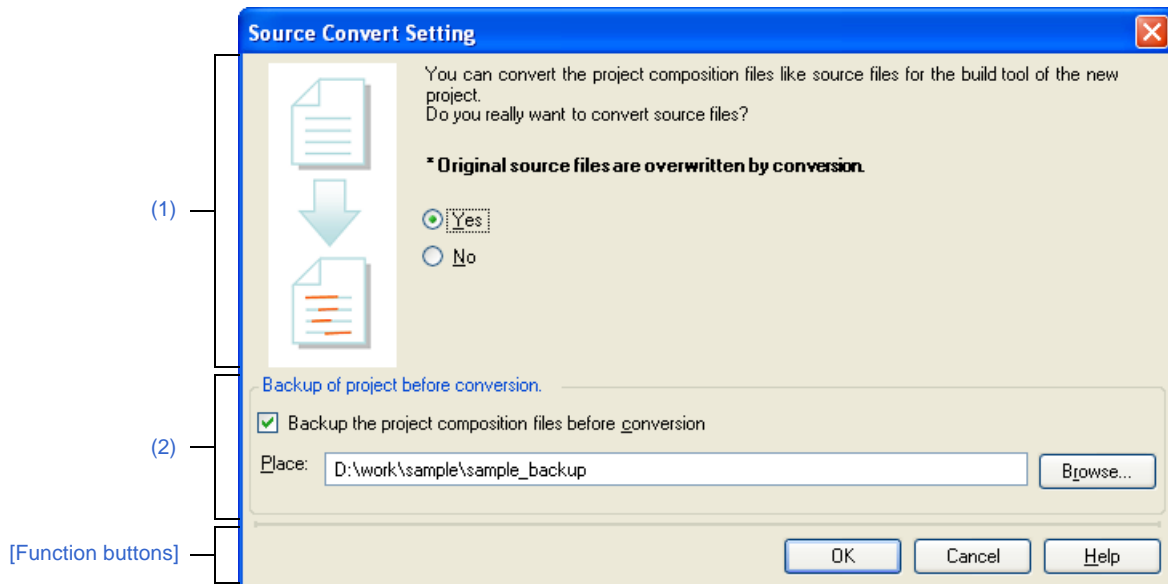
Button	Function
Create	Creates a project according to the designated condition and closes the dialog box. When the [Pass the file composition of an existing project to the new project] check box is checked, creates a project by diverting the file composition of the project (main or sub-project) designated in [Project to be passed] . If the build tool of the source project is CA850, and the build tool of the project to create is CX, opens the Source Convert Setting dialog box [CX] to select whether to convert the source files and the like.
Cancel	Cancels the designated condition and closes the dialog box.
Help	Displays the help of this dialog box.

Source Convert Setting dialog box [CX]

This dialog box configures the composition files of the source project (the source files and the like) to convert them for the build tool of the project to be created.

Caution This dialog box is only displayed if the build tool of the source project is CA850, and that of the project to be created is CX.

Figure A-17. Source Convert Setting Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Create Project dialog box](#), set the build tool of the source project to CA850, and that of the project to create to CX, and then click the [Create] button.

[Description of each area]

(1) Source convert selection area

Select whether to convert the source files when creating the project.

- When [Yes] is selected
The source files are converted.
- When [No] is selected
The source files are not converted.

(2) [Backup of project before conversion.] area

Make settings relating to backing up the source project.

(a) [Backup the project composition files before conversion]

Select this check box if you wish to backup the entire source project and its source files as a single bundle.

(b) [Place]

Specify the location in which to save the entire source project and its source files.

Enter the absolute path directly, or click the [Browse...] button, and select the location via the [Browse For Folder dialog box](#).

By default, "project-folder-of-project-being-created_backup" is displayed. If there is already a folder with the same name, a number (2, 3, ...) will be added to the folder name.



This field is enabled only when the [\[Backup the project composition files before conversion\]](#) check box is checked.

(c) Button

Browse...	Browse For Folder dialog box appears. When a folder is selected, the path is shown in the text box.
-----------	--

Remarks 1. Up to 247 characters can be entered in [\[Place\]](#). When the input violates any restriction, the following messages are shown.

Message	Description
The folder name including the path is too long. Make it within 247 characters.	The folder name is more than 247 characters.
The folder name is invalid. The following characters cannot be used: :, *, ?, ", <, >,	The folder name with the invalid path is designated. The characters, :, *, ?, ", <, >, , cannot be used for the file name and folder name.
The folder name is invalid. Specify a folder other than the diverted project folder.	A folder inside the source project folder is specified.

2. When the path name is too long to be shown in the text area,  is displayed. The absolute path pops up when the mouse cursor is hovered over .

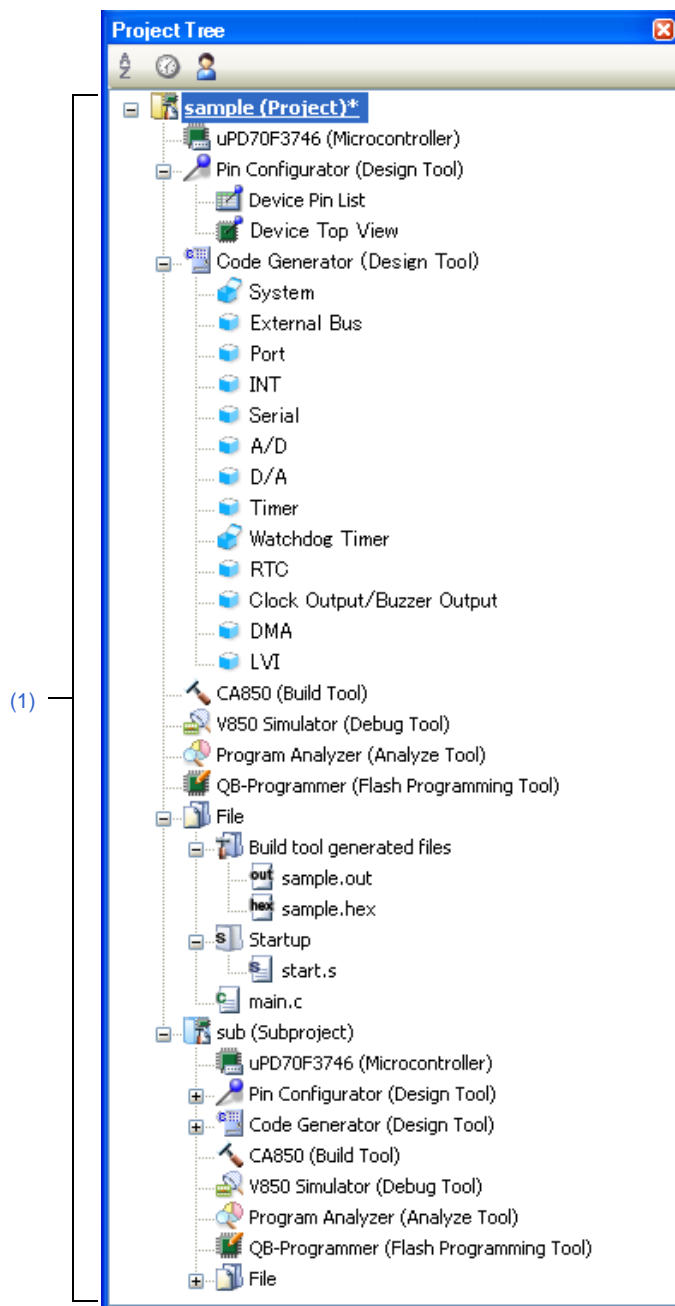
[Function buttons]

Button	Function
OK	Closes this dialog box and creates a project according to the designated condition.
Cancel	Cancels the designated condition and closes the dialog box.
Help	Displays the help of this dialog box.

Project Tree panel

This panel is used to display components of the microcontroller, build tool, and source file of the project in tree view.

Figure A-18. Project Tree Panel



The following items are explained here.

- [\[How to open\]](#)
- [\[Description of each area\]](#)
- [\[Toolbar\]](#)
- [\[\[Edit\] menu \(Project Tree panel-dedicated items\)\]](#)
- [\[\[Help\] menu \(Project Tree panel-dedicated items\)\]](#)
- [\[Context menu\]](#)

[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.

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

Node	Explanation
<i>Project name</i> (Project) (Subsequently called "Project node")	The project's name.
<i>Microcontroller name</i> (Microcontroller) (Hereafter referred to as "Microcontroller node")	The microcontroller used in the project.
<i>Design Tool name</i> (Design Tool) (Hereafter referred to as "Design Tool node")	The design tool (pin configurator, code generator, etc.) used. Note that Code Generator (Design Tool) node is not shown when the project type is a debug-dedicated project.
<i>Build tool name</i> (Build tool) (Hereafter referred to as "Build Tool node")	The build tool (compiler, assembler, etc.) used. When the project type is a debug-dedicated project, "None" is shown as <i>build tool name</i> .
<i>Debug tool name</i> (Debug tool) (Hereafter referred to as "Debug Tool node")	The debug tool (in-circuit emulator, simulator, etc.) used.
Program Analyzer (Analyze Tool) (Hereafter referred to as "Analyze Tool node")	The analyze tool used. Note that this node is not shown when the project type is a debug-dedicated project.
QB-Programmer (Flash Programming Tool) (Hereafter referred to as "Flash Programming Tool node")	The flash programming tool used.
File (Hereafter referred to as "File node")	Files registered to the project are displayed directly below the File node.
Download files (Hereafter referred to as "Download files node")	This is a node for adding download files to the project. Note that this node is shown only when the project type is a debug-dedicated project.
Build tool generated files (Hereafter referred to as "Build tool generated files node")	This node is created during a build. Files created by the build tools are displayed directly below the node (except for object files). Note that this node is not shown when the project type is a debug-dedicated project.
Startup (Hereafter referred to as "Startup node")	This is a node for adding other than standard startup files to the project. This node is always shown under the File node. Note that this node is not shown when the project type is a debug-dedicated project.
<i>Category name</i> (Hereafter referred to as "Category node")	These user-defined categories are used to classify files into modules. Note that this node is always shown under the files node.

Node	Explanation
<i>Subproject name</i> (Subproject) (Hereafter referred to as "Subproject node")	Subprojects added to the project.

- Remarks 1.** Only the tools corresponding to the microcontroller in use are shown.
- 2.** When more than one components are selected, only the tab that is common to all the components is displayed.
When more than one files are selected and their common properties are different, that field is left blank.
- 3.** See "[APPENDIX E USING AN EXTERNAL BUILD TOOL](#)" for details on a debug-dedicated project.

This area has the following functions.

(a) Import project files

If you drag the project file from the windows explorer and drop in this area, you can open the project in Main Window.

(b) Add subprojects

You can add subprojects by one of the following procedure.

<1> Add an existing subproject

- Select Project node or Subproject node. Then select [Add] >> [Add Subproject...] in [File] or [Project] menu. [Add Existing Subproject dialog box](#) appears. Select subproject files to add.
- Select [Add] >> [Add Subproject...] in the context menu of Project node or Subproject node. [Add Existing Subproject dialog box](#) appears. Select subproject files to add.

<2> Add new subproject

- Select Project node or Subproject node. Then select [Add] >> [Add New Subproject...] in [File] or [Project] menu. [Create Project dialog box](#) appears. Create a new subproject file.
- Select [Add] >> [Add New Subproject...] in the context menu of Project node or Subproject node. [Create Project dialog box](#) appears. Create a new subproject file.

(c) Remove a subproject from a project

You can remove a subproject from a project by one of the following procedure.

- Select the subproject that you want to delete, then select [Remove from Project] in [Project] menu.
- Select the subproject that you want to delete, then select [Remove from Project] in the context menu.

Caution If the selected subproject is the active project, then it cannot be removed from the project.

(d) Move subprojects

You can move subprojects by the following procedure.

- Drag the subproject you want to move, then drop it in the destination.

Remark You can run multiple CubeSuite and drop the subproject to a different project. In this case, the dropped subproject is copied, not moved.

(e) Select multiple nodes

You can select multiple nodes in sequence by the following procedure.

- Click or place the cursor on the start node then click with pressing the [Shift] key or place the cursor on the end node.

You can select multiple nodes one by one by the following procedures.

- Click on the node with pressing the [Ctrl] key.
- Place the cursor key on the node with pressing the [Ctrl] key then press [Space] key.

(f) Running the editor

The file with the specific extension is opened in the [Editor panel](#). When an external editor is set to be used in the [Option dialog box](#), the file is opened with the external editor that has been set. Other files are opened with the applications associated by the host OS.

Caution The file with the extension that is not associated with the host OS is not displayed.

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

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










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

- C source file (*.c)
- Assembler source file (*.asm, *.s)
- Header file (*.h, *.inc)
- Symbol information file (*.sfg)
- Link directive file (*.dir, *.dr)
- Link map file (*.map)
- Hex file (*.hex)
- Text file (*.txt)

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

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

[Toolbar]

	<p>Sorts files and Category nodes in Project tree area in order of their names.</p> <ul style="list-style-type: none">  : Ascending order  : Descending order  : Ascending order
	<p>Sorts files in Project tree area in order of the time stamp.</p> <ul style="list-style-type: none">  : Descending order  : Ascending order  : Descending order
	<p>Sorts files and Category nodes in Project tree area in order of the user definition (default). Display order is changed by dragging and dropping the file and Category node.</p>

[[Edit] menu (Project Tree panel-dedicated items)]

Copy	<p>Copies the selected file or Category node to the clip board.</p> <p>While editing the file name or the category name, the characters of the selection are copied to the clip board.</p> <p>Note that this menu is only enabled when the file or the Category node is selected.</p>
Paste	<p>Inserts the contents of the clip board to the root of the node that is selected on the Project Tree.</p> <p>While editing the file name or the category name, insert the contents of the clip board.</p> <p>Note that this menu is only enabled when the file or the Category node exists. If a project with the same contents is already on the clipboard, however, then if multiple files or Category nodes are selected, and the build tool is running, then it will be invalid.</p>
Delete	<p>Deletes the selected files and Category nodes.</p> <p>When the file is selected, it is deleted from the file system.</p> <p>When the selected file is also added to other project, it is not deleted from the file system but removed from the project.</p> <p>While editing the file name or the category name, the characters of the selection are deleted.</p> <p>Note that this menu is only enabled when the file or the Category node is selected.</p>
Rename	<p>You can rename the selected project, subproject, file, and Category node. Press [Enter] key to confirm the rename. Press [ESC] key to cancel.</p> <p>When the file is selected, the actual file name is also changed.</p> <p>When the selected file is added to other project, those file names are also changed.</p> <p>Note that this menu is only enabled when the project, subproject, file, and Category node is selected. Note that rename is disabled when the build tool is operating.</p>

[[Help] menu (Project Tree panel-dedicated items)]

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

[Context menu]

(1) When a Project node is selected

Build <i>Actively project</i>	<p>Builds the active project.</p> <p>If the active project is the main project, its subproject is not built.</p> <p>Note that this menu is disabled when the build tool is in operation.</p>
Rebuild <i>Actively project</i>	<p>Rebuilds the active project.</p> <p>If the active project is the main project, its subproject is not rebuilt.</p> <p>Note that this menu is disabled when the build tool is in operation.</p>
Clean <i>Actively project</i>	<p>Cleans the active project.</p> <p>If the active project is the main project, its subproject is not cleaned.</p> <p>Note that this menu is disabled when the build tool is in operation.</p>
Open Folder with Explorer	Opens the folder that contains the project file of the selected project with Explorer.

Add	Shows the cascading menu to add subprojects and files to the project.
Add Subproject...	Opens the Add Existing Subproject dialog box to add the selected subproject to a project.
Add New Subproject...	Opens the Create Project dialog box to add the created subproject to a project.
Add File...	Opens the Add Existing File dialog box to add the selected file to a project.
Add New File...	Opens the Add File dialog box to create a file with the selected file type and add to a project. The added file can be opened with the application corresponds to the file extension.
Add New Category	Adds a new Category node to the root of the File node. This allows the category name to be changed. The default category name is "New category". The new category name can be changed to the same name as the existing Category node. Note that this menu is disabled when the build tool is in operation.
Set <i>Selected project</i> Active Project	Sets the selected project to an active project.
Save Project and CubeSuite as Package	Saves a set of the CubeSuite and the project by copying them in a folder.
Paste	This menu is always disabled.
Rename	You can rename the selected project.
Property	Displays the selected project's property in Property panel .

(2) When a Subproject node is selected

Build <i>Actively project</i>	Builds the active project. Note that this menu is disabled when the build tool is in operation.
Rebuild <i>Actively project</i>	Rebuilds the active project. Note that this menu is disabled when the build tool is in operation.
Clean <i>Actively project</i>	Cleans the active project. Note that this menu is disabled when the build tool is in operation.
Open Folder with Explorer	Opens the folder that contains the subproject file of the selected subproject with Explorer.

Add	Shows the cascading menu to add subprojects, files, and Category nodes to the project.
Add Subproject...	Opens the Add Existing Subproject dialog box to add the selected subproject to a project. The subproject cannot be added to another subproject.
Add New Subproject...	Opens the Create Project dialog box to add the created subproject to a project. The subproject cannot be added to another subproject.
Add File...	Opens Add Existing File dialog box to add the selected file to a project.
Add New File...	Opens the Add File dialog box to create a file with the selected file type and add to a project. The added file can be opened with the application corresponds to the file extension.
Add New Category	Adds a new Category node to the root of the File node. This allows the category name to be changed. The default category name is "New category". The new category name can be changed to the same name as the existing Category node. Note that this menu is disabled when the build tool is in operation.
Set <i>Selected subproject</i> Active Project	Sets the selected subproject to an active project.
Remove from Project	Removes the selected subproject from the project. The subproject file itself is not deleted from the file system with this operation. When the selected subproject is the active project, it cannot be removed from the project. Note that this menu is disabled when the build tool is in operation.
Paste	This menu is always disabled.
Rename	You can rename the selected subproject.
Property	Displays the selected subproject's property in Property panel .

(3) When a Microcontroller node is selected

Property	Displays the selected microcontroller's property in Property panel .
----------	--

(4) When a Build Tool node is selected

Only menu items that are displayed when a Build Tool node for a debug-dedicated project (see "[APPENDIX E USING AN EXTERNAL BUILD TOOL](#)") is selected are shown here. See the "CubeSuite Build" for the information about the Build Tool node for a project other than a debug-dedicated project.

Build Project	Runs a build the selected project (main project or subproject). A build of a subproject is also run when it is added in the project. Note that this menu is disabled when the build tool is in operation.
Rebuild Project	Runs a rebuild the selected project (main project or subproject). A rebuild of a subproject is also run when it is added in the project. Note that this menu is disabled when the build tool is in operation.
Clean Project	Runs a clean of the selected project (main project or subproject). A clean of a subproject is also run when it is added in the project. Note that this menu is disabled when the build tool is in operation.

Set to Default Build Option for Project	Sets the current build options as the standard for the project. When a subproject is added, its setting is not made. When the build option that is different from the standard option is set, its property is displayed in boldface.
Property	Displays the selected build tool's property on the Property panel .

(5) When a Download file node is selected

This node is displayed only for a debug-dedicated project (see "[APPENDIX E USING AN EXTERNAL BUILD TOOL](#)").

Add	Shows the cascading menu to add download files to the project.
Add File...	Opens the Add Existing File dialog box to add the selected file to the project as a download file.
Add New File...	Opens the Add File dialog box to create a file with the selected file type and add to the file to the project as a download file.
Add New Category	Adds a new Category node directly below this node. This allows the category name to be changed. Up to 200 characters can be specified. The default category name is "New category". You can also add a Category node with the same name as the existing Category node. This menu is disabled when the build tool is in operation and when categories are nested 20 levels.
Property	Displays this node's property on the Property panel .

(6) When the File node is selected

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

(7) When a file is selected

Only menu items that are displayed when a file for a debug-dedicated project (see "[APPENDIX E USING AN EXTERNAL BUILD TOOL](#)") is selected are shown here. See the "CubeSuite Build" for the information about a file for a project other than a debug-dedicated project.

Open	Opens the selected file with the application corresponds to the file extension (see " (f) Running the editor "). Note that this menu is disabled when multiple files are selected.
Open with Internal Editor...	Opens the selected file with the Editor panel . Note that this menu is disabled when multiple files are selected.
Open with Selected Application...	Opens the Open with Program dialog box to open the selected file with the designated application. Note that this menu is disabled when multiple files are selected.
Open Folder with Explorer	Opens the folder that contains the selected file with Explorer.
Add	Shows the cascading menu to add files and Category nodes to the project.
Add File...	Opens the Add Existing File dialog box to add the selected file to the project. The file is added to the same level as the selected file.
Add New File...	Opens the Add File dialog box to create a file with the selected file type and add to the file to the project. The file is added to the same level as the selected file. The added file can be opened with the application corresponds to the file extension.
Add New Category	Adds a new Category node at the same level as the selected file. You can rename the category. Up to 200 characters can be specified. The default category name is "New category". You can also add a Category node with the same name as the existing Category node. This menu is disabled when the build tool is in operation and when categories are nested 20 levels.
Remove from Project	Removes the selected file from the project. The file itself is not deleted from the file system. Note that this menu is disabled when the build tool is in operation.
Copy	Copies the selected file to the clipboard. When the file name is in editing, the selected characters are copied to the clipboard.
Paste	This menu is always disabled.
Rename	You can rename the selected file. The actual file is also renamed. When the selected file is added to other project, that file name is also changed.
Property	Displays the selected file's property on the Property panel .

(8) When a Category node is selected

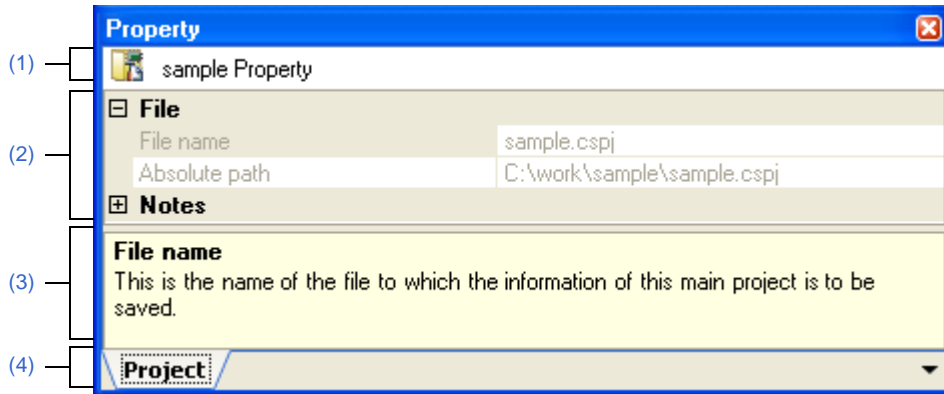
Add	Shows the cascading menu to add files and Category nodes to the project.
Add File...	Opens the Add Existing File dialog box to add the selected file to the project. The file is added directly below this node. The added file can be opened with the application corresponds to the file extension.
Add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

- Remarks 1.** When more than one components are selected, the context menu of the last selected component is displayed.
- 2.** The context menu of the currently selected component is displayed in the blank area under the project tree.

Property panel

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

Figure A-19. Property Panel



The following items are explained here.

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

[How to open]

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

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



[Description of each area]


(1) Name for the selected node area

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

(2) Detailed information display/change area

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

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

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

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

(3) Property description area

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

(4) Tab selection area

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

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

(a) Project node is selected in Project Tree panel

Project node	- [Project] tab
Subproject node	- [Subproject] tab
Microcontroller node	- [Microcontroller Information] tab
Design Tool node	- [Pin Configurator Information] tab - [Top View Setting] tab - [Generation] tab - [Macro Setting] tab See the "CubeSuite Design" for details about this tab.
Build Tool node (Except for a debug-dedicated project ^{Note 1})	- [Common Options] tab - [Compile Options] tab - [Assemble Options] tab - [Link Options] tab - [Object Convert Options] tab [CA78K0][CA78K0R] - [Create Library Options] tab [CA78K0][CA78K0R][CX] - [Variables/Functions Relocation Options] tab [CA78K0R] - [Memory Bank Relocation Options] tab [CA78K0] - [ROMization Process Options] tab [CA850] - [Hex Convert Options] tab [CA850] - [Archive Options] tab [CA850] - [Section File Generate Options] tab [CA850] - [Dump Options] tab [CA850] - [Cross Reference Options] tab [CA850] - [Memory Layout Visualization Options] tab [CA850] - [ROMize Options] tab [CX] - [Hex Output Options] tab [CX] See the "CubeSuite Build" for details about this tab.

Build Tool node (For a Debug-dedicated project ^{Note 1})	- [Build Options] tab
Debug Tool node	- [Connect Settings] tab - [Debug Tool Settings] tab - [Flash Self Emulation Settings] tab [IECUBE] - [DataFlash Emulation Settings] tab [IECUBE[78K0R]] [IECUBE[V850]] - [Download File Settings] tab - [Flash Options Settings] tab [MINICUBE[V850E2M]] - [Hook Transaction Settings] tab See the "CubeSuite Debug" for details about this tab.
Analyze Tool node	- [Settings] tab - See the "CubeSuite Analysis" for details about this tab.
File node	- [Build Settings] tab (for C source file, assembler source file, object module file, link directive file, symbol information file [CX], and library file) - [Individual Compile Options] tab (for C source file) - [Individual Assemble Options] tab (for assembler source ^{Note 2}) - [File Information] tab See the "CubeSuite Build" for details about this tab.
Category node	- [Category information] tab See the "CubeSuite Build" for details about this tab.

Notes 1. For details on the debug-dedicated project, see "[APPENDIX E USING AN EXTERNAL BUILD TOOL](#)".

2. This tab is also displayed in the following case:

- [CA78K0][CA78K0R]When [Yes] is selected in [Output assemble file] property in [Assembly File] category in [Individual Compile Options] tab.

- [CA850]

When [Yes] is selected in [Set individual compile option] property in [Build] category in [Build Settings] tab.

- [CA850]

When a C source file is selected, and on the [Individual Compiler Options] tab, under the [Output Files] category, the [Output assembly files] property is set to [Yes (-Fs)].

(b) Click the [Generate Code] button in the Code Generator panel

- [Macro Setting] tab See the "CubeSuite Design" for details about this tab.

(c) On the Code Generator Preview panel, select file

- [File Setting] tab See the "CubeSuite Design" for details about this tab.
--

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

[Dialog boxes opened from the Property panel]

The following dialog boxes are opened from the Property panel.

- Character String Input dialog box
See the "[Character String Input dialog box](#) (CubeSuite Start)", "CubeSuite Build" or "CubeSuite Debug" for details.
- Text Edit dialog box
See the "[Text Edit dialog box](#) (CubeSuite Start)", "CubeSuite Build" or "CubeSuite Debug" for details.
- Path Edit dialog box
See the "CubeSuite Build" for details.

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

Undo	Undoes any property changes being done.
Cut	Cuts the selected text to the clip board while editing the property.
Copy	Copies the selected text in the property to the clip board.
Paste	Pastes the contents of the clip board to the property while editing the property.
Delete	Deletes the selected text while editing the property.
Select All	Selects all the text in the selected property while editing the property.

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

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

[Context menu]

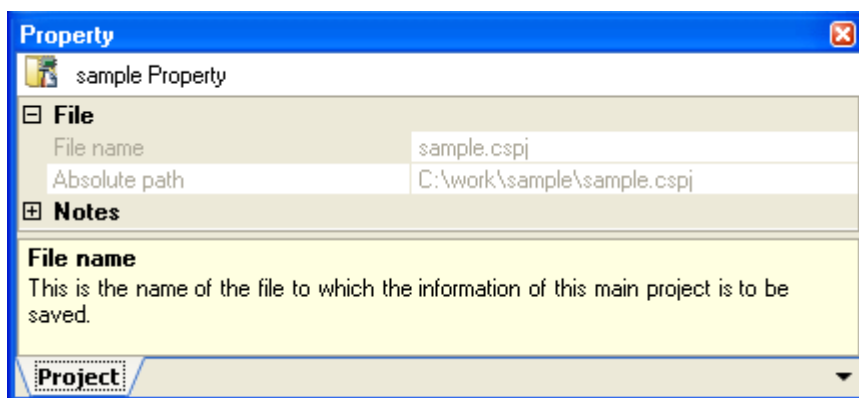
Undo	Undoes any property changes being done.
Cut	Cuts the selected text to the clip board while editing the property.
Copy	Copies the selected text in the property to the clip board.
Paste	Pastes the contents of the clip board to the property while editing the property.
Delete	Deletes the selected text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

[Project] tab

This tab shows the detailed information on projects (main project) categorized by the following and the configuration can be changed.

- (1) [File]
- (2) [Notes]

Figure A-20. Property Panel : [Project] Tab



[Description of each category]

(1) [File]

The detailed information on files are displayed.

File name	Displays the file name of the file to save the information on the main project.	
	Default	<i>Name of the main project file</i>
	How to change	Changes not allowed
Absolute path	Displays the absolute path of the main project file to save the information on them.	
	Default	<i>Absolute name of main projects</i>
	How to change	Changes not allowed

(2) [Notes]

The detailed information on records is displayed and their configuration can be changed.

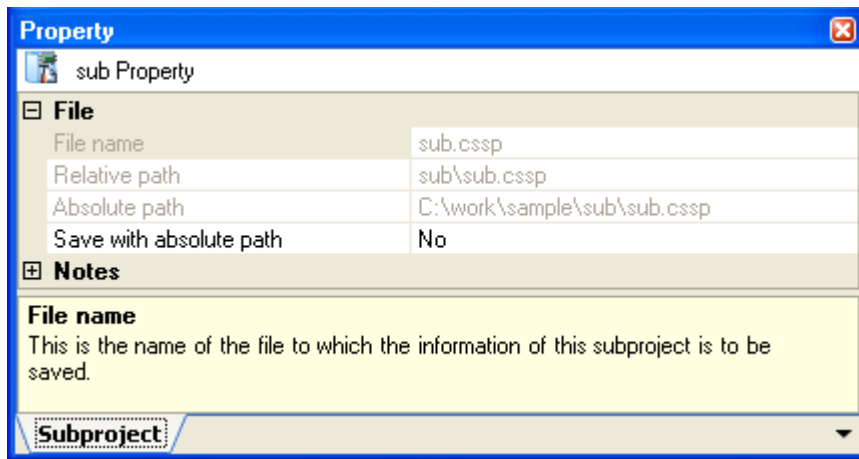
Memo	Add memos to main projects. Memo is added one item in one line. The added memos are displayed as subproperty.	
	Default	Memo [<i>Number of items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use a text box directly enter the text.
	Restriction	Up to 256 characters Up to 256 characters are allowed.

[Subproject] tab

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

- (1) [File]
- (2) [Notes]

Figure A-21. Property Panel : [Subproject] Tab



[Description of each category]

(1) [File]

The detailed information on files are displayed.

File name	Displays the file name of the file to save the information on the subproject.	
	Default	<i>Name of the subproject file</i>
	How to change	Changes not allowed
Relative path	Displays the relative path of the subproject file to which subproject information is saved, from the location of the main project. This property is shown only when the subproject is selected in Project Tree panel . Note that if the relative path does not exist (such as the case that the main project and subproject are in the different drive), it is not displayed.	
	Default	<i>Relative path from the main project</i>
	How to change	Changes not allowed
Absolute path	Displays the absolute path of the subproject file to save the information on them.	
	Default	<i>Absolute name of subprojects.</i>
	How to change	Changes not allowed

Save with absolute path	Select whether to save the subproject information to a subproject file using the absolute path.		
	Default	No	
	How to change	Select a value from the drop-down list.	
	Ristriction	Yse	Saves the file using the absolute path.
No		Saves the file using the relative path.	

(2) [Notes]

The detailed information on records is displayed and their configuration can be changed.

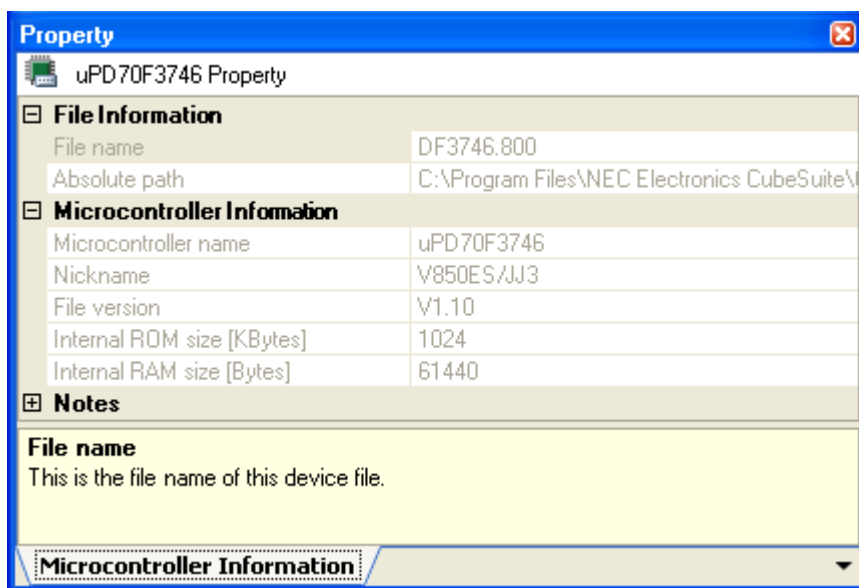
Memo	Add memos to subprojects. Memo is added one item in one line. The added memos are displayed as subproperty.		
	Default	Memo [<i>Number of items</i>]	
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use a text box directly enter the text.	
	Ristriction	Up to 256 characters Up to 256 characters are allowed.	

[Microcontroller Information] tab

This tab shows the detailed information on microcontrollers configured in the project, categorized by the following and the configuration can be changed.

- (1) [File Information]
- (2) [Microcontroller Information]
- (3) [Notes]

Figure A-22. Property Panel : [Microcontroller Information] Tab



[Description of each category]

(1) [File Information]

The detailed information on the file is shown.

File name	Displays device file name.	
	Default	<i>Device file name</i>
	How to change	Changes not allowed
Absolute path	Displays the absolute path of the device file.	
	Default	<i>The absolute path of the device file</i>
	How to change	Changes not allowed

(2) [Microcontroller Information]

The detailed information on microcontrollers is displayed and their configuration can be changed.

Microcontroller name	Displays microcontrollers to use in the project or the subproject.	
	Default	<i>Microcontrollers used in the main project or the subproject</i>
	How to change	Changes not allowed
Nickname	Displays the nickname of the microcontroller to use.	
	Default	<i>The nickname of the microcontroller to use</i>
	How to change	Changes not allowed
File version	Displays the device file version of the microcontroller to use.	
	Default	<i>The device file version of the microcontroller to use</i>
	How to change	Changes not allowed
Bank size [KBytes]	Displays the bank size of the microcontroller being used in Kbytes, as a decimal number. Note that this property is only shown for microcontrollers with a memory bank.	
	Default	<i>The bank size recorded in your microcontroller's device file</i>
	How to change	Changes not allowed
Internal ROM size [KBytes]	Displays internal ROM size [KBytes]of the microcontroller to use in decimal number.	
	Default	<i>Internal ROM size of the microcontroller to use</i>
	How to change	Changes not allowed
Internal RAM size [Bytes]	Displays internal RAM size [KBytes] of the microcontroller to use in decimal number.	
	Default	<i>Internal RAM size of the microcontroller to use</i>
	How to change	Changes not allowed

(3) [Notes]

The detailed information on records is displayed and their configuration can be changed.

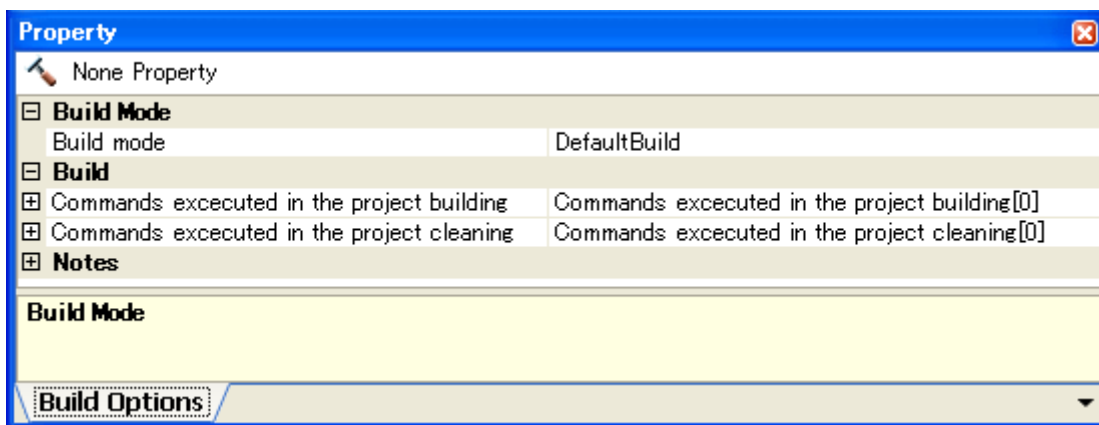
Memo	Add memos to the microcontroller information. The memo is added one item in one line. The added memos are displayed as subproperty.s	
	Default	Memo [<i>Number of items</i>]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can use a text box directly enter the text.
	Restriction	Up to 256 characters Up to 256 characters are allowed.

[Build Options] tab

This tab shows the detailed information on the build tool for the debug-dedicated project (see "[APPENDIX E USING AN EXTERNAL BUILD TOOL](#)") categorized by the following and the configuration can be changed.

- (1) [Build Mode]
- (2) [Build]
- (3) [Notes]

Figure A-23. Property Panel: [Build Options] Tab



[Description of each category]

(1) [Build Mode]

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

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

(2) [Build]

The detailed information on a build is displayed and their configuration can be changed.

Commands executed in the project building	Specify the commands to be executed when running a build (see "E.5.1 Run a build") of the debug-dedicated project. The following macro name is available as an embedded macro. %BuildModeName%: Replaces with the build mode name. %ProjectFolder%: Replaces with the absolute path of the project folder.	
	Default	Commands executed in the project building[0]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can enter directly in the text box.
	Restriction	Up to 1023 characters Up to 64 commands can be specified.
Commands executed in the project cleaning	Specify the commands to be executed when running a clean (see "E.5.3 Run a clean") of the debug-dedicated project. The following macro name is available as an embedded macro. %BuildModeName%: Replaces with the build mode name. %ProjectFolder%: Replaces with the absolute path of the project folder.	
	Default	Commands executed in the project cleaning[0]
	How to change	Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can enter directly in the text box.
	Restriction	Up to 1023 characters Up to 64 commands can be specified.

(3) [Notes]

The detailed information on records is displayed and their configuration can be changed.

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

Editor panel

This panel is used to display and edit text files and source files.

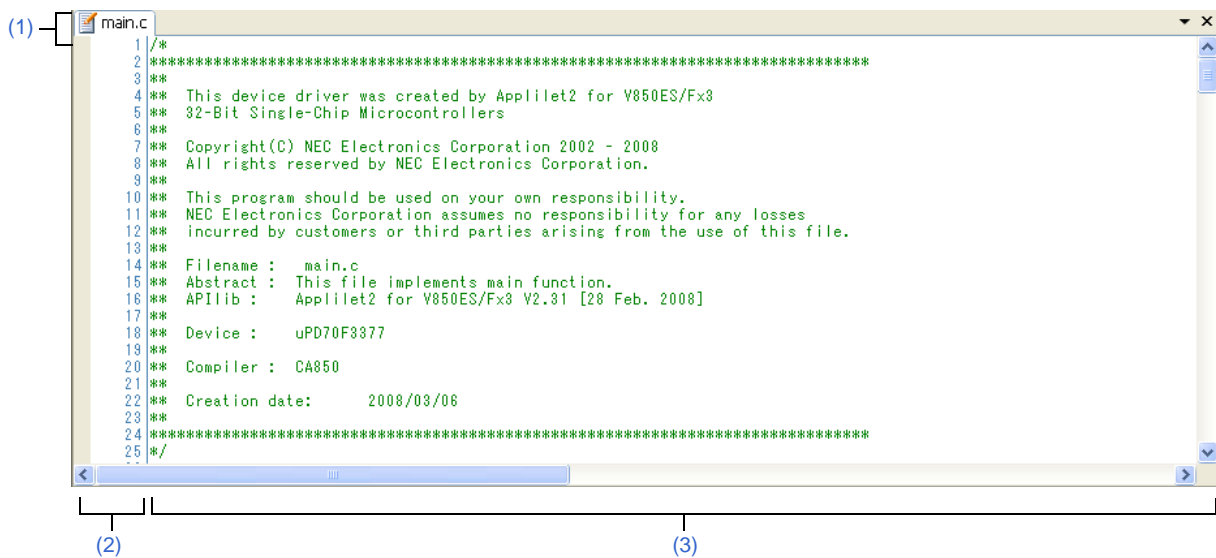
The file is opened by automatically distinguishing the encoding (Shift_JIS/EUC-JP/UTF-8) and line feed code of the file and the encoding is retained when it is saved.

If the encoding and newline code is specified in the [Save Settings dialog box](#), however, then the file is saved in accordance with those settings.

This panel can be multiply opened (max:100 panels).

Remark A message is shown when the downloaded load module file is older than the source file to be opened.

Figure A-24. Editor Panel



The following items are explained here.

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

[How to open]

- On the [Project Tree panel](#), double click a file.
- On the [Project Tree panel](#), select a source 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.

[Description of each area]**(1) Title bar**

The name of the open text file or source file is displayed.

Marks that are displayed at the end of the file name indicate as follows.

Mark	Description
*	The contents of the editing file is changed.
(Uneditable)	The opened text file is write disabled.
<i>ID number</i>	The same text file is multiply opened.

(2) Line number area

This area displays the line number of the opened text file or source file.

(3) Characters area

This area displays character strings of text files and source files and you can edit it.

This area has the following functions.

(a) Character editing

Characters can be entered from the keyboard.

Various shortcut keys can be used to enhance the edit function.

(b) File monitor

The following function for monitoring is provided to manage source files.

- If the contents of the currently displayed file are changed not with CubeSuite, a message is displayed to indicate whether to save the file. You can either select yes or no.

Remark The following items can be customized by setting the [Option dialog box](#).

- Display fonts
- Tab interval
- Display, hide, and colors of control characters (control codes including a blank symbol)
- Colors of reserved words and comments

[[File] menu (only available for the Editor panel)]

The following items are exclusive for the [File] menu in the Editor panel (other items are common to all the panels).

Close <i>file name</i>	Closes the currently editing the Editor panel. When the contents of the panel have not been saved, a confirmation message is shown.
Save <i>file name</i>	Overwrites the contents of the currently editing the Editor panel. 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 <i>file name</i> As...].
<i>file name</i> Save Settings...	This dialog box is used to open the Save Settings dialog box to set the encoding and newline code of the file that is editing on this panel.
Save <i>file name</i> As...	Opens the Save As dialog box to newly save the contents of the currently editing the Editor panel.
Page Setup...	Opens the Page Setup dialog box of Windows.
Print...	Opens the Print dialog box of Windows for printing the contents of the currently editing the Editor panel.

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

The following items are exclusive for the [Edit] menu in the Editor panel (other items are all invalid).

Undo	Cancels the previous operation on the Editor panel and restores the characters and the caret position (max 100 times).
Redo	Cancels the previous [Undo] operation on the Editor panel and restores the characters and the caret position.
Cut	Cuts the selected characters and copies them to the clip board.
Copy	Copies the selected characters to the clipboard.
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.
Delete	Deletes one character at the caret position. When there is a selection area, all the characters in the area are deleted.
Select All	Selects all the characters from the beginning to the end in the currently editing text file.
Find...	Opens the Search and Replace dialog box with the [Quick Search] tab target. When there is a selection area, search is only taken place in the selection area.
Replace...	Opens the Search and Replace dialog box with the [Quick Replace] tab target. When there is a selection area, replace is only taken place in the selection area.
Move To...	Opens the Go to the Location dialog box to move the caret to the designated line.

[Context menu]

[Characters area/Line number area]

Jump To Function	Jumps to the function regarding the selected characters and the words at the caret position as a function ^{Note} . This menu is only enabled if there is text on the caret line.
Back To Last Cursor Position	Goes back to the position before the cursor is jumped.
Forward To Next Cursor Position	Forwards to the position before operating [Back To Last Cursor Position].
Tag jump	Jumps to the caret line in the editor indicated by the message (file, line, and column).
Cut	Cuts the selected characters and copies them to the clip board.
Copy	Copies the selected characters to the clip board.
Paste	Inserts the contents of the clipboard into the caret position.
Open in New Panel	Opens a new Editor panel with the same contents as the current Editor panel (the title bar of the newly opened Editor panel displays the file name and ID number). The Editor panel can be opened up to 100 panels.

Note This function is only available when the following conditions are satisfied (see "CubeSuite Debug" for details):

- When the build tool to be used is CX
 - The target is a function, variable, or label in C language.
- When the build tool to be used is CA78K0, CA78K0R or CA850
 - The load module file with the symbol information has been built.
 - The target has been defined in the active project.
 - The type of a project specified as active project is "Application".
- When a project type is "Debug Only" (a debug-dedicated project)
 - The load module file with the symbol information has been added to the Download files node.
 - The target has been defined in the active project.

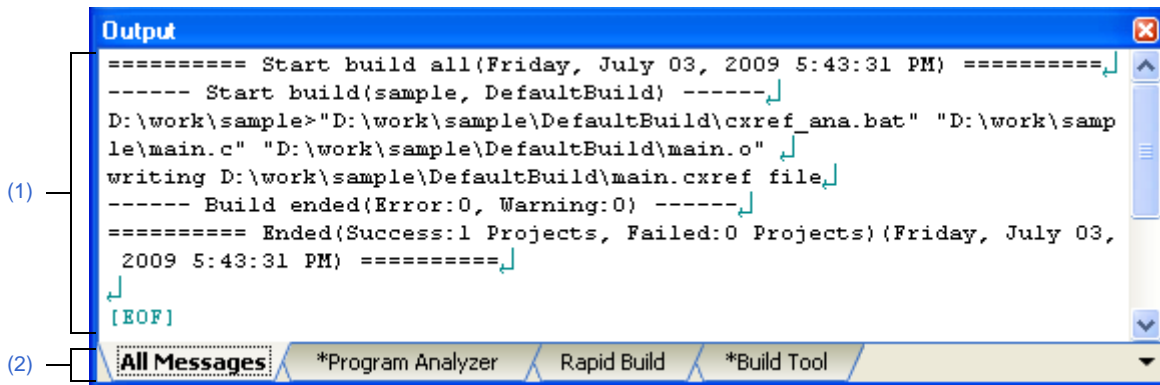
Output panel

The message that is output from the build tool/debug tool/each plug-in or the result of the Whole Search with the [Search and Replace dialog box](#) is displayed.

Messages are shown individually on the tab categorized by the output tool.

The [All Messages] tab consolidates and displays all output messages.

Figure A-25. Output Panel



The following items are explained here.

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

Displays messages and the search results output from each tool.

The [All Messages] tab consolidates and displays all output messages.

The message colors differ as follows depends on the type of the output message (the character color/background color is set in [\[General - Font and Color\] category](#) in [Option dialog box](#)).

Message type	Example (Default)		Description	
Normal message	AaBbCc	Character color	Black	Information on something.
		Background color	White	
Warning	AaBbCc	Character color	Blue	Warning for the operation.
		Background color	Normal color	
Error message	AaBbCc	Character color	Red	Fatal error or operation disabled because of an error in operation.
		Background color	Light gray	

This area has the following functions.

(a) Tag jump

When the output message is double-clicked, or the [Enter] key is pressed with the caret over the message, Editor panel appears and the destination line number of the file is displayed.

You can jump to the line of the source file that generated the error from the error message output when building.

(b) Display of help

Displays the help with regard to the displayed message or the message at the current caret by selecting [Help for Message] in the context menu or pressing [F1] key while the caret is in the line where the warning message or the error message is displayed.

(c) Save of log

The contents displayed on the currently selected tab can be saved in a text file (*.txt) by selecting [Save Output - *tab name* As...] from [File] menu and opens the [Save As dialog box](#) (messages on the tab that is not selected will not be saved).

(2) Tab selection area

Select tabs that messages are output from.

Tabs that are displayed are as follows.

Tab Name	Description
Build Tool	Displays the messages output by the build tool when a build, rebuild, or batch build is executed.
Rapid Build	Displays the messages output by the build tool when a rapid build is executed.
Debug Tool	Displays the message output from the debug tool.
Code Generator	Displays the messages output by the code generator.
Program Analyzer	Displays the message output from the analyze tool.
Find References	Displays a list of locations of references to functions/variables output by the analyze tool
Flash Programming Tool	Displays the message output from the flash programming tool.
Search and Replace	Displays the Whole Search result with the Search and Replace dialog box .
All Messages	Shows all the messages above by order of output.

Caution Tab is not automatically switched when a new message is output on the non-selected tab. If this is the case, * is added to the tab informing a new message is output.

[[File] menu (Output panel-dedicated items)]

The following items are exclusive for [File] menu in the Output panel (other items are common to all the panels).

Output - Save <i>tab name</i>	Saves the contents on the currently selecting tab in the previously saved text file (*.txt) (see "(c) Save of log"). When this menu is selected for the first time after launching the program, the operation is equivalent to when selecting [Save <i>tab name</i> As...]. Note that this operation is invalid while building.
Save Output - <i>tab name</i> As...	Opens the Save As dialog box to save the contents on the currently selecting tab in the designated text file (*.txt) (see "(c) Save of log"). Note that this operation is invalid while user program is executed.

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

The following items are exclusive to [Edit] menu in the Output panel (other items are all invalid).

Copy	Copies the selected characters to the clip board.
Select All	Selects all the messages displayed on the panel.
Find...	Opens the Search and Replace dialog box with the [Quick Search] tab target.
Replace...	Opens the Search and Replace dialog box with the [Whole Replace] tab target.

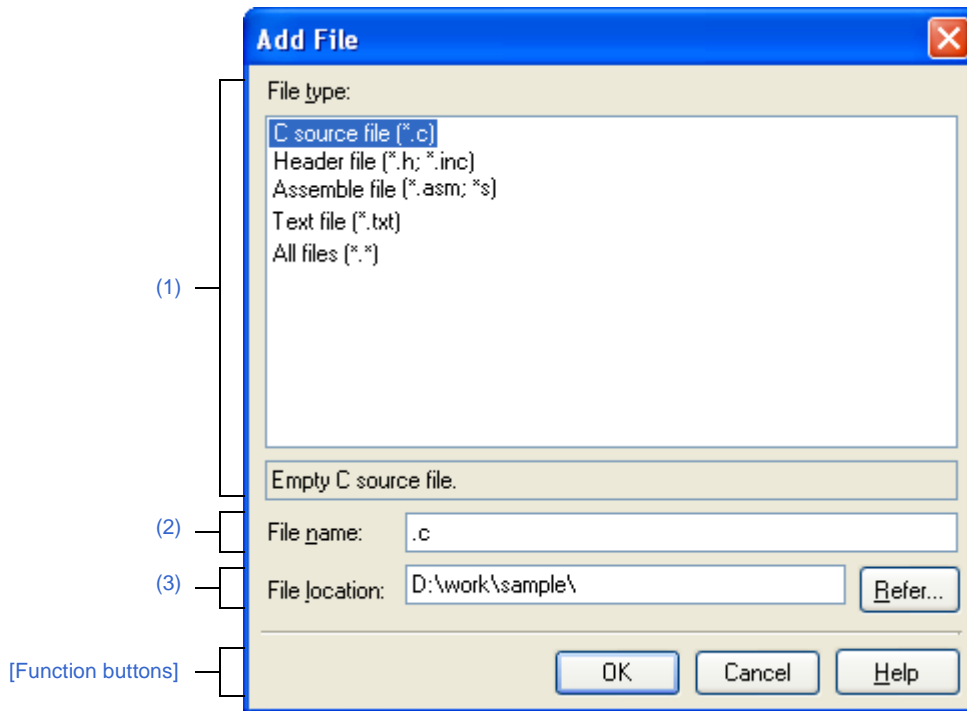
[Context menu]

Copy	Copies the selected characters to the clip board.
Select All	Selects all the messages displayed on the panel.
Clear	Deletes all the messages displayed on the panel.
Tag Jump	Jumps to the caret line in the editor indicated by the message (file, line, and column).
Help for Message	Displays the help with regard to the displayed message or the message at the current caret. Note that the help is only for warning/error messages.

Add File dialog box

This dialog box is used to create a new file and add it to the project.

Figure A-26. Add File Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [File] menu, select [Add] >> [Add New File...].
- On the [Project Tree](#) panel, select the Download files node, and then select [Add] >> [Add New File...] from the context menu.

[Description of each area]

(1) [File type] area

Select the type of the file to be created.

When the file type is selected, the description is displayed at the lower box.

The file types to be displayed are as follows.

- C source file (*.c)
- Header file (*.h; *.inc)
- Assemble file (*.asm; *.s)
- Text file (*.txt)
- All files (*.*)

(2) [File name] area

Directly enter the name of the file to be created.
".c" is displayed by default.

Remark If any extension is not designated, the one selected in the [File type] area will be added.
Also, if the extension different from the one selected in the [File type] area is designated, the one selected in the [File type] area will be added (for example, if you designate "aaa.txt" as the file name and select "C source file (*.c)" as the file type, the file is named as "aaa.txt.c").
Note that if [All files (*.*)] is selected in the [File type] area, no extension will be added.

(3) [File location] area

Designate the location to be created the file by directly entering its path or selecting from the [Refer...] button.
The path of the project folder is displayed by default.

(a) Button

Refer...	Opens the Browse For Folder dialog box. If a folder is selected, the path will be added in the text box.
----------	---

- Remarks 1.** If the text box is blank, it is assumed that the project folder is designated.
2. If the relative path is designated, the reference point of the path is the project folder.

Remark Up to 259 characters (path and file name combined) can be specified in the [File name] area and [File location] area. When the input violates any restriction, the following messages will be shown on the [File name] area in the tooltip.

Message	Description
The file name including the path is too long. Make it within 259 characters.	The file name with the path is more than 259 characters.
The specified path contains a folder that does not exist.	The path contains a folder that does not exist.
The file name or path name is invalid. The following characters cannot be used: \, /, :, *, ?, ", <, >,	The file name with the invalid path is designated. The following characters cannot be used for the file name and folder name: \, /, :, *, ?, ", <, >,

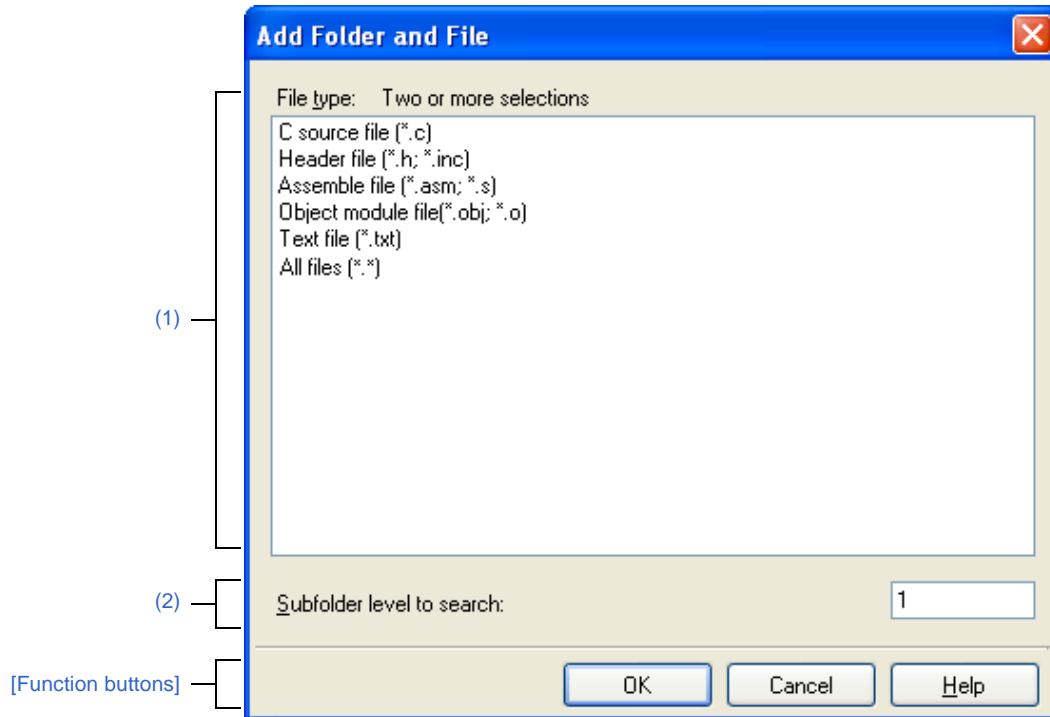
[Function buttons]

Button	Function
OK	Creates the file with the entered file name, adds it to the project, and opens with the Editor panel . And then closes this dialog box.
Cancel	Does not create a file and closes this dialog box.
Help	Displays the help of this dialog box.

Add Folder and File dialog box

This dialog box is used to add existing files and folder hierarchies to the project.
The folder is added as a category.

Figure A-27. Add Folder and File Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- Drag the folder from Explorer or the like, and drop it on the [Project Tree panel](#).

[Description of each area]

(1) [File type] area

Select the type of the file to be added to the project.

You can select multiple file types by left clicking while holding down the [Ctrl] or [Shift] key.

If nothing is selected, it is assumed that all types are selected.

The file types to be displayed are as follows.

- C source file (*.c)
- Header file (*.h; *.inc)
- Assemble file (*.asm; *.s)
- Object module file (*.obj; *.o)
- Text file (*.txt)
- All files (*.*)

(2) [Subfolder level to search] area

Directly enter the number of levels of the subfolder to be added to the project.
 "1" is displayed by default.

Remark Up to 10 (decimal number) can be specified.
 When the input violates any restriction, the following messages will be shown in the tooltip.

Message	Description
Fewer than 0 or more than 10 values cannot be specified.	More than 10 subfolder levels have been specified.
Specify in decimal.	A number in other than decimal or a string has been specified.

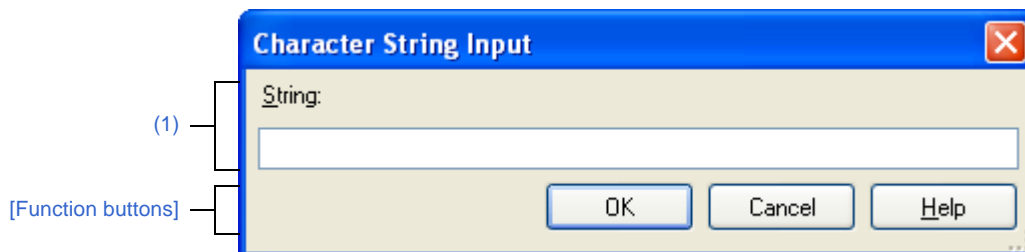
[Function buttons]

Button	Function
OK	Adds the folder that was dragged and dropped and the files in that folder to the project. And then closes this dialog box.
Cancel	Does not add the folder and files, and then closes this dialog box.
Help	Displays the help of this dialog box.

Character String Input dialog box

This dialog box is used to input and edit characters in one line.

Figure A-28. Character String Input Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- In the [Build Mode Settings dialog box](#), select a build mode to be duplicated from [Build mode list], and then click the [Duplicate...] button

[Description of each area]

(1) [String] area

Input characters in one line.

By default, the current value of the area that this dialog box is called from is reflected to this area.

You cannot start a new line.

Remark Up to 32767 characters can be entered.

When the input violates any restriction, the following messages will be shown in the tooltip.

Message	Description
More than <i>maximum number of restriction in the property that called this dialog box</i> characters cannot be specified.	The numbers of input characters exceeds the maximum number of restriction in the property that called this dialog box.

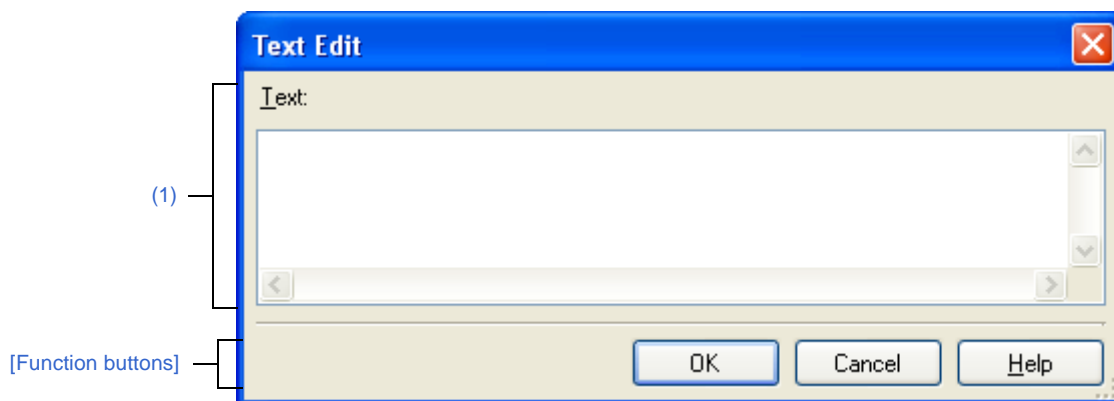
[Function buttons]

Button	Function
OK	Reflects the entered characters to the area that this dialog box is called from and closes this dialog box.
Cancel	Does not reflect the entered characters to the area that this dialog box is called from and closes this dialog box.
Help	Displays the help of this dialog box.

Text Edit dialog box

This dialog box is used to input and edit texts in multiple lines.

Figure A-29. Text Edit Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- On the [Build Options] tab of the Property panel, select either one of the following properties, and then click the [...] button.
 - The [Commands executed in the project building] property in the [Build] category
 - The [Commands executed in the project cleaning] property in the [Build] category

[Description of each area]

(1) [Text] area

Edit texts in multiple lines.

By default, the current value of the area that this dialog box is called from is reflected in this area.

Remark When the input violates any restriction, the following message will be shown in the tooltip.

Message	Description
More than <i>maximum number of restriction in the property that called this dialog box</i> characters cannot be specified. The current number of characters is displayed between brackets at the beginning of the line in excess of the limit.	The numbers of input characters exceeds the maximum number of restriction in the property that called this dialog box.

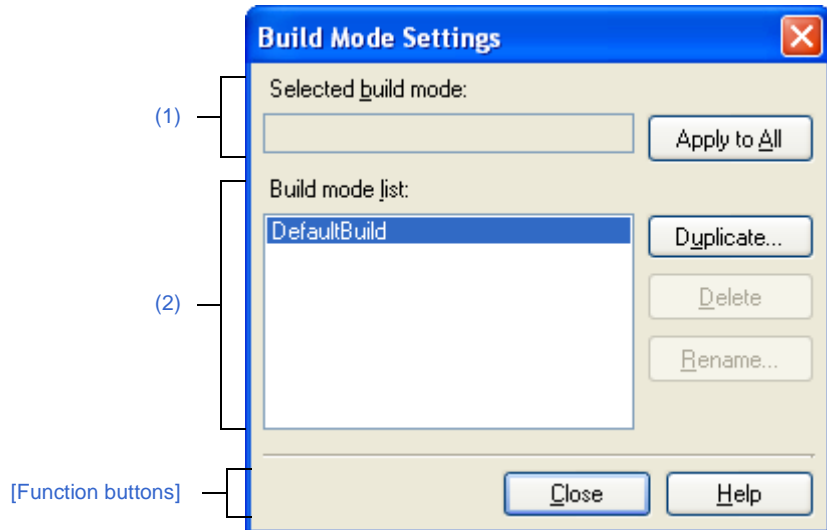
[Function buttons]

Button	Function
OK	Reflects the entered text to the text box that opened this dialog box and closes this dialog box.
Cancel	Does not reflect the entered text to the text box that opened this dialog box and closes this dialog box.
Help	Displays the help of this dialog box.

Build Mode Settings dialog box

This dialog box is used to add and delete build modes and configure the current build mode in batch.

Figure A-30. Build Mode Settings Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Build] menu, select [Build Mode Settings...].

[Description of each area]

(1) [Selected build mode] area

This area displays the build mode selected in the [Build mode list] area.

(a) Button

Apply to All	Set the build mode of the main project and all subprojects of the currently opened project to the currently displayed build mode.
--------------	---

(2) [Build mode list] area

This area displays all build modes that exist in the currently opening project (main project and subproject) in a list. The current build modes of all projects are same, the build mode is selected by default. If they are not same, "DefaultBuild" will be selected.

The build mode that exists only in part of the main project and subproject is shown with the mark "**".

Note that the "DefaultBuild" is the default build mode and is always displayed at the top.

(a) Button

Duplicate...	<p>Duplicates the selected build mode.</p> <p>The Character String Input dialog box opens. And the build mode is duplicated with the name entered in the dialog box and added to the main project and all the subprojects in the currently opening project.</p> <p>When the build mode with "*" mark does not exist in the main project or subproject and duplicate the build mode, "DefaultBuild" will be duplicated.</p> <p>Up to 20 build modes can be added.</p>
Delete	<p>Deletes the selected build mode.</p> <p>Note that "DefaultBuild" cannot be deleted.</p>
Rename...	<p>Renames the selected build mode.</p> <p>Rename the build mode with entered name in the opening the Character String Input dialog box.</p>

Caution When duplicating or renaming the build mode, the existing build mode name cannot be used.

Remarks 1. Up to 127 characters can be specified as a build mode name. When the input violates any restriction, the following messages are shown in the tooltip.

Message	Description
A build mode with the same name already exists.	The entered build mode name already exists.
More than 127 characters cannot be specified.	Build mode name is too long (more than 128 characters).
The build mode name is invalid. The following characters cannot be used: \, /, :, *, ?, ", <, >,	Invalid build mode name is entered. The characters (\, /, :, *, ?, ", <, >,) cannot be used because the build mode name is used for the folder name.

2. Up to 20 build modes can be added. When the input violates any restriction, the following messages are shown in the tooltip.

Message	Description
The maximum number of build modes that can be set per project/subproject is 20.	The number of build modes exceed 20.

[Function buttons]

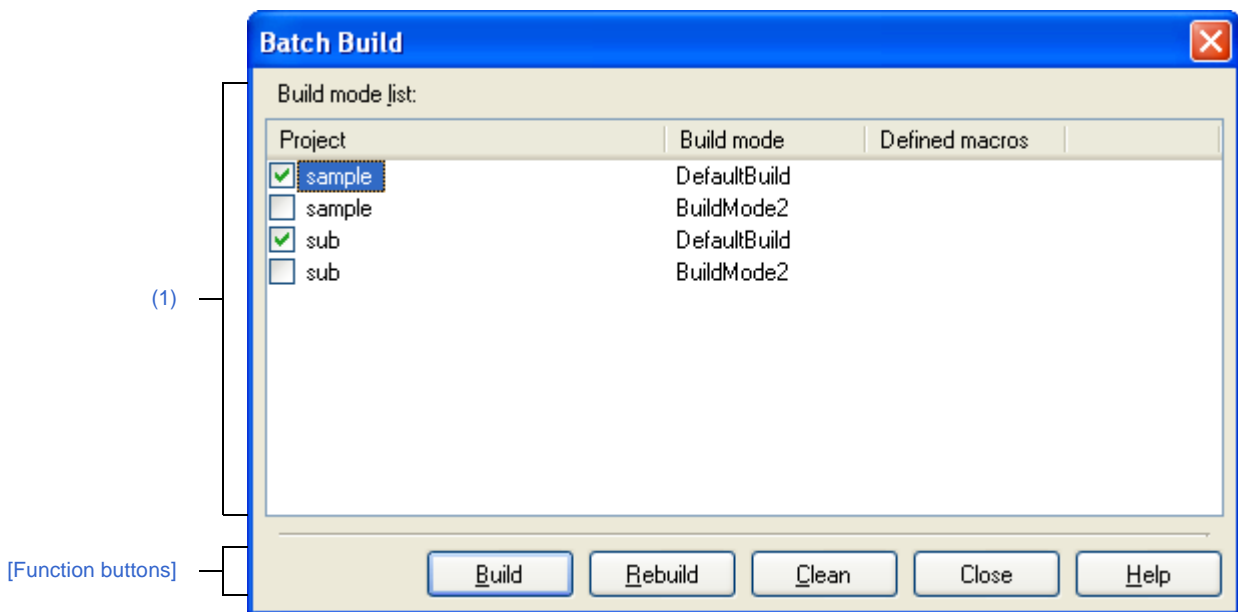
Button	Function
Close	Closes this dialog box.
Help	Displays the help of this dialog box.

Batch Build dialog box

This dialog box is used to run builds, rebuilds and cleans in batch with the build modes that the project (main project and subproject) has.

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

Figure A-31. Batch Build Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Build] menu, select [Batch Build...].

[Description of each area]

(1) [Build mode list] area

This area displays the list of the combinations of the names of the main project and subprojects in the currently opened project, their build modes, and their macro definitions.

(a) [Project]

This area displays the main project and subprojects in the currently opened project.

Select the check boxes for the combinations of the main project and subprojects and their build modes which you wish to run a build.

When this dialog box is opened for the first time after the project is created, all the check boxes are unchecked. From the second time, the previous setting is retained.

(b) [Build mode]

This area displays the build modes which the main project and subprojects have.

(c) [Defined macros]**<1> When the target project type is other than a debug-dedicated project**

For the combination of the main project and subprojects and their build modes, the defined macros which have been set in the [Compile Options] tab and the [Assemble Options] tab on the Property panel are separated with "|" and displayed.

The defined macro in the compile option comes before the one in assemble option. They are separated with ", " and displayed.

<2> When the target project type is a debug-dedicated project

This item is invalid.

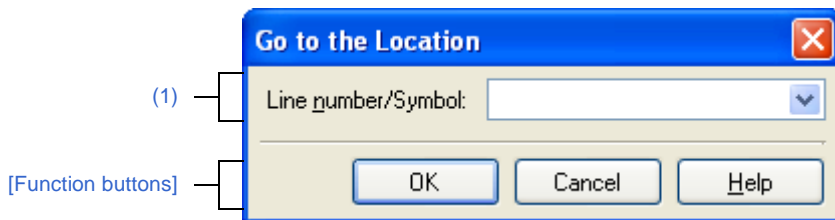
[Function buttons]

Button	Function
Build	<p>Closes this dialog box and runs builds of the selected projects in the respective build modes.</p> <p>The execution result of the builds is displayed on the Output panel.</p> <p>After the builds are complete, the build mode configuration restores to the one before this dialog box was opened.</p> <p>If any project is not selected, this button will be disabled.</p>
Rebuild	<p>Closes this dialog box and runs rebuilds of the selected projects in the respective build modes.</p> <p>The execution result of the rebuilds is displayed on the Output panel.</p> <p>After the rebuilds are complete, the build mode configuration restores to the one before this dialog box was opened.</p> <p>If any project is not selected, this button will be disabled.</p>
Clean	<p>Closes this dialog box and deletes the files which are built in the respective build modes set for the selected projects.</p> <p>The execution result of the cleans is displayed on the Output panel.</p> <p>After the cleans are complete, the build mode configuration restores to the one before this dialog box was opened.</p> <p>If any project is not selected, this button will be disabled.</p>
Close	Closes this dialog box.
Help	Displays the help of this dialog box.

Go to the Location dialog box

This dialog box is used to move the caret to the designated location.

Figure A-32. Go to the Location Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Edit] menu, select [Move To...].

[Description of each area]

(1) [Line number/Symbol] area

Specify the line number (decimal number) or symbol name of the location to which the caret is moved. You can directly enter the characters into the text box or select from the input history in the drop down list (maximum numbers of the history: 10).

[Function buttons]

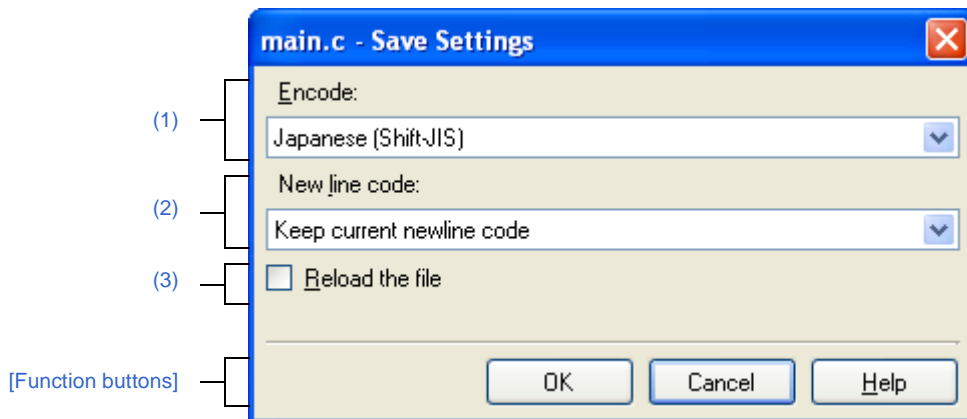
Button	Function
OK	Displays the designated location at the middle of the target panel display and moves the caret there.
Cancel	Cancels the criteria and closes this dialog box.
Help	Displays the help of this dialog box.

Save Settings dialog box

This dialog box is used to set the encoding and newline code of the file that is being edited on the [Editor panel](#).

Remark The target file name is displayed on the title bar.

Figure A-33. Save Settings Dialog Box



The following items are explained here.

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

[How to open]

- Focus the [Editor panel](#), and then select [*file name* Save Settings...] from the [File] menu.

[Description of each area]

(1) [Encode]

Select the encoding to be set from the drop-down list.

The items of the drop-down list are displayed according to the following sequence.

Note that the same encoding and encoding which are not supported by the current OS will not be displayed.

- *Encoding of the current file (default)*
- *Default encoding of the current OS*
- *Encoding of code page 932 (SJIS)*
- *Encoding of code page 50222 (JIS)*
- *Encoding of code page 51932 (EUC)*
- *Encoding of code page 65001 (UTF8)*

(2) [Newline code]

Select the newline code to be set from the drop-down list.

You can select any of items below.

- Keep current newline code
- Windows (CR LF)
- Macintosh (CR)
- Unix (LF)

"Keep current newline code" is selected by default.

After the newline code is changed, the set newline code is selected by default.

(3) [Reload the file]

Use this check box to select whether to reload the file with the selected encoding and newline code when the [OK] button is clicked.

The check box is not selected by default.

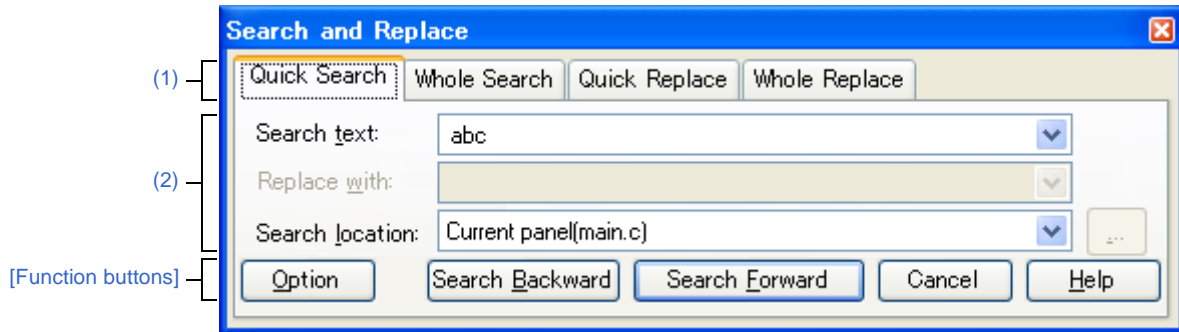
[Function buttons]

Button	Function
OK	Sets the selected encoding and newline code to the target file and closes this dialog box. If [Reload the file] is selected, sets the selected encoding and newline code to the target file and reloads the file. And then closes this dialog box.
Cancel	Cancels the settings of the encoding and newline code and closes the dialog box.
Help	Displays the help of this dialog box.

Search and Replace dialog box

This dialog box is used to search and replace the designated characters.

Figure A-34. Search and Replace Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Edit] menu, select [Search...].
- From the [Edit] menu, select [Replace...].

[Description of each area]

(1) Tab selection area

Search/replace is switched when a tab is selected.

This dialog box has the following tabs.

- [Quick Search] tab
- [Whole Search] tab
- [Quick Replace] tab
- [Whole Replace] tab

(2) Search/replace criteria setting area

Detailed criteria for searching/replacing is set.

Please see the description of the relevant tabs for details of the contents/how to set.

[Function buttons]

Buttons for execute search/replace.

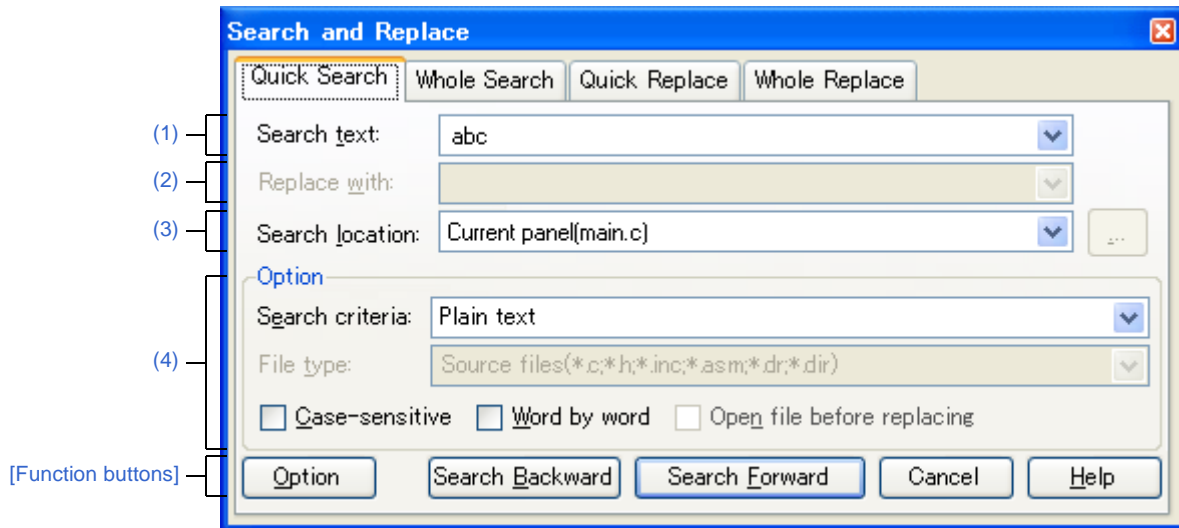
Please see the description of the relevant buttons for details.

[Quick Search] tab

This tab searches the designated characters and moves the caret to the searched position with the position being selected.

Note that this tab will be disabled if you call the [Search and Replace dialog box](#) from a panel other than the [Output panel](#) / [Editor panel](#).

Figure A-35. Search and Replace Dialog Box: [Quick Search] Tab



The following items are explained here.

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

[How to open]

- From the [Edit] menu, select [Search...].
- From the [Edit] menu, select [Replace...].

[Description of each area]**(1) [Search text]**

Designate characters to search.

You can directly enter the characters into the text box (maximum characters: 1024) or select from the input history in the drop-down list (maximum numbers of the history: 10).

If this dialog box is opened from the panel with the character being selected, the selected characters are shown by default.

If this dialog box is opened from the Editor panel, words (variable/function) at the caret position are shown by default.

(2) [Replace with]

This item is disabled.

(3) [Search location]

Designate the location to search.

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

Item	Operation
Selection area	Searches the selection in the search enabled panel which was active the last time. If there is no characters in selection in the panel which was last active, or the panel cannot be searched, this item will be disabled.
Current panel (<i>Panel Name</i>)	Searches in the panel which was last active and can be searched. If the panel which was lastly active cannot be searched or the panel does not exist, this item will be disabled.

Remark Up to 10 items for the input history are recorded in the drop-down list.

(4) [Option] area

The following options can be designated as search criterias.

This area is shown when the [Option] button is clicked (not shown by default).

(a) [Search criteria]

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

Item	Operation
Plain text	Searches the characters designated in [Search text] .
Wild-card	Searches using the following wildcard.
	* Arbitrary characters.
	? Arbitrary one character.

(b) [File type]

This item is disabled.

(c) [Case-sensitive]

<input checked="" type="checkbox"/>	Searches with the designated characters in case-sensitive.
<input type="checkbox"/>	Searches with the designated characters in not case-sensitive (default).

(d) [Word by word]

<input checked="" type="checkbox"/>	Searches with a designated exact word.
<input type="checkbox"/>	Searches with at least one of the words (default).

(e) [Open file before replacing]

This item is disabled.

[Function buttons]

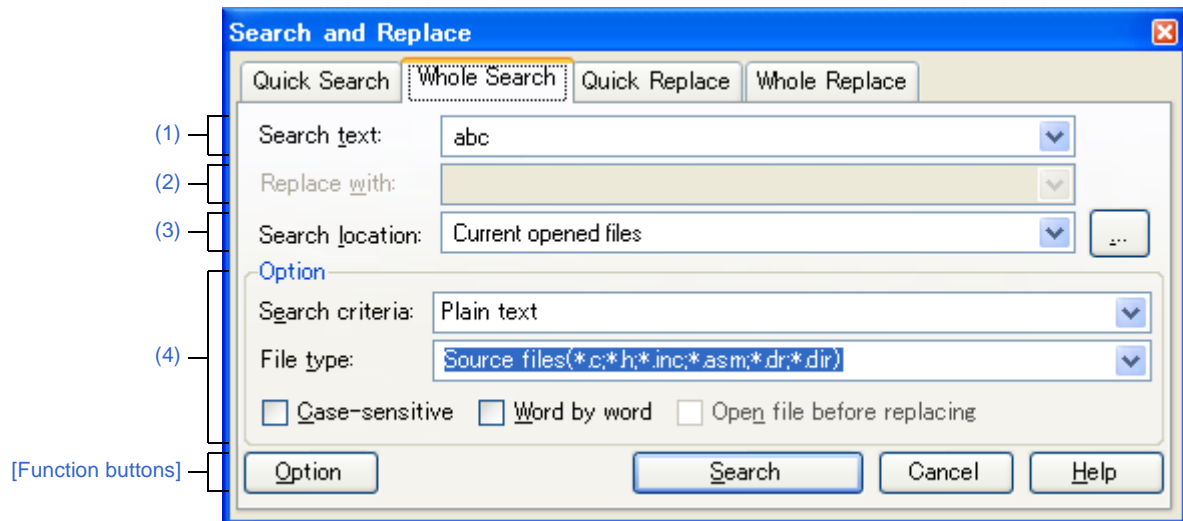
Button	Function
Option	Switches between display/hide the [Option] area in this tab.
Search Backward	Searches from the current caret position to the top of the file with the designated criteria. Selects the characters that are searched and moves the caret ^{Note} .
Search Forward	Searches from the current caret position to the end of the file with the designated criteria. Selects the characters that are searched and moves the caret ^{Note} .
Cancel	Ignores the setting and closes this dialog box.
Help	Displays the help of this dialog box

Note If the designated characters cannot be searched, "Can not found *Search text*." is displayed on the status bar of the [Main window](#).

[Whole Search] tab

In this tab, the designated characters are searched in batch and the search results are listed in the [Output panel](#). The [Output panel](#) is used to jump to the relevant location by double-clicking the search result.

Figure A-36. Search and Replace Dialog Box: [Whole Search] Tab



The following items are explained here.

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

[How to open]

- From the [Edit] menu, select [Search...].
- From the [Edit] menu, select [Replace...].

[Description of each area]**(1) [Search text]**

Designate characters to search.

You can directly enter the characters into the text box (maximum characters: 1024) or select from the input history in the drop-down list (maximum numbers of the history: 10).

If this dialog box is opened from the panel with the character being selected, the selected characters are shown by default.

If this dialog box is opened from the Editor panel, words (variable/function) at the caret position are shown by default.

(2) [Replace with]

This item is disabled.

(3) [Search location]

Designate the location to search.

Select either one of the following items from the drop-down list or directly enter the file location from the keyboard (maximum number: 10).

Item	Operation
Current opened files	Searches within all the opening the Editor panel. If no file is opened in the Editor Panel, this item is disabled.
Active project	Searches within the text file included in the active project. When [File type] is specified, searches only the specified type. Note that is the current project does not exist, this item is disabled.
Main project and subprojects	Searches within the text file included in the main project and subproject. When [File type] is specified, searches only the specified type. Note that is the current project does not exist, this item is disabled.
Folder Name	Searches within the text file in the folder specified by directly entering (the maximum characters: 259) the path (relative path is from the project folder), or specified in the Browse For Folder dialog box opened by clicking the [...] button in this area. When folders are not specified, the project folder name is shown in "()" by default folder (if the project does not exist, the current user document folder is shown). When [File type] is specified, searches only the specified type.

Remark Up to 10 items for the input history are recorded in the drop-down list.

(4) [Option] area

This area is shown when the [Option] button is clicked (not shown by default).

The following options can be designated as search criteria.

(a) [Search criteria]

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

Item	Operation
Plain text	Searches the characters designated in [Search text] .
Wild-card	Searches using the following wildcard.
	* Arbitrary characters.
	? Arbitrary one character.

(b) [File type]

Specify File types to search.

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

Item	Operation
Source files (<i>Extensions</i> ^{Note})	Files to search are limited to the source files.
*.txt	Files to search are limited to the text files.
.	Searches all the files.

Note Show extensions of the source file added to the [Project Tree panel](#).

Note that the searches can be operated by limiting the search criteria by directly entering the file name in the text box (maximum characters: 1024).

If this is the case, the wildcard "*" can be used and multiple file names can be specified by separating them with ";".

Remark Up to 10 items for the input history are recorded in the drop-down list.

(c) [Case-sensitive]

<input checked="" type="checkbox"/>	Searches with the designated characters in case-sensitive.
<input type="checkbox"/>	Searches with the designated characters in not case-sensitive (default).

(d) [Word by word]

<input checked="" type="checkbox"/>	Searches with a designated exact word.
<input type="checkbox"/>	Searches with at least one of the words (default).

(e) [Open file before replacing]

This item is disabled.

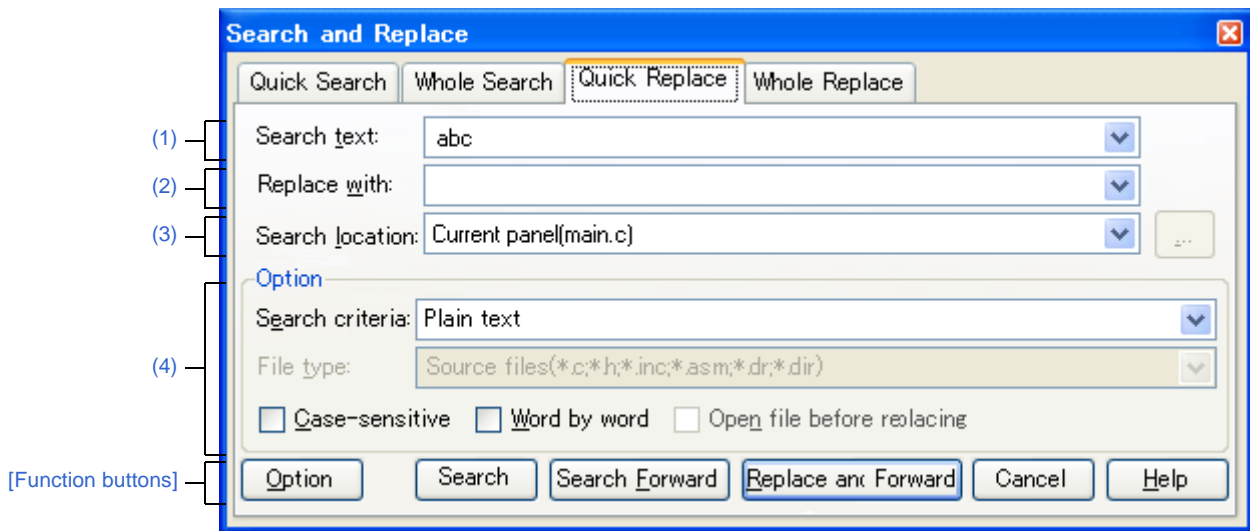
[Function buttons]

Button	Function
Option	Switches between display/hide the [Option] area in this tab.
Search	Searches characters with designated criteria in batch and shows the search results in list in the Output panel .
Cancel	Ignores the setting and closes this dialog box.
Help	Displays the help of this dialog box.

[Quick Replace] tab

In this tab, search is done with the designated characters and then they are replaced to the characters to be replaced. Note that this tab will be disabled if you call the [Search and Replace dialog box](#) from a panel other than the [Editor panel](#).

Figure A-37. Search and Replace Dialog Box: [Quick Replace] Tab



The following items are explained here.

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

[How to open]

- From the [Edit] menu, select [Search...].
- From the [Edit] menu, select [Replace...].

[Description of each area]**(1) [Search text]**

Designate characters to search.

You can directly enter the characters into the text box (maximum characters: 1024) or select from the input history in the drop-down list (maximum numbers of the history: 10).

If this dialog box is opened from the panel with the character being selected, the selected characters are shown by default.

If this dialog box is opened from the Editor panel, words (variable/function) at the caret position are shown by default.

(2) [Replace with]

Designate characters to be replaced.

You can directly enter the characters into the text box (maximum characters: 1024) or select from the input history in the drop-down list (maximum numbers of the history: 10).

(3) [Search location]

Designate the location to search.

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

Item	Operation
Selection area	Searches the selection in the search enabled panel which was active the last time. If there is no characters in selection in the panel which was last active, or the panel cannot be searched, this item will be disabled.
Current panel (<i>Panel Name</i>)	Searches in the panel which was last active and can be searched. If the panel which was lastly active cannot be searched or the panel does not exist, this item will be disabled.

Remark Up to 10 items for the input history are recorded in the drop-down list.

(4) [Option] area

This area is shown when the [Option] button is clicked (not shown by default).

The following options can be designated as search criteria.

(a) [Search criteria]

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

Item	Operation	
Plain text	Searches the characters designated in [Search text] .	
Wild-card	Searches using the following wildcard.	
	*	Arbitrary characters.
	?	Arbitrary one character.

(b) [File type]

This item is disabled.

(c) [Case-sensitive]

<input checked="" type="checkbox"/>	Searches with the designated characters in case-sensitive.
<input type="checkbox"/>	Searches with the designated characters in not case-sensitive (default).

(d) [Word by word]

<input checked="" type="checkbox"/>	Searches with a designated exact word.
<input type="checkbox"/>	Searches with at least one of the words (default).

(e) [Open file before replacing]

This item is disabled.

[Function buttons]

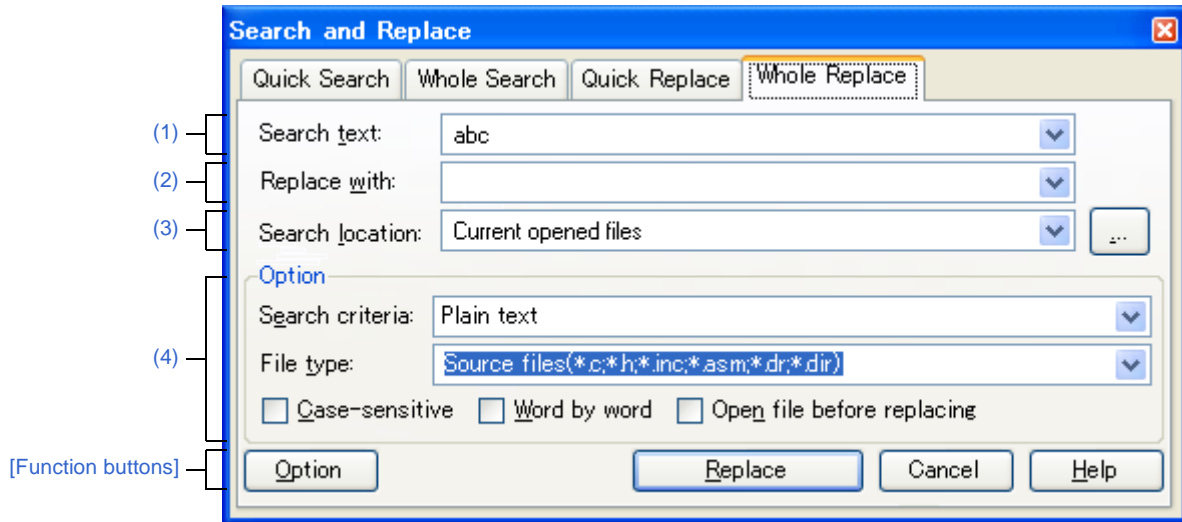
Button	Function
Option	Switches between display/hide the [Option] area in this tab.
Search	Searches from the current caret position to the top of the file with the designated criteria. Selects the characters that are searched and moves the caret ^{Note} .
Search Forward	Searches from the current caret position to the end of the file with the designated criteria. Selects the characters that are searched and moves the caret ^{Note} .
Replace and Forward	Replaces the selected characters to the characters to be replaced then searches the next (backwards) candidate and selects them ^{Note} .
Cancel	Ignores the setting and closes this dialog box.
Help	Displays the help of this dialog box.

Note If the designated characters cannot be searched, "Can not found *Search text*." is displayed on the status bar of the [Main window](#).

[Whole Replace] tab

In this tab, batch search is done with the designated characters and then they are replaced to the characters to be replaced in batch.

Figure A-38. Search and Replace Dialog Box: [Whole Replace] Tab



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Edit] menu, select [Search...].
- From the [Edit] menu, select [Replace...].

[Description of each area]**(1) [Search text]**

Designate characters to search.

You can directly enter the characters into the text box (maximum characters: 1024) or select from the input history in the drop-down list (maximum numbers of the history: 10).

If this dialog box is opened from the panel with the character being selected, the selected characters are shown by default.

If this dialog box is opened from the Editor, words (variable/function) at the caret position are shown by default.

(2) [Replace with]

Designate characters to be replaced.

You can directly enter the characters into the text box (maximum characters: 1024) or select from the input history in the drop-down list (maximum numbers of the history: 10).

(3) [Search location]

Designate the location to search.

Select either one of the following items from the drop-down list or directly enter the file location from the keyboard (maximum number: 10).

Item	Operation
Current opened files	Searches within all the opening the Editor panel. If no file is opened in the Editor panel, this item is disabled.
Active project	Searches within the text file included in the active project. When [File type] is specified, searches only the specified type. Note that if the current project does not exist, this item is disabled.
Main project and subprojects	Searches within the text file included in the main project and subproject. When [File type] is specified, searches only the specified type. Note that if the current project does not exist, this item is disabled.
Folder Name	Searches within the text file in the folder specified by directly entering (the maximum characters: 259) the path (relative path is from the project folder), or specified in the Browse For Folder dialog box opened by clicking the [...] button in this area. When folders are not specified, the project folder name is shown in "()" by default folder (if the project does not exist, the current user document folder is shown). When [File type] is specified, searches only the specified type.

Remark Up to 10 items for the input history are recorded in the drop-down list.

(4) [Option] area

This area is shown when the [Option] button is clicked (not shown by default).

The following options can be designated as search criteria.

(a) [Search criteria]

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

Item	Operation
Plain text	Searches the characters designated in [Search text] .
Wild-card	Searches using the following wildcard.
	* Arbitrary characters.
	? Arbitrary one character.

(b) [File type]

Specify File types to search.

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

Item	Operation
Source files (<i>Extensions</i> ^{Note})	Files to search are limited to the source files.
Text files (*.txt)	Files to search are limited to the text files.
All files (*.*)	Searches all the files.

Note Shows extensions of the source file added to the [Project Tree panel](#).

Note that the searches can be operated by limiting the search criteria by directly entering the file name in the text box (maximum characters: 1024).

If this is the case, the wildcard "*" can be used and multiple file names can be specified by separating them with ";".

Remark Up to 10 items for the input history are recorded in the drop-down list.

(c) [Case-sensitive]

<input checked="" type="checkbox"/>	Searches with the designated characters in case-sensitive.
<input type="checkbox"/>	Searches with the designated characters in not case-sensitive (default).

(d) [Word by word]

<input checked="" type="checkbox"/>	Searches with a designated exact word.
<input type="checkbox"/>	Searches with at least one of the words (default).

(e) [Open file before replacing]

<input checked="" type="checkbox"/>	Replace is done after opening the file to search/replace characters in the Editor panel.
<input type="checkbox"/>	Replace is done without opening the file to search/replace characters (default).

[Function buttons]

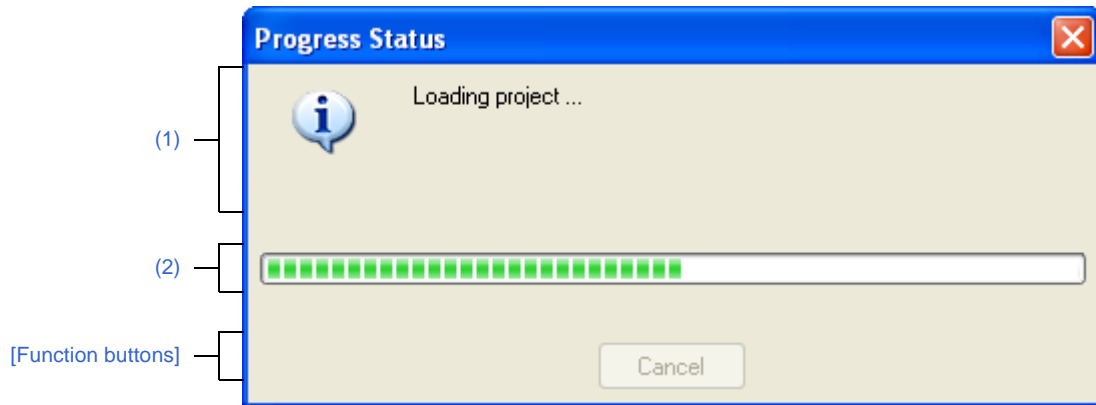
Button	Function
Option	Switches between display/hide the [Option] area in this tab.
Replace	Searches characters with designated criteria in batch and replaces the searched characters to the one designated to be replaced.
Cancel	Ignores the setting and closes this dialog box.
Help	Displays the help of this dialog box.

Progress Status dialog box

This dialog box is used to display how the process has been progressed when the time consuming process is taken place.

This dialog box automatically closes when the process in progress is done.

Figure A-39. Progress Status Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- The dialog box automatically opens when a message is output while the time consuming process is in progress.

[Description of each area]

(1) Message Display Area

Displays the message output while process is in progress (edit not allowed).

(2) Progress bar

The progress bar shows the current progress of the process in progress with the bar length.

When the process is 100% done (the bar gets to the right end), this dialog box automatically closed.

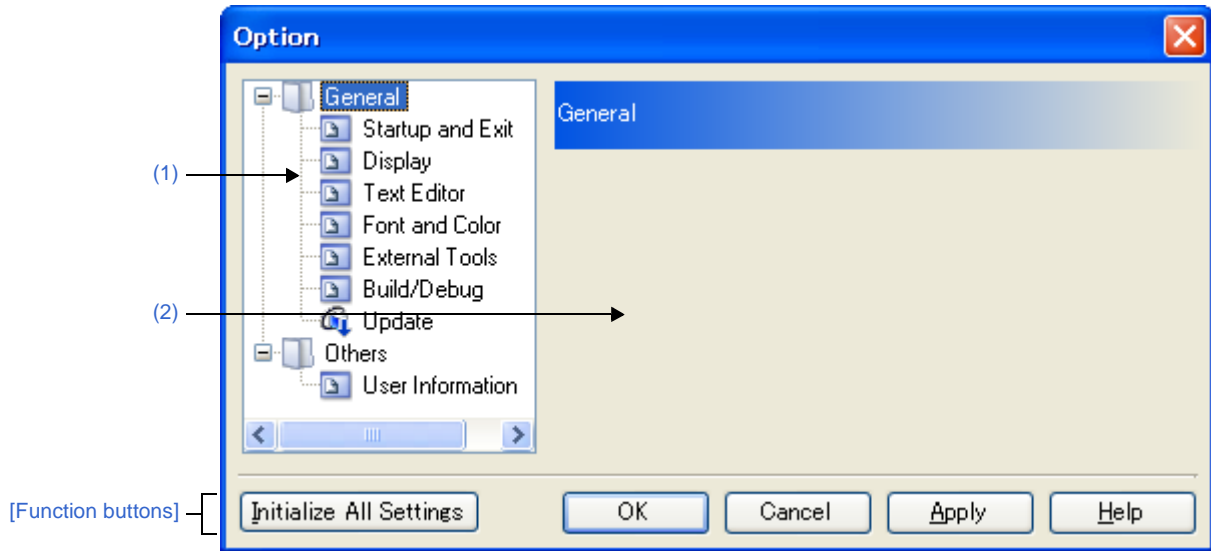
[Function buttons]

Button	Function
Cancel	Cancels the process in progress and closes this dialog box. Note that if the process termination is impossible, this button is disabled.

Option dialog box

This dialog box is used to configure the CubeSuite environment.
 All settings made via this dialog box are saved as preferences for the current user.

Figure A-40. Option Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Tool] menu, select [Option...].

[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 - Text Editor] category	Configure the 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 - Update] category	Configure update.
[Other - User Information] category	Configure user information.

(2) Settings

This area is used to configure the various options for the selected category.

For details about configuration for a particular category, see the section for the category in question.

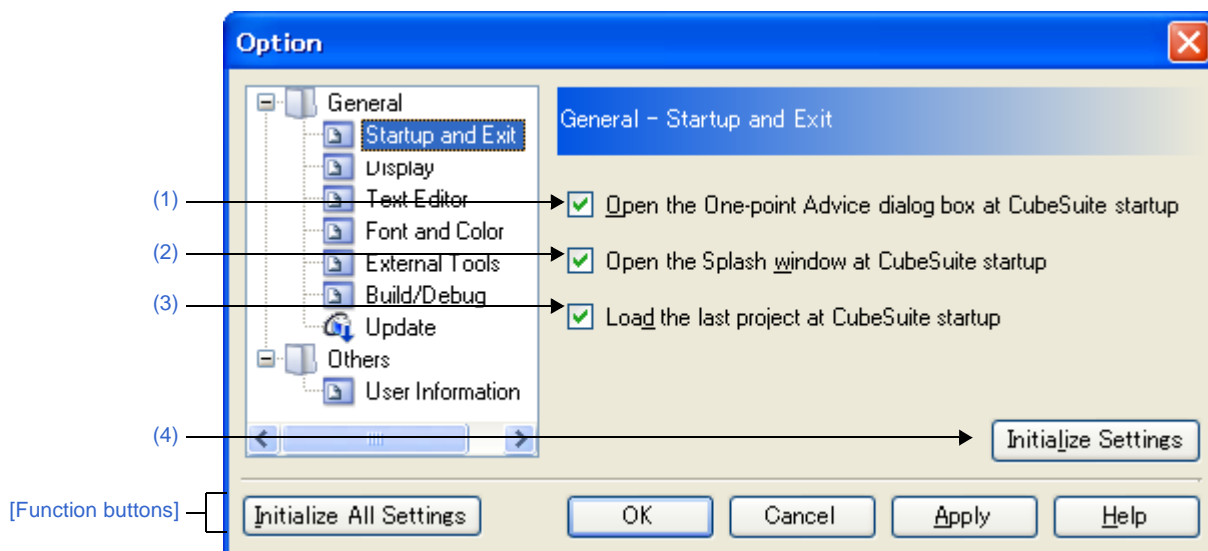
[Function buttons]

Button	Function
Initialize All Settings	Restores all settings on this dialog box to their default values. Note, however, that newly added items in the [General - External Tools] category will not be removed.
OK	Applies all setting and closes this dialog box.
Cancel	Ignores the setting and closes this dialog box.
Apply	Applies all setting (does not close this dialog box).
Help	Displays the help of this dialog box.

[General - Startup and Exit] category

Use this category to configure general settings relating to startup and shutdown.

Figure A-41. Option Dialog Box ([General - Startup and Exit] Category)



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Tool] menu, select [Option...].

[Description of each area]

(1) [Open the One-point Advice dialog box at CubeSuite startup]

<input checked="" type="checkbox"/>	Shows the FormOnePoint dialog box on startup (default).
<input type="checkbox"/>	Does not show the FormOnePoint dialog box on startup.

(2) [Open the Splash window at CubeSuite startup]

<input checked="" type="checkbox"/>	Shows the Splash window on startup (default).
<input type="checkbox"/>	Does not show the Splash window on startup.

(3) [Load the last project at CubeSuite startup]

<input checked="" type="checkbox"/>	Automatically loads the last project on startup (default).
<input type="checkbox"/>	Does not automatically load the last project on startup.

(4) Buttons

Initialize Settings	Returns all currently displayed setting to their default values.
---------------------	--

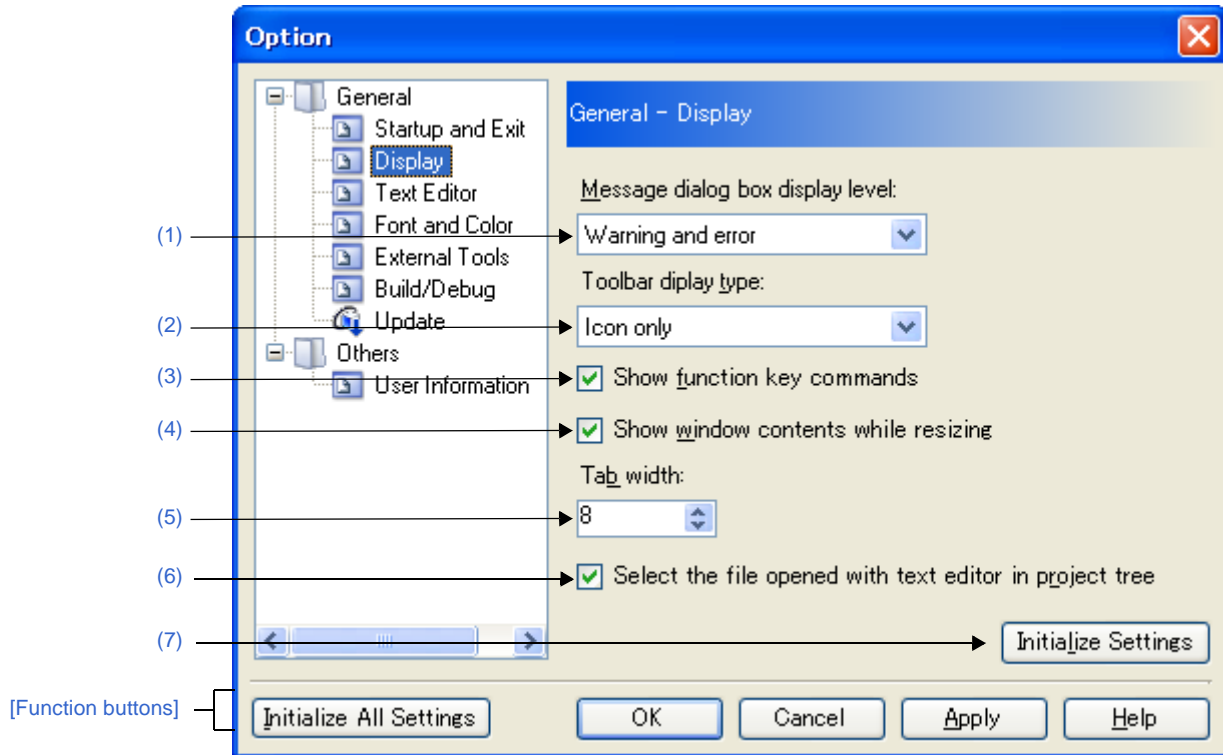
[Function buttons]

Button	Function
Initialize All Settings	Restores all settings on this dialog box to their default values. Note, however, that newly added items in the [General - External Tools] category will not be removed.
OK	Applies all setting and closes this dialog box.
Cancel	Ignores the setting and closes this dialog box.
Apply	Applies all setting (does not close this dialog box).
Help	Displays the help of this dialog box.

[General - Display] category

Use this category to configure general settings relating to program messages.

Figure A-42. Option Dialog Box ([General - Display] Category)



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Tool] menu, select [Option...].

[Description of each area]

(1) [Message dialog box display level]

Select the Message dialog box display level (verbosity) from the following drop-down list. Regardless of this setting, all messages are displayed in the [Output panel](#).

Information, warning and error	Displays all messages in a Message dialog box.
Warning and error	Displays warning and error messages in a Message dialog box (default).
Error only	Only displays error messages in a Message dialog box.
Fatal error only	Only displays fatal error messages in a Message dialog box.

(2) [Toolbar display type]

Use this area to select the format in which to display toolbars on each panel, via the following drop-down list.

Icon only	Displays icons only (default).
Icon and label	Displays both icons and labels (text).

(3) [Show function key commands]


<input checked="" type="checkbox"/>	Displays the Function Key bar in the Main window (default).
<input type="checkbox"/>	Does not display the Function Key bar in the Main window .

(4) [Show window contents while resizing]

<input checked="" type="checkbox"/>	Resizes the window while displaying areas that were not visible before.
<input type="checkbox"/>	Resizes the window without displaying areas that were not visible before (default).

(5) [Tab width]

Specify the number of tab columns.

Either enter a number between 1 and 16 directly via the keyboard, or specify a number via the  buttons. The default is 8.

(6) [Select the file opened with text editor in project tree]

<input checked="" type="checkbox"/>	When the Editor panel is activated, the file currently being opened is selected in the Project Tree panel (default).
<input type="checkbox"/>	Even if the Editor panel is activated, the file currently being opened is not selected in the Project Tree panel .

(7) Buttons

Initialize Settings	Returns all currently displayed setting to their default values.
---------------------	--

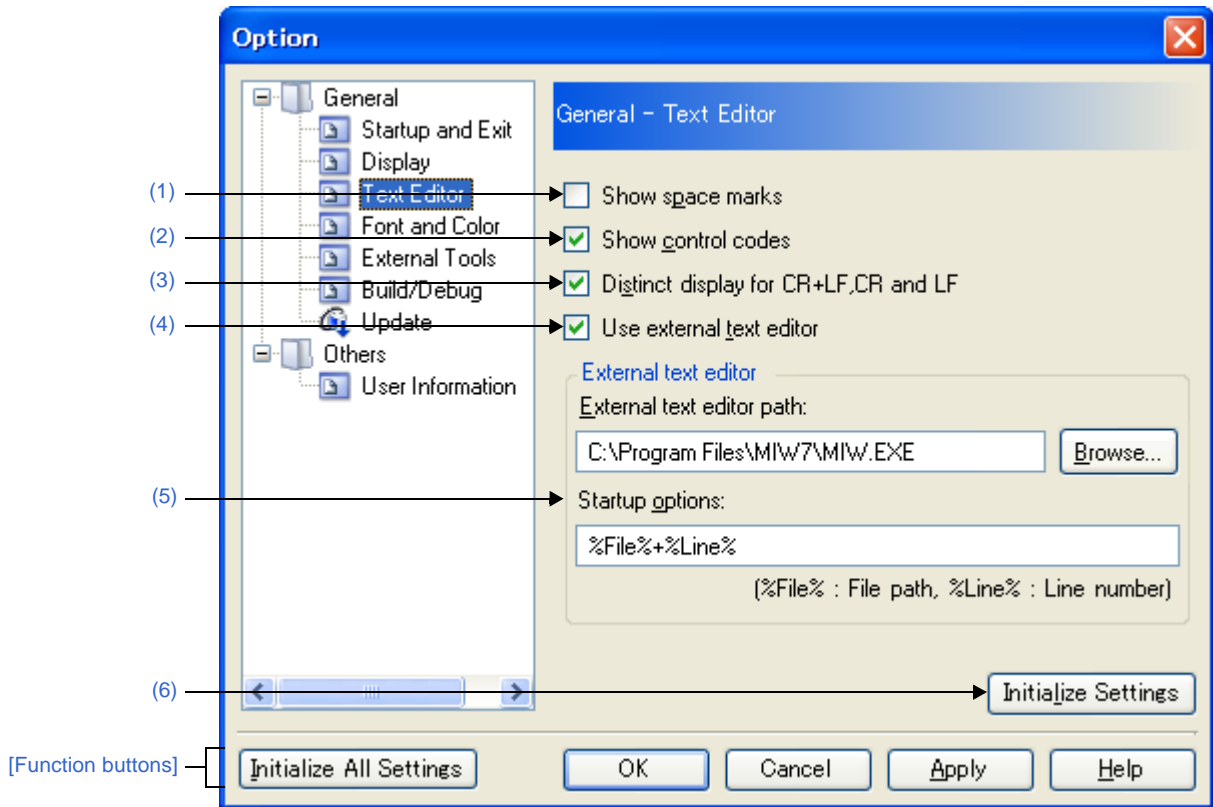
[Function buttons]

Button	Function
Initialize All Settings	Restores all settings on this dialog box to their default values. Note, however, that newly added items in the [General - External Tools] category will not be removed.
OK	Applies all setting and closes this dialog box.
Cancel	Ignores the setting and closes this dialog box.
Apply	Applies all setting (does not close this dialog box).
Help	Displays the help of this dialog box.

[General - Text Editor] category

Use this category to configure general settings relating to the text editor.

Figure A-43. Option Dialog Box ([General - Text Editor] Category)



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Tool] menu, select [Option...].

[Description of each area]

(1) [Show space marks]

<input checked="" type="checkbox"/>	Shows space characters in the text editor (default).
<input type="checkbox"/>	Does not show space characters in the text editor.

(2) [Show control codes]

<input checked="" type="checkbox"/>	Shows control codes (e.g. tab, newline, and EOF) in the text editor (default).
<input type="checkbox"/>	Does not show control codes (e.g. tab, newline, and EOF) in the text editor.

(3) [Distinct display for CR+LF, CR and LF]

This item is only enabled if the [\[Show control codes\]](#) check box is selected.

<input checked="" type="checkbox"/>	Visually distinguishes line break types (CR+LF, CR, and LF) in the text editor (default).
<input type="checkbox"/>	Does not visually distinguish line break types (CR+LF, CR, and LF) in the text editor.

(4) [Use external text editor]

<input checked="" type="checkbox"/>	Uses an external text editor. The [External text editor] area is enabled. Use it to specify the external text editor to use.
<input type="checkbox"/>	Uses the Editor panel as the text editor (default).

(5) [External text editor] area

This area is only enabled if the [\[Use external text editor\]](#) check box is selected.

Use this area to specify the external text editor to use, and the startup options.

(a) [External text editor path]

Either type in the name of the executable file (including absolute path) for the external text editor directly via the keyboard (up to 259 characters), or click the [\[Browse...\]](#) button, and in the [Select External Text Editor dialog box](#), specify the name of the executable file.

(b) [Startup options]

Specify the startup options for the external text editor (up to 256 characters).

The startup option variables that can be specified here are as follows.

If you wish to specify more than one option, separate them by spaces.

%File%	Passes the name of the file to display (with absolute path) to the external text editor.
%Line%	Passes the caret position (line number) to the external text editor.

(6) Buttons

Initialize Settings	Returns all currently displayed setting to their default values.
---------------------	--

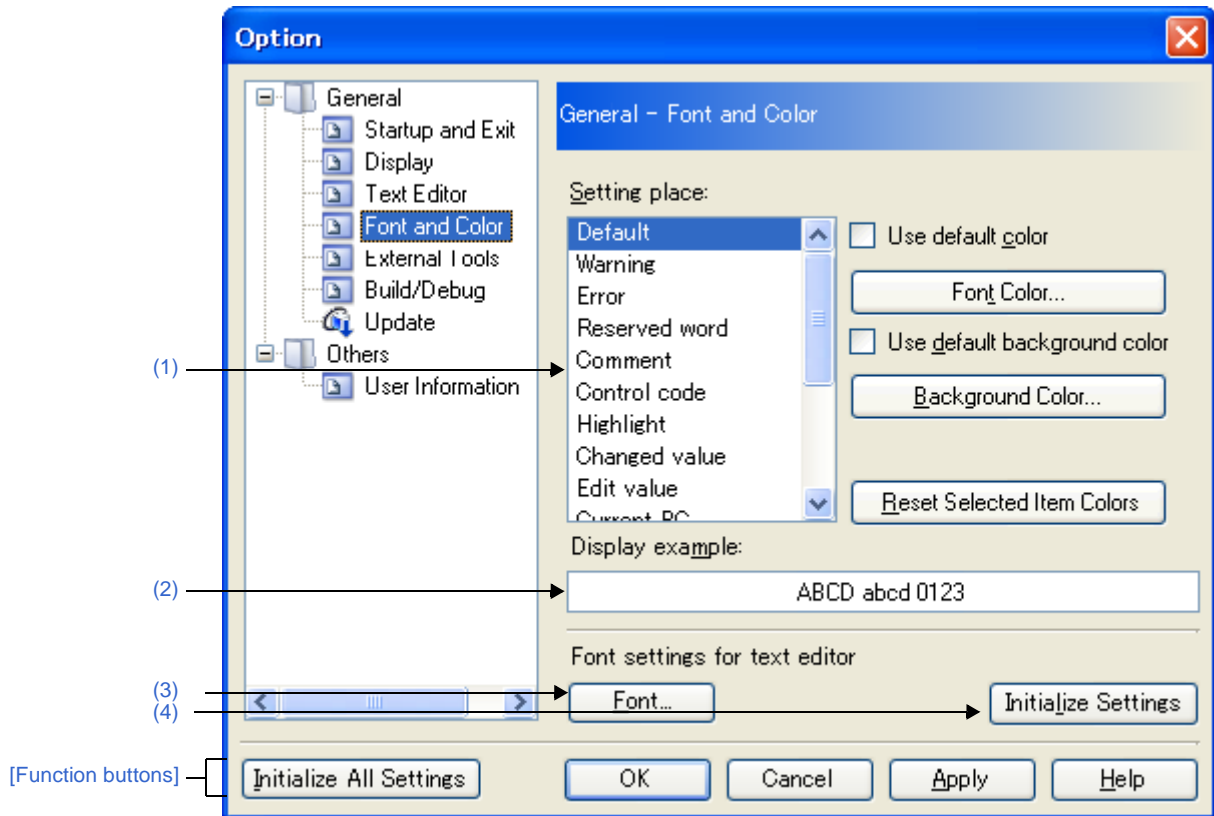
[Function buttons]

Button	Function
Initialize All Settings	Restores all settings on this dialog box to their default values. Note, however, that newly added items in the [General - External Tools] category will not be removed.
OK	Applies all setting and closes this dialog box.
Cancel	Ignores the setting and closes this dialog box.
Apply	Applies all setting (does not close this dialog box).
Help	Displays the help of this dialog box.

[General - Font and Color] category

Use this category to configure general settings relating to fonts and colors on each panel.

Figure A-44. Option Dialog Box ([General - Font and Color] Category)



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Tool] menu, select [Option...].

[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	The standard display color in all windows and panels.
		Background color	White	
Warning	AaBbCc	Font color	Blue	Warning messages are displayed in this color in the Output panel , and file names with warnings are displayed in this color in the Project Tree panel .
		Background color	Default color	
Error	AaBbCc	Font color	Red	Error messages are displayed in this color in the Output panel , and file names with errors are displayed in this color in the Project Tree panel .
		Background color	LightGray	
Reserved word	AaBbCc	Font color	Brown	The reserved words of your compiler/ assembler are displayed in this color in the Editor panel.
		Background color	Default color	
Comment	AaBbCc	Font color	Green	Comments (in the case of a C source file, <code>/* ... */</code>) are displayed in this color in the Editor panel.
		Background color	Default color	
String literal	AaBbCc	Font color	Gray	String literals are displayed in this color in the Editor panel.
		Background color	Default color	
Control code	AaBbCc	Font color	Teal	Control characters are displayed in this color in the Editor panel.
		Background color	Default color	
Highlight	AaBbCc	Font color	White	Highlighted areas in plug-in products and the like are displayed in this color.
		Background color	Magenta	
Changed value	AaBbCc	Font color	Tan	Values changed via the execution of a user program are displayed in this color in the Memory panel, CPU Register panel, Local Variables panel, SFR panel, Watch panel.
		Background color	Cream	
Edit value	AaBbCc	Font color	Blue	Values forcibly modified by the user are displayed in this color in the Memory panel, CPU Register panel, Local Variables panel, SFR panel, Watch panel.
		Background color	Default color	
Current PC	AaBbCc	Font color	Black	The row with the current PC position is displayed in this color in the Editor panel.
		Background color	LightOrange	
Breakpoint	AaBbCc	Font color	Black	The line at which a break point is set is displayed in this color in the Editor panel.
		Background color	SalmonPink	
Update periodic	AaBbCc	Font color	Pink	Areas configured for real-time display updates are shown in this color in the Memory panel and Watch panel.
		Background color	Default color	
Read or fetch	AaBbCc	Font color	Default color	Locations that have been read or fetched are displayed in this color in the Trace panel [IECUBE] [Simulator].
		Background color	LightGreen	
Write	AaBbCc	Font color	Default color	Locations that have been written are displayed in this color in the Trace panel [IECUBE] [Simulator].
		Background color	Orange	

Item	Example		Description
Read and write		Font color	Default color
		Background color	LightSkyBlue
Coverage 100%		Font color	Default color
		Background color	LightGreen
Coverage 1 - 99%		Font color	Default color
		Background color	LightPink
Coverage 0%		Font color	Default color
		Background color	LightGray
Invalid		Font color	Gray
		Background color	Default color

Note The [Default] text and background colors depend 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]

<input checked="" type="checkbox"/>	Displays items selected via the [Setting place] area using the standard text color.
<input type="checkbox"/>	Displays items selected via the [Setting place] area with a user-defined text color. The [Font color...] button is enabled.

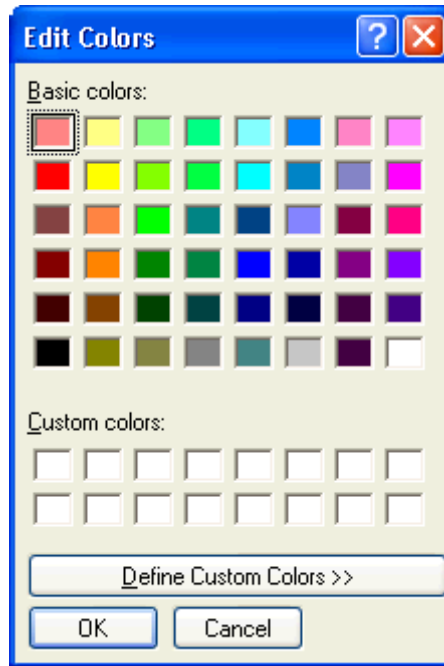
(c) [Use default background color]

<input checked="" type="checkbox"/>	Displays items selected via the [Setting place] area using the standard background color.
<input type="checkbox"/>	Displays 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	Resets the color information for the item selected via the [Setting place] area to the defaults.

Figure A-45. Edit Colors Dialog Box



(2) [Display example] area

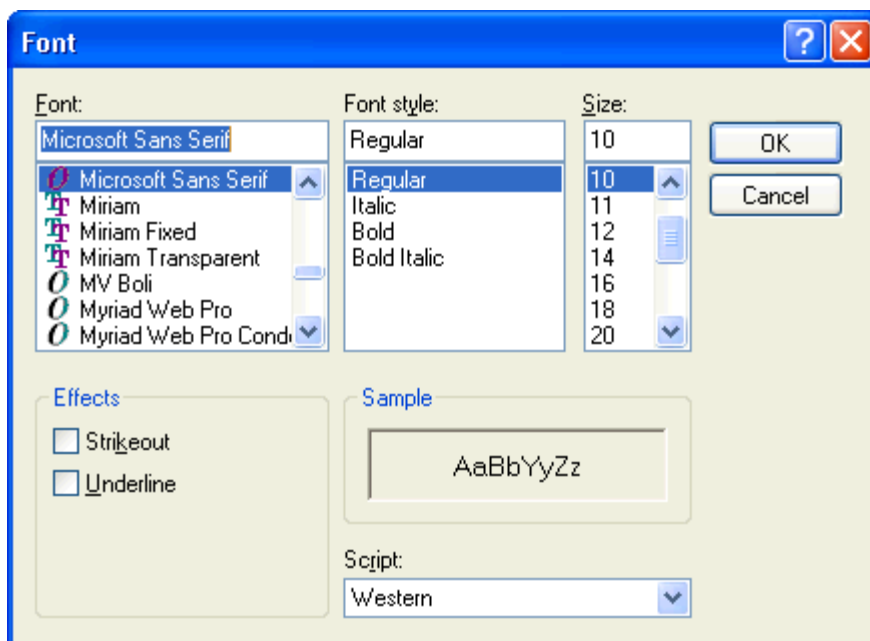
Display sample text using the color and font settings from the [Color options area](#) and the [\[Font settings for text editor\] area](#).

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-46. Font Dialog Box



(4) Buttons

Initialize Settings	Returns all currently displayed setting to their default values.
---------------------	--

[Function buttons]

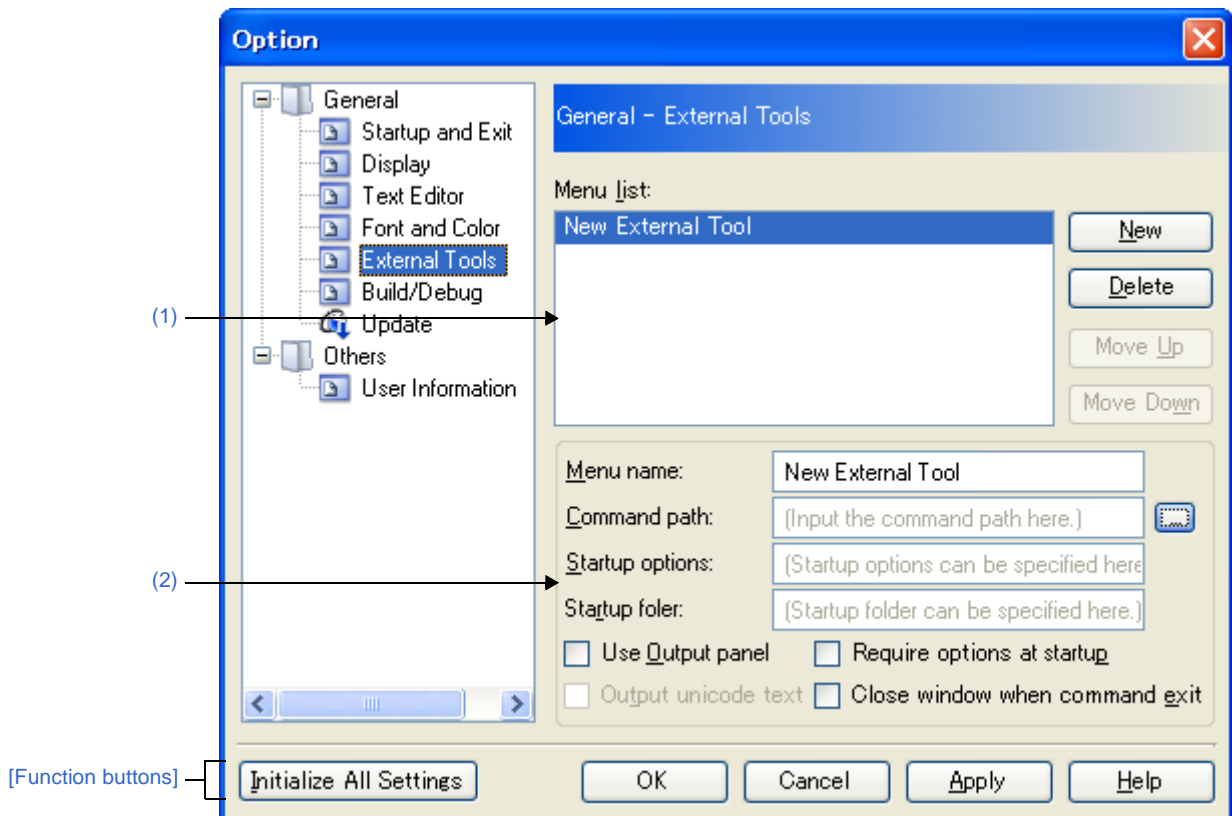
Button	Function
Initialize All Settings	Restores all settings on this dialog box to their default values. Note, however, that newly added items in the [General - External Tools] category will not be removed.
OK	Applies all setting and closes this dialog box.
Cancel	Ignores the setting and closes this dialog box.
Apply	Applies all setting (does not close this dialog box).
Help	Displays the help of this dialog box.

[General - External Tools] category

Use this category to register external tools that can be launched directly from CubeSuite, and configure these external tools when they are so launched.

Registering an external tool here allows it to be launched directly via the [Tool] menu.

Figure A-47. Option Dialog Box ([General - External Tools] Category)



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Tool] menu, select [Option...].

[Description of each area]

(1) Registered external tools area

(a) [Menu list] area

This area displays a list of menu items (added to the [Tool] menu) for launching external tools that have been added via this dialog box.

To add a new external tool, click the [New] button.

When you click the [New] button, the item "New External Tool" is added to the bottom of the list. In this state, configure the details of the external tool to add in the [New registration area](#).

Up to 8 external tools can be registered.

To change the settings of an external tool that has already been registered, select the name of the external tool to modify and edit the desired setting.

The order of the items on this list is the same as the order in the menu.

(b) Buttons

New	Adds the item "New External Tool" to the bottom of the list for the registration of a new external tool. Note that this button will be disabled if 8 external tools have already been registered.
Delete	Removes the external tool that is selected in the list.
Move Up	Moves the external tool selected in the list up one row.
Move Down	Moves the external tool selected in the list down one row.

(2) New registration area

Use this area to configure the details of a newly added external tool.

Up to 8 external tools can be registered.

(a) [Menu name]

Specify the name to use in the menu for launching the external tool (up to 100 characters).

The name specified here will appear in the [Tool] menu, selecting that item will launch the external tool.

When you finish entering the menu name (the text box loses focus), the name automatically replaces the "New External Tool" string in the list.

(b) [Command path]

Either type in the name of the executable file (including absolute path) for the new external tool directly via the keyboard (up to 259 characters), or click the [...] button, and in the [Select Program dialog box](#), specify the name of the executable file ([[Startup folder](#)] is automatically set to the absolute path to the folder where the executable file is located).

(c) [Startup options]

Specify the startup options for the new external tool (up to 247 characters).

The startup option variables that can be specified here are as follows.

If you wish to specify more than one option, separate them by spaces.

%File%	The name of the file currently selected in the Project Tree panel (with absolute path) is passed to the external tool. If more than one file is selected, the information for the file in which the cursor is located is passed).
%OutputFile%	The program passes to the external tool the name of the module file (with absolute path) that is output when the project for the item currently selected in the Project Tree panel is built (if there is no selection, then nothing is passed).
%Line%	If the file currently selected in the Project Tree panel is being edited in the Editor panel, then the line on which the caret is located is passed (if the Editor panel does not have focus, then "1" is passed).

(d) [Startup folder]

Specify the absolute path to the folder for launching the external tool (up to 247 characters).

The start folder variables that can be specified here are as follows.

If you wish to specify more than one option, separate them by spaces.

%FileDir%	The name of the folder for the file currently selected in the Project Tree panel (with absolute path) is passed to the external tool. If more than one file is selected, the information for the file in which the cursor is located is passed (if there is no selection, then nothing is passed).
%OutputDir%	The program passes the name of the folder (with absolute path) for the module that is output when the project for the item currently selected in the Project Tree panel is built (if there is no selection, then nothing is passed).
%ProjectDir%	The program passes the name of the folder (with absolute path) of the project for the item currently selected in the Project Tree panel (if there is no selection, then nothing is passed).

(e) [Use Output panel]

<input checked="" type="checkbox"/>	Displays messages that the new external tool outputs to stdout and stderr in the Output panel . Messages will be output to the Output panel 's [Tool Name] tab ("Tool Name" is the string specified in [Menu name]).
<input type="checkbox"/>	Uses the Windows command prompt for messages that the new external tool outputs to stdout and stderr (default).

(f) [Require options at startup]

<input checked="" type="checkbox"/>	The Character String Input opens, enabling you to specify startup parameters for the new external tool.
<input type="checkbox"/>	Does not specify startup parameters for the new external tool (default).

(g) [Output unicode text]

This item is only enabled if the [\[Use Output panel\]](#) check box is selected.

<input checked="" type="checkbox"/>	Displays messages that the new external tool outputs to stdout and stderr in the Output panel using the Unicode character set.
<input type="checkbox"/>	Displays messages that the new external tool outputs to stdout and stderr in the Output panel using the ASCII character set (default).

(h) [Close window when command exit]

This item is disabled if the [\[Use Output panel\]](#) check box is selected.

<input checked="" type="checkbox"/>	Closes the Windows command prompt when the new external tool exits.
<input type="checkbox"/>	Leaves the Windows command prompt open after the new external tool exits (default).

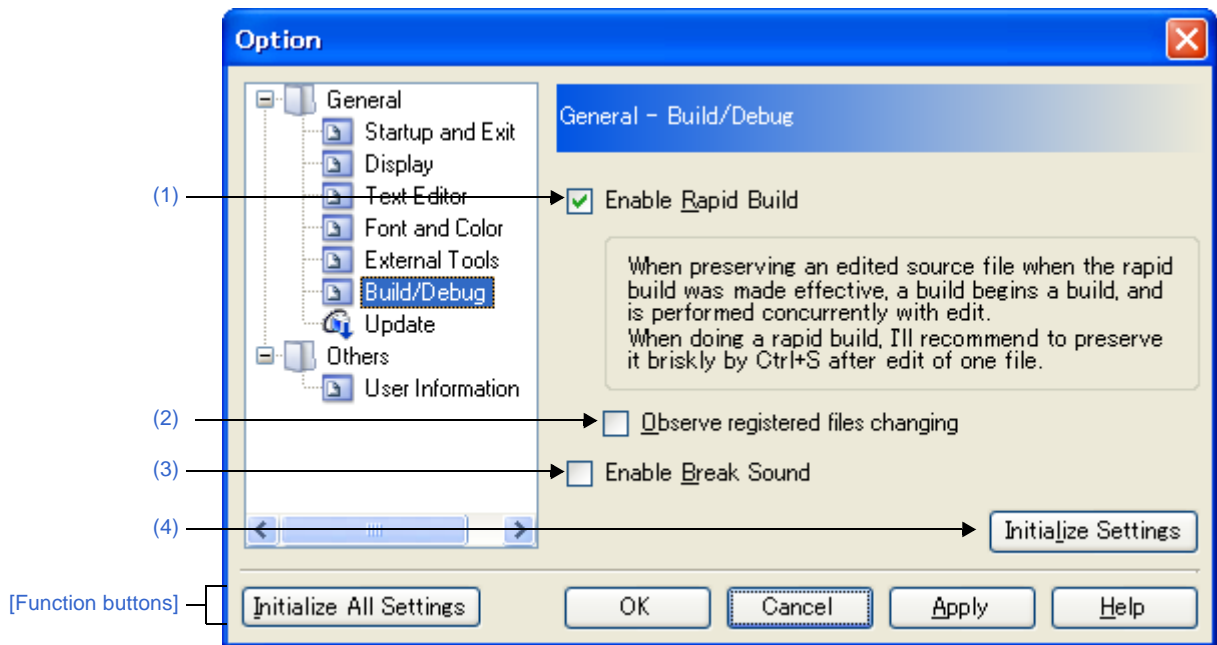
[Function buttons]

Button	Function
Initialize All Settings	Restores all settings on this dialog box to their default values. Note, however, that newly added items in the [General - External Tools] category will not be removed.
OK	Applies all setting and closes this dialog box.
Cancel	Ignores the setting and closes this dialog box.
Apply	Applies all setting (does not close this dialog box).
Help	Displays the help of this dialog box.

[General - Build/Debug] category

Use this category to configure general setting relating to building and debugging.

Figure A-48. Option Dialog Box ([General - Build/Debug] Category)



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Tool] menu, select [Option...].

[Description of each area]

(1) [Enable Rapid Build]

<input checked="" type="checkbox"/>	Enables the rapid build ^{Note} feature (default).
<input type="checkbox"/>	Does 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. For details on the rapid build feature, see the following.

- When the target project type is other than a debug-dedicated project: "CubeSuite Build"
- When the target project type is a debug-dedicated project: "E.5.4 Run a rapid build"

(2) [Observe registered files changing]

<input checked="" type="checkbox"/>	Start s a rapid build when a source file registered in the project is edited or saved by an external text editor or the like.
<input type="checkbox"/>	Does 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.

Caution Files that are in the project folder and have been registered to the project can be monitored. 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.

(3) [Enable Break Sound]

<input checked="" type="checkbox"/>	Beeps when the execution of a user program is halted due to a break event (hardware or software break).
<input type="checkbox"/>	Does not beep when the execution of a user program is halted due to a break event (hardware or software break) (default).

(4) Buttons

Initialize Settings	Returns all currently displayed setting to their default values.
---------------------	--

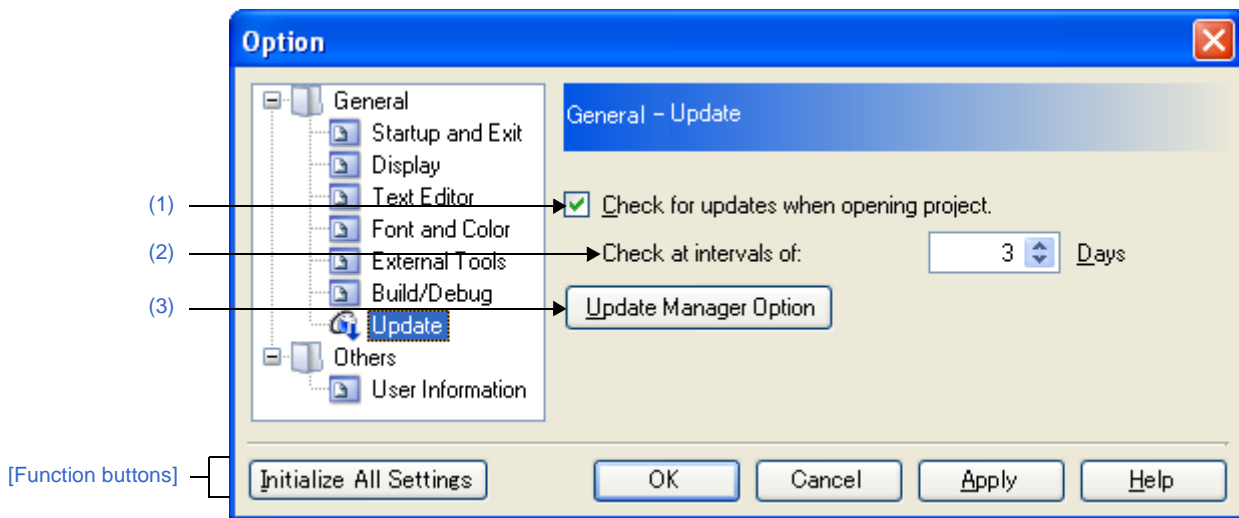
[Function buttons]

Button	Function
Initialize All Settings	Restores all settings on this dialog box to their default values. Note, however, that newly added items in the [General - External Tools] category will not be removed.
OK	Applies all setting and closes this dialog box.
Cancel	Ignores the setting and closes this dialog box.
Apply	Applies all setting (does not close this dialog box).
Help	Displays the help of this dialog box.

[General - Update] category

Use this category to configure general setting relating to update.

Figure A-49. Option Dialog Box ([General - Update] Category)



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Tool] menu, select [Option...].

[Description of each area]

(1) [Check for updates when opening project.]

<input checked="" type="checkbox"/>	Automatically checks for updates when a project is opened.
<input type="checkbox"/>	Does not automatically check for updates when a project is opened.

(2) [Check at intervals of]

Specify the interval at which to check for updates.

Either enter a number between 0 and 99 directly via the keyboard, or specify a number via the buttons.

(3) Buttons

Update Manager Option	Displays the Update Manager Options dialog box .
-----------------------	--

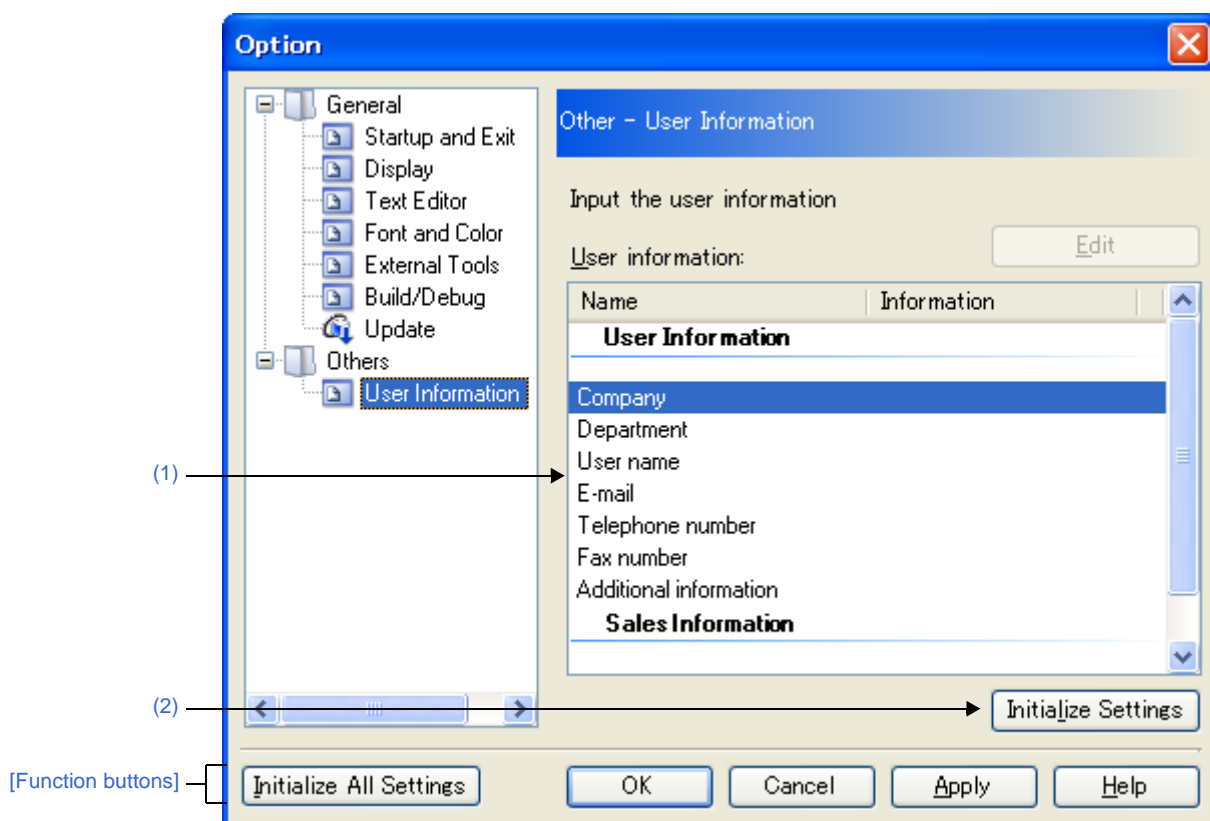
[Function buttons]

Button	Function
Initialize All Settings	Restores all settings on this dialog box to their default values. Note, however, that newly added items in the [General - External Tools] category will not be removed.
OK	Applies all setting and closes this dialog box.
Cancel	Ignores the setting and closes this dialog box.
Apply	Applies all setting (does not close this dialog box).
Help	Displays the help of this dialog box.

[Other - User Information] category

Use this category to configure other setting relating to user information.

Figure A-50. Option Dialog Box ([Other - User Information] Category)



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Tool] menu, select [Option...].

[Description of each area]

(1) Information area

(a) [User information] area

This area displays a list of user information.

The content in the [Information] field can be edited. To edit the information, select one of the items from the list, click the [Edit] button, and then type the information directly into the text box (up to 256 characters).

(b) Buttons

Edit	Edits the content of the selected [Information] item by typing directly in the text box. This button is disabled if nothing is selected in the list.
------	---

(2) Buttons

Initialize Settings	Returns all currently displayed setting to their default values.
---------------------	--

[Function buttons]

Button	Function
Initialize All Settings	Restores all settings on this dialog box to their default values. Note, however, that newly added items in the [General - External Tools] category will not be removed.
OK	Applies all setting and closes this dialog box.
Cancel	Ignores the setting and closes this dialog box.
Apply	Applies all setting (does not close this dialog box).
Help	Displays the help of this dialog box.

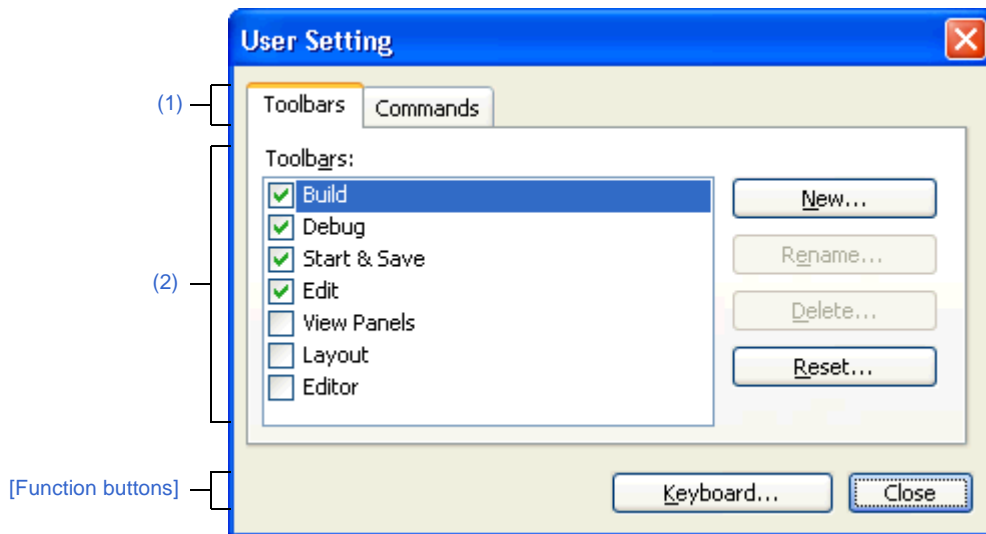
User Setting dialog box

This dialog box allows you to customize toolbars and menus displayed in the [Main window](#).

When this dialog box is open, any button on a toolbar or any menu item in a menu bar currently displayed in the [Main window](#) can be dragged and dropped to the desired position to change the sequence of buttons or menu items or perform button/menu item deletion.

Caution While this dialog box is open, you cannot use functions of toolbars/menus.
You can use these functions as usual after you close this dialog box.

Figure A-51. User Setting Dialog Box



The following items are explained here.

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

[How to open]

- From the [Tool] menu, select [User Setting...].

[Description of each area]

(1) Tab selection area

Tab selection allows you to switch between the customization targets.

This dialog box has the following tabs:

- [\[Toolbars\] tab](#)
- [\[Commands\] tab](#)

(2) Customization area

You can set detailed customization conditions.

For details of displayed items/setting method, see the description of each tab.

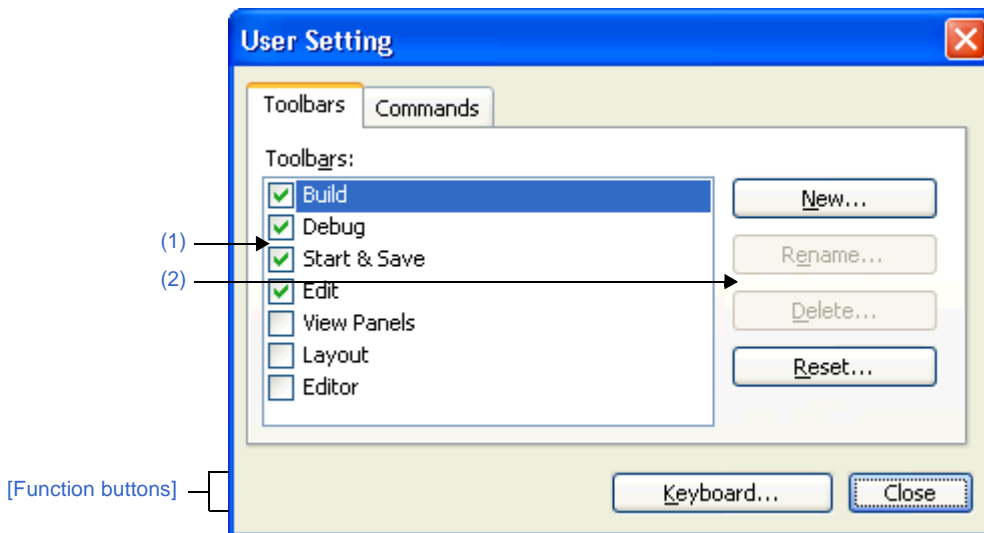
[Function buttons]

Button	Function
Keyboard...	Opens the Customize Keyboard dialog box to assign customized items to keys on the keyboard.
Close	Cancels the toolbar/menu customization setting and closes this dialog box.

[Toolbars] tab

You can set whether toolbars are displayed or not, change toolbar names, and make new toolbars.

Figure A-52. User Setting Dialog Box: [Toolbars] Tab



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Tool] menu, select [User Setting...].

[Description of each area]

(1) [Toolbars] area

Display a list of the names of registered toolbars.

A check mark appears in front of the name of the toolbar currently displayed in the [Main window](#). By removing the check mark, you make a setting so that the toolbar will not be displayed.

(2) Buttons

New...	<p>Opens the New Toolbar dialog box to make a new toolbar.</p> <p>The name of the new toolbar will be added to the list displayed in the [Toolbars] area with a check mark placed in front.</p> <p>Note that there are no buttons on the newly created toolbar. The register buttons on it, use the [Commands] tab.</p>
Rename...	<p>Opens the Rename Toolbar dialog box to change the name of the currently selected toolbar.</p> <p>This function is disabled when a toolbar other than those created by the user is selected.</p>

Delete...	Deletes the selected toolbar. This function is disabled when a toolbar other than those created by the user is selected.
Reset...	Discards all changes made to the selected toolbar and returns it to the default state. When a toolbar created by the user is selected, this button returns the selected toolbar to the state with no buttons registered on it.

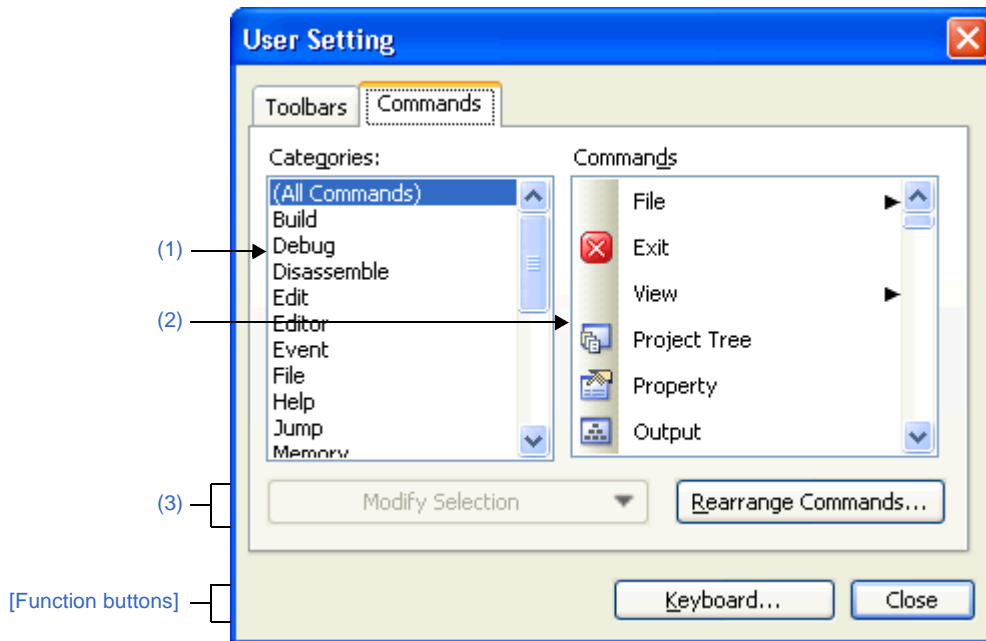
[Function buttons]

Button	Function
Keyboard...	Opens the Customize Keyboard dialog box to assign customized items to keys on the keyboard.
Close	Cancels the toolbar/menu customization settings and closes this dialog box.

[Commands] tab

You can customize items to include on a toolbar or a menu.

Figure A-53. User Setting Dialog Box: [Commands] Tab



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Tool] menu, select [User Setting...].

[Description of each area]**(1) [Categories] area**

Display a list of the categories of commands that CubeSuite provides.

(2) [Commands] area

Display a list of the names of commands belonging to the category selected in the [Categories] area, together with their icons (if exist).

When "(All Commands)" is selected in the [Categories] area, the name of all commands that CubeSuite provides are displayed, together with their icons (if exist).

To add a command on a toolbar/menu, drag and drop the command name in this area onto the toolbar/menu displayed in the actual [Main window](#).

(3) Buttons

Modify Selection	In order that the menu item or the button on a toolbar currently selected in the Main window , displays the following menu items beneath this button:	
	Menu Item	Description
	Reset	Resets the selected menu item/button.
	Delete	Deletes the selected menu item/button.
	Name	Displays the name of the selected menu item/button.
	Default Style	Checking this menu item returns the display style of the selected menu item/button to the default state (by default).
	Text Only (Always)	Checking this menu item displays the selected menu item/button by text only (the icon will not be displayed).
	Text Only (in Menu)	This is enabled only when a menu item is selected. Checking this menu item displays the selected menu item/button by text only (the icon will not be displayed).
	Image and Text	Checking this menu item displays the selected menu item/button by both text and icon.
	Begin a Group	Inserts separator just before the selected menu item/button.
Recently Used	This item is not supported in this version.	
Rearrange Commands...	Opens the Rearrange Commands dialog box for changing the arrangement (including addition and deletion) of menu items and toolbar buttons in the Main window .	

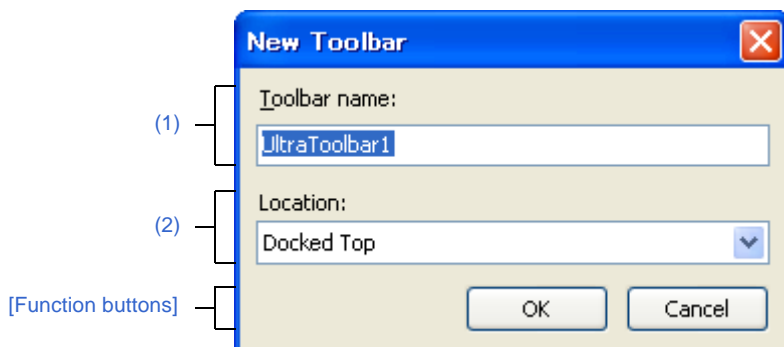
[Function buttons]

Button	Function
Keyboard...	Opens the Customize Keyboard dialog box to assign customized items to keys on the keyboard.
Close	Cancels the toolbar/menu customization setting and closes this dialog box.

New Toolbar dialog box

This dialog box is used to create a new toolbar to appear in the [Main window](#).

Figure A-54. New Toolbar Dialog Box



The following items are explained here.

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

[How to open]

- In the [\[Toolbars\]](#) tab of the [User Setting dialog box](#), click the [\[New...\]](#) button.

[Description of each area]

(1) [Toolbar name]

Type in the name of the new toolbar directly via the keyboard.
 "UltraToolbar1" is specified by default.

(2) [Location]

Select the location for the new toolbar from the following drop-down list.
 The location specified here is the location where the new toolbar will appear immediately after it is created (toolbars can be moved freely by dragging and dropping).

Docked Top	Displays the toolbar at the top of the Main window (default).
Docked Bottom	Displays the toolbar at the bottom of the Main window .
Docked Left	Displays the toolbar on the leftedge of the Main window .
Docked Right	Displays the toolbar on the rightedge of the Main window .
Floating	Displays the toolbar above the Main window , without docking it.

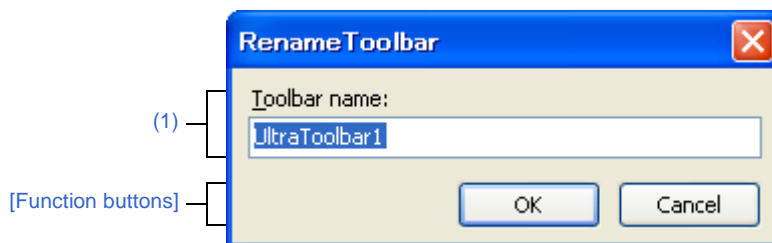
[Function buttons]

Button	Function
OK	Creates a new toolbar with the specified information, and closes this dialog box. The new toolbar appears in the list on the User Setting dialog box's [Toolbars] tab , with its check box selected.
Cancel	Ignores the setting and closes this dialog box.

Rename Toolbar dialog box

This dialog box is used to edit the name of a toolbar created by the user.

Figure A-55. Rename Toolbar Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- In the [Toolbars] tab of the User Setting dialog box, select the name of a user toolbar and then click the [Rename...] button.

[Description of each area]

(1) [Toolbar name]

Edit the toolbar name directly via the keyboard.
By default, the name of the currently selected toolbar is shown.

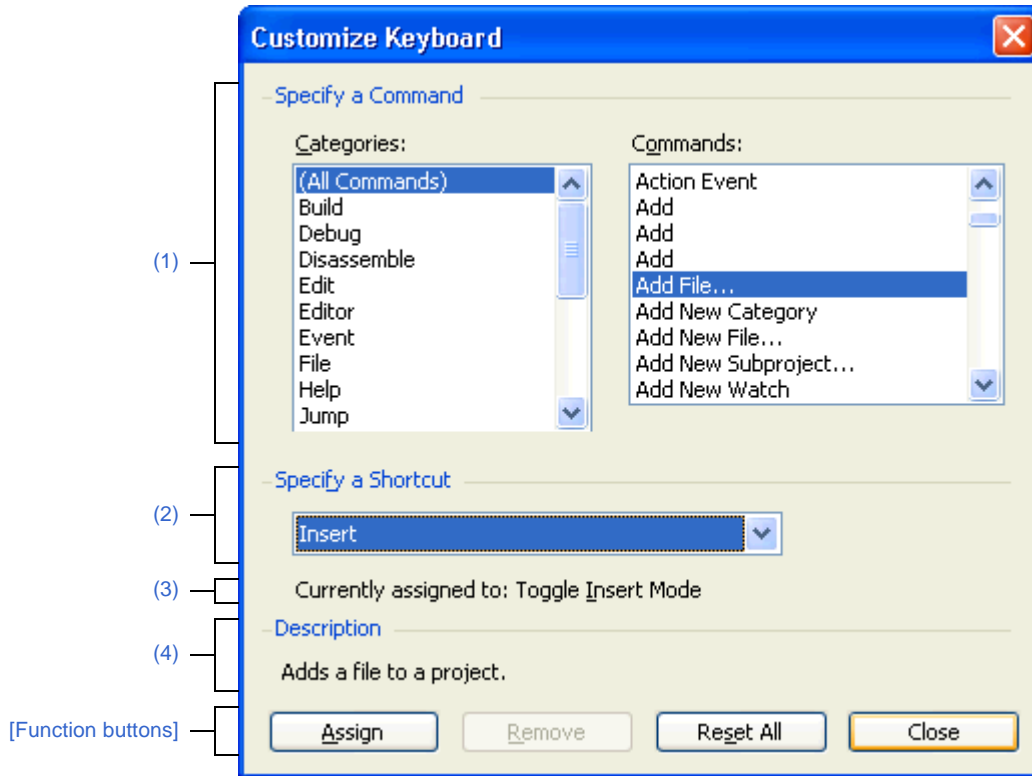
[Function buttons]

Button	Function
OK	Changes the selected toolbar to the specified name, and closes this dialog box.
Cancel	Ignores the setting and closes this dialog box.

Customize Keyboard dialog box

This dialog box is used to assign shortcut keys to the various commands.

Figure A-56. Customize Keyboard Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- In the [Toolbars] tab of the User Setting dialog box, click the [Keyboard...] button.

[Description of each area]

(1) [Specify a Command] area

(a) [Categories]

Display a list of the categories of commands provided by CubeSuite.

(b) [Commands]

Display a list of the commands belonging to the category selected under [Categories] and their associated icons (if any).

If "(All Commands)" is selected under [Categories], then all commands provided by CubeSuite appear, with their associated icons (if any).

(2) [Specify a Shortcut] area

This area displays the default shortcut key currently assigned to the command selected under [Commands] (if no keys area assigned, then "None" appears).

To change the assigned shortcut key, select a key from the following drop-down list, and then click the [Assign] button.

None	Shift + F1 to F12	Ctrl + Shift + 0 to 9	Alt + Right
Insert	Ctrl + Insert	Ctrl + Shift + A to Z	Alt + Down
Delete	Ctrl + Delete	Ctrl + Shift + F1 to F12	Alt + 0 to 9
F1 to F12	Ctrl + 0 to 9	Alt + Backspace	Alt + F1 to F12
Shift + Insert	Ctrl + A to Z	Alt + Left	
Shift + Delete	Ctrl + F1 to F12	Alt + Up	

(3) [Current assigned to]

This area displays the command currently assigned to the shortcut key specified in the [Specify a Shortcut] area (if no commands are assigned to this key, then "None" appears).

(4) [Description] area

This area displays a popup describing the function of the command selected under [Commands].

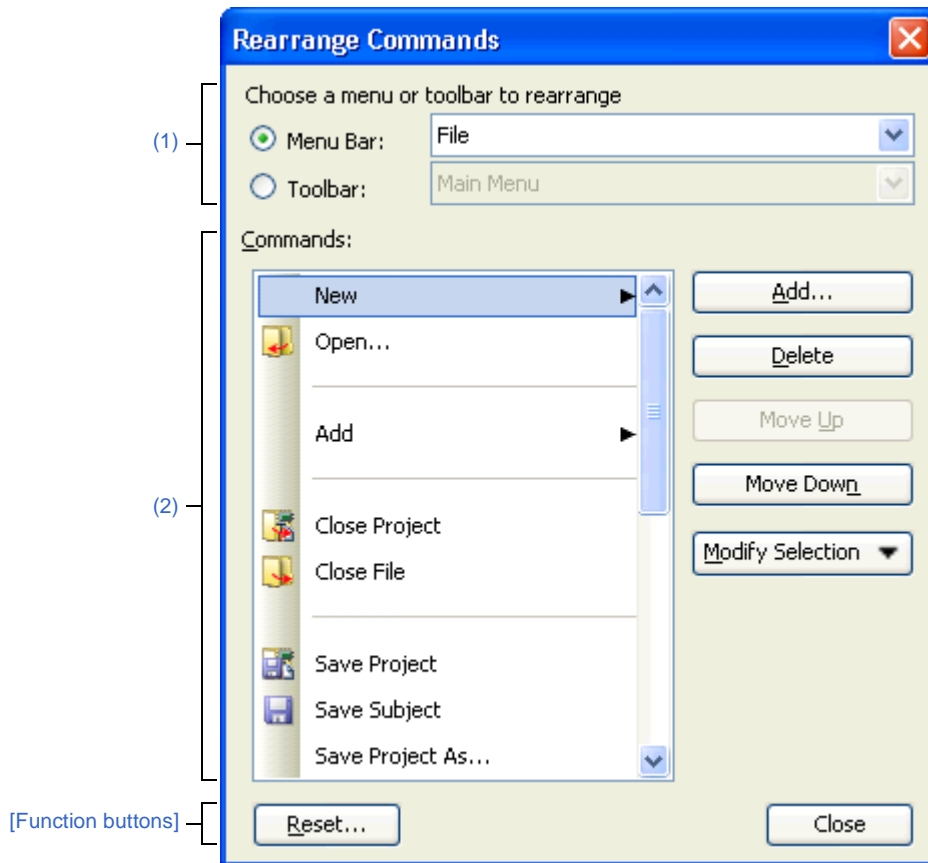
[Function buttons]

Button	Function
Assign	Assigns the shortcut key selected under [Commands] to the command selected under the [Specify a Shortcut] area. Note, however, that this button will be disabled if the key selected in the [Specify a Shortcut] area is already assigned to another command.
Remove	Removes the assignment of the shortcut key selected under the [Specify a Shortcut] area to the command selected under [Commands] ("None" will appear in the [Specify a Shortcut] area drop-down list). Note, however, that this button will be disabled if no keys have been assigned to the command selected under [Commands].
Reset All	Resets all shortcut key setting to their default values.
Close	Ends the shortcut key assignment and closes this dialog box.

Rearrange Commands dialog box

This dialog box allows you to change the arrangement (including addition and deletion) of menu items and buttons in the [Main window](#).

Figure A-57. Rearrange Commands Dialog Box



The following items are explained here.

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

[How to open]

- In the [\[Commands\]](#) tab of the [User Setting dialog box](#), click the [\[Rearrange Commands...\]](#) button.

[Description of each area]

(1) [Choose a menu or toolbar to rearrange] area

This area allows you to specify the item whose position you want to change.

First select the [\[Menu Bar\]](#) if you want to change the menu item or [\[Toolbar\]](#) if you want to change the toolbar button using the option button, and then select the category to be changed from the drop-down list.

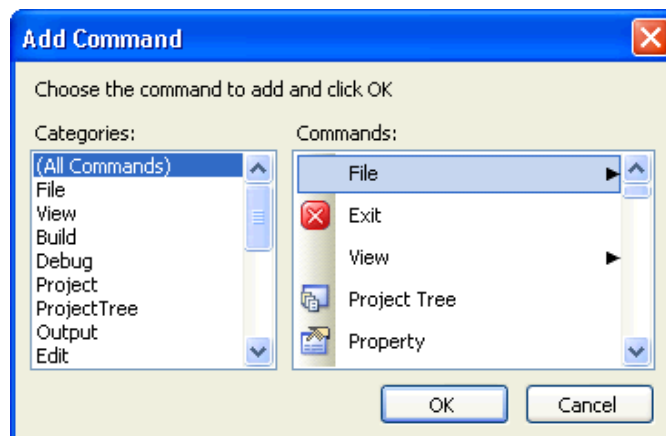
(2) [Commands] area

This area displays a list of commands belonging to the category selected in the [Choose a menu or toolbar to rearrange] area which will be displayed in the Main window.

You can change the arrangement of these commands using the following buttons in this area.

Add...	Opens the Add Command Dialog Box for selecting a command to be added above the command currently selected in this area. In the Add Command Dialog Box , all commands that can be added are displayed, grouped by their categories. First select the category of command in the [Categories] area, then select the command you want to add and press the [OK] button (pressing the [Cancel] button cancels the addition of the command and closes this dialog box).	
Delete	Deletes the command currently selected in this area.	
Move Up	Moves the command currently selected in this area up one line.	
Move Down	Moves the command currently selected in this area down one line.	
Modify Selection	Displays the following menu items for editing the arrangement of the menu item or button currently selected in this area.	
	Menu Item	Description
	Reset	Resets the selected menu item/button.
	Delete	Deletes the selected menu item/button.
	Name	Displays the name of the selected menu item/button.
	Default Style	Checking this menu item returns the display style of the selected menu item/button to the default state (by default).
	Text Only (Always)	Checking this menu item displays the selected menu item/button by text only (the icon will not be displayed).
	Text Only (in Menus)	This is enabled only when a menu item is selected. Checking this menu item displays the selected menu item/button by text only (the icon will not be displayed).
	Image and Text	Checking this menu item displays the selected menu item/button by both text and icon.
	Begin a Group	Inserts separator just before the selected menu item/button.
	Recently Used	This item is not supported in this version.

Figure A-58. Add Command Dialog Box



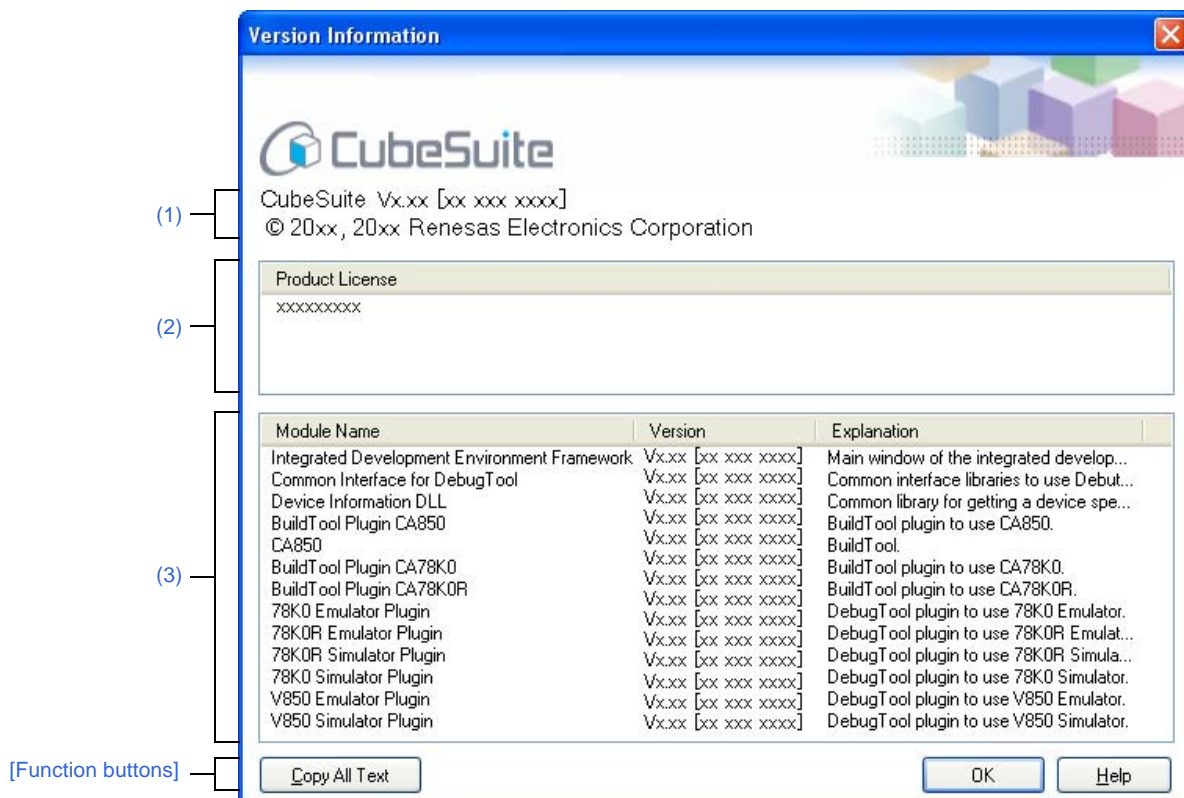
[Function buttons]

Button	Function
Reset...	Restores the arrangement of menu items or toolbar buttons belonging to the category currently selected in the [Choose a menu or toolbar to rearrange] area to its default state.
Close	Finishes the arrangement of commands and closes this dialog box.

Version Information dialog box

This dialog box is used to displays versions of CubeSuite and each plug-in product.

Figure A-59. Version Information Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Help] menu, select [About *Product name...*].

[Description of each area]

(1) Product name area

Shows the current version of CubeSuite.

(2) [Product license] area

This displays license information corresponding to the license key.

(3) Installed product area

List each module name and its version used in CubeSuite.

Module Name	Shows the module name.
Version	Shows the current version of the module.
Explanation	Shows the brief description of the module.

[Function buttons]

Button	Function
Copy All Text	Copies all the characters displayed to the clip board.
OK	Closes this dialog box.
Help	Displays the help of this dialog box.

Contact Information for Technical Support dialog box

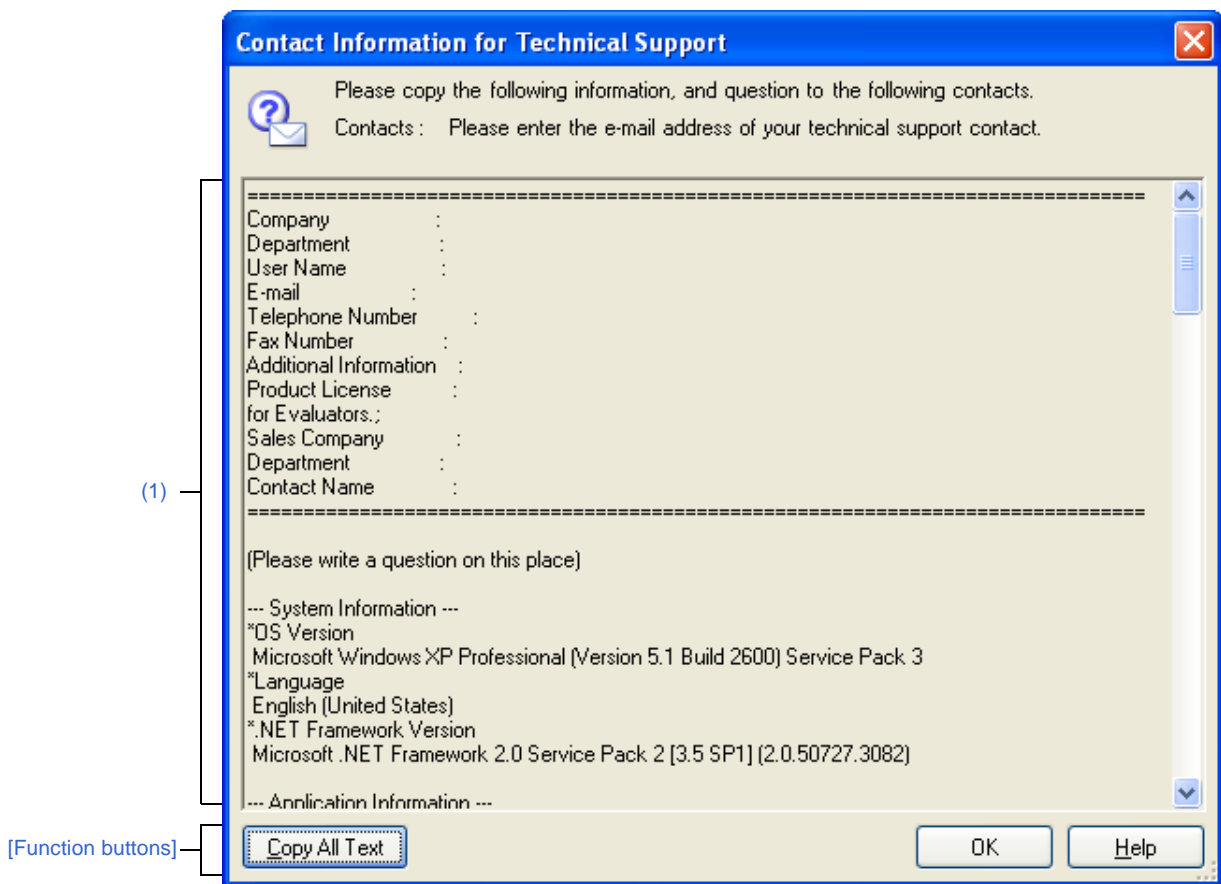
This dialog box is used to displays necessary information to contact to Tool Support Center.

If you have an inquiry regarding CubeSuite, copy the information shown in this dialog box, and include it with your question when contacting technical support.

Note that technical support by the Development Tools Support Center is limited to purchasers of the Development Tool-kit.

You cannot directly enter to the dialog box (edit is not allowed).

Figure A-60. Contact Information for Technical Support Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [[Edit] menu (Contact Information for Technical Support dialog box -dedicated items)]
- [Context menu]
- [Function buttons]

[How to open]

- From the [Help] menu, select [Contact Info for Tech-support Center...].
- Click the [Contact Information For The Tool Support Center...] button in the Message dialog box.

[Description of each area]

(1) Information text area

Show the information needed to contact to Tool Support Center.

The necessary information is as follows.

Information		Contents
User Information	Company	Showned/input in Option dialog box
	Department	
	User Name	
	E-mail	
	Telephone Number	
	Fax Number	
	Additional Information	
	Product License	
	Sales Company	
	Department	
	Contact Name	
System Information	OS Version	Windows information
	Language	
	.NET Framework Version	
Application Information	Package Version	CubeSuite information
	Version	
	Assembly Version	
	Sales Area ^{Note 1}	
	Execution Place	
Plug-in Information	Version	DLL information for each CubeSuite in use
	Assembly Version	
	DLL file name	
	URL ^{Note 1}	

Information		Contents
Present Project Information ^{Note 2}	Main Project Information (Subproject Information)	This shows the absolute path of the project file (the absolute path including the *.cspj file name for main projects, and the absolute path including the *.cssp file name for subprojects).
	Microcontroller Information	The following item to the specified microcontroller information - Device name - Device file name: List of version
	Build Tool Information	The following out of the set build tool plug-in information ^{Note 3} - Build tool name - Version of the build tool plug-in - Version of the compiler package to use ^{Note 1} .
	Debug Tool Information	The following out of the set debug tool plug-in information (if the debug tool is not set, hide) - Connected debug tool name - Version of the debug tool plug-in - Version of the debug tool control program
The Error Information Which Occurred ^{Note 4}		Detailed information on the occurred error information.

- Notes 1.** This is not shown when the information does not exist.
- 2.** This is only shown when projects are opened (when there is subproject, line them up).
- 3.** These are not shown when the project type is a debug-dedicated project.
- 4.** This is only shown when this dialog box is opened by the [Contact Information For The Tool Support Center...] button in Message dialog box.

[[Edit] menu (Contact Information for Technical Support dialog box -dedicated items)]

Copy	Copies the selected characters in the information text area to the clip board.
Select All	Selects all the characters in the information text area.

[Context menu]

Copy	Copies the selected characters in the information text area to the clip board.
Select All	Selects all the characters in the information text area.

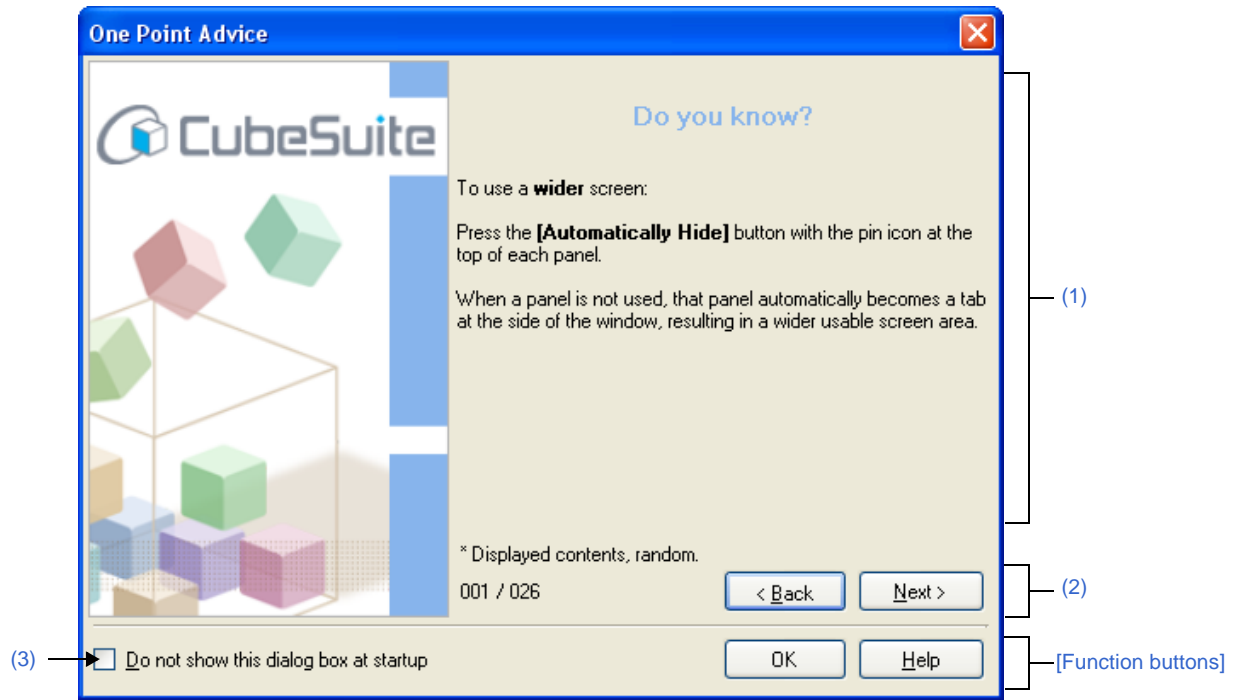
[Function buttons]

Button	Function
Copy All Text	Copies all the characters displayed in the information text area to the clip board.
OK	Closes this dialog box.
Help	Displays the help of this dialog box.

One Point Advice dialog box

This dialog box is used to display tips for using CubeSuite.

Figure A-61. One Point Advice Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Help] menu, select [One Point Advice...].
- Automatically opens at CubeSuite startup when [Do not show this dialog box at startup] is unchecked.

[Description of each area]

(1) Tips display Area

Tips for using CubeSuite are randomly shown (edit not allowed).

(2) Page feed area

The current page number in this dialog box and the buttons for page feed are shown.

Button	Function
Back	One previous page is shown. The button is disabled when the first page is shown.
Next	One next page is shown. The button is disabled when the last page is shown.

(3) [Do not show this dialog box at startup]

This configuration is saved as the one for the active user.

<input checked="" type="checkbox"/>	Does not open this dialog box automatically at startup.
<input type="checkbox"/>	Opens this dialog box automatically at startup when Main window appears (default).

Remark This property can also be set from the [Option dialog box](#), under the [\[General - Startup and Exit\] category](#).

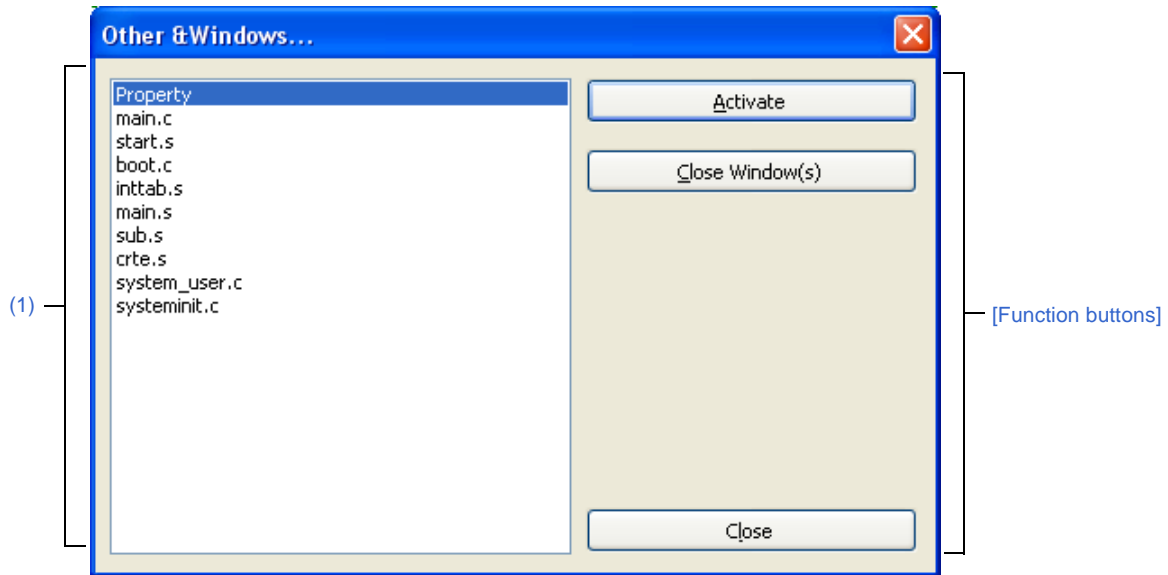
[Function buttons]

Button	Function
OK	Closes this dialog box.
Help	Displays the help of this dialog box.

Other &Windows... dialog box

This dialog box is used to select one of the divide panels shown in the [Main window](#) to activate or close.

Figure A-62. Other &Windows... Dialog Box



The following items are explained here.

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

[How to open]

- From the [Window] menu, select [Other Windows...].

[Description of each area]

(1) Select divide panel area

Selecting a divide panel switches the target.

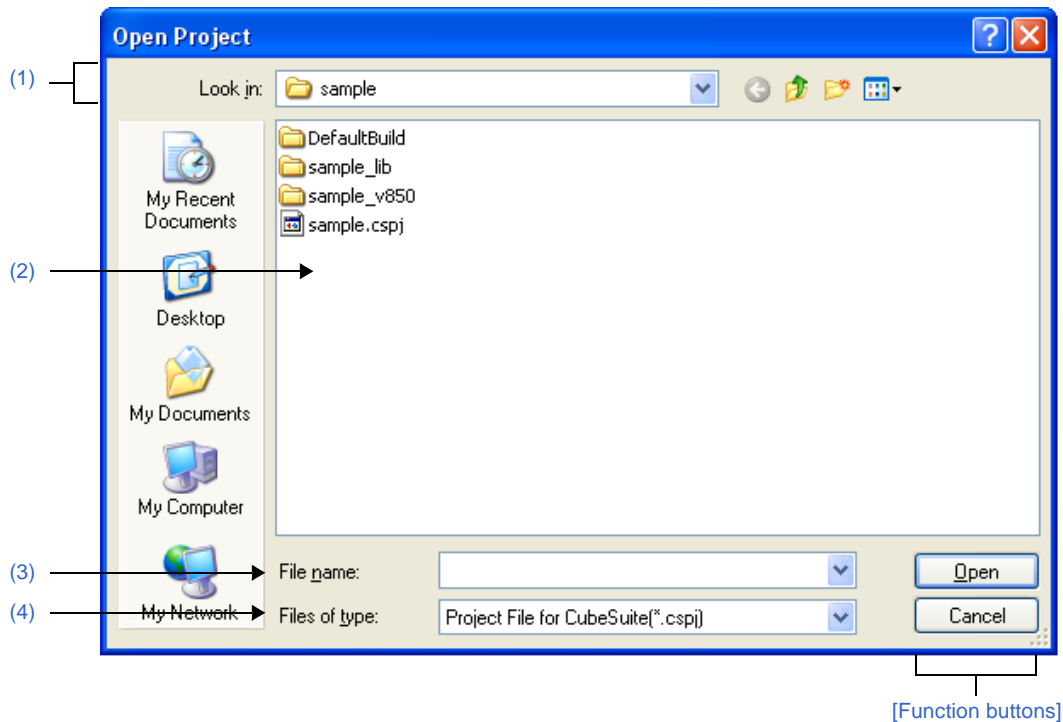
[Function buttons]

Button	Function
Activate	Activates the divide panel selected in the Select divide panel area, and close this dialog box.
Close Window(s)	Closes the divide panel selected in the Select divide panel area, and close this dialog box.
Close	Closes this dialog box.

Open Project dialog box

This dialog box is used to open an existing project or select the project file to designate the project to divert when creating a new project.

Figure A-63. Open Project Dialog Box



The following items are explained here.

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

[How to open]

- From the [Project] menu, select [Open Project...].
- In the [Create Project dialog box](#), click the [Browse...] button in [Diverting project] in the project file area.

[Description of each area]

(1) [Look in] area

Select the folder that the project file of the project that you want to open exists.

When you first copy a project, 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

File list that matches to the selections in [Look in] and [Files of type] is shown.

(3) [File name] area

Specify the project file name that you want to open.

(4) [Files of type] area

Select the file type (file type) of the project file you want to open.

(a) When the dialog box is opened from [Project] menu.

Project file for CubeSuite (*.cspj)	Project file for CubeSuite
Work space file (*.prw) for PM+.	Work space file for PM+
Project file (*.prj) for PM+.	Project file for PM+

(b) When the dialog box is opened from the [Create Project dialog box](#)

Project file for CubeSuite (*.cspj)	Project file for CubeSuite
Subproject file for CubeSuite (*.cssp)	Subproject file for CubeSuite

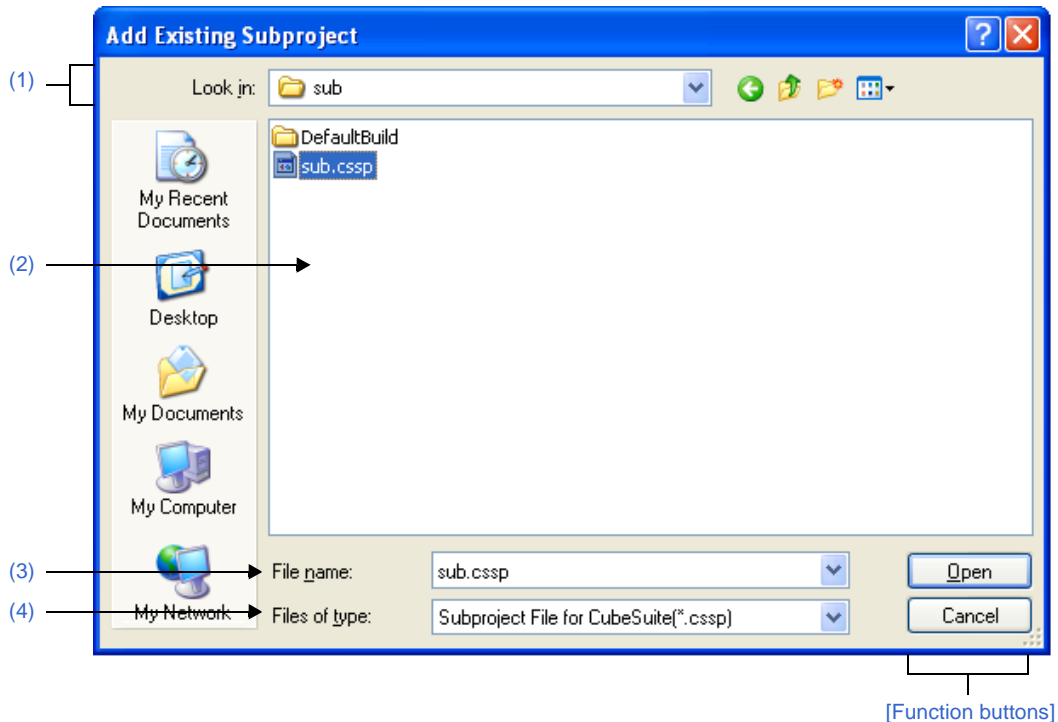
[Function buttons]

Button	Function
Open	<ul style="list-style-type: none"> - When the dialog box is opened from [Project] menu Opens the specified project file. - When the dialog box is opened from the Create Project dialog box Specifies the selected project file to [Project to be passed] in the caller dialog box.
Cancel	Closes this dialog box.

Add Existing Subproject dialog box

This dialog box is used to select subprojects for adding existing subprojects to projects.

Figure A-64. Add Existing Subproject Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [File] or [Project] menu, select [Add] >> [Add Subproject...].
- On the [Project Tree panel](#), select the Project node or Subproject node, and then select [Add] >> [Add Subproject...] from the context menu.

[Description of each area]

(1) [Look in] area

Select the folder that the subproject file of the subproject to add exists.
The project folder is selected by default.

(2) List of files area

File list that matches to the selections in [Look in] and [Files of type] is shown.

(3) [File name] area

Specify the subproject file name of the subproject to add.

(4) [Files of type] area

The following file types (file type) are displayed.

Subproject file for CubeSuite (*.cssp)	Subproject file for CubeSuite
--	-------------------------------

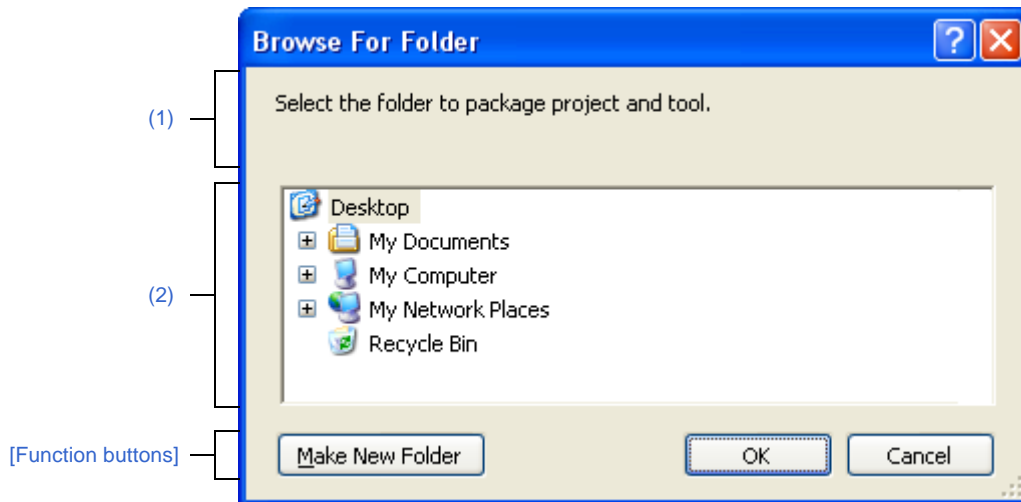
[Function buttons]

Button	Function
Open	Adds the specified subproject to a project.
Cancel	Closes this dialog box.

Browse For Folder dialog box

This dialog box is used to select the folder or file output destination (e.g. source code or report file) for the caller of this dialog box.

Figure A-65. Browse For Folder Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the [Project] menu, select [Save Project and CubeSuite as Package].
- On the [Project Tree panel](#), select the Project node and then [Save Project and CubeSuite as Package] from the context menu.
- In the [\[Whole Search\] tab](#) or [\[Whole Replace\] tab](#) of the [Search and Replace dialog box](#), click the [...] button.
- In the [Create Project dialog box](#), click the [Browse...] button in [Place] in the project file area.
- In the [Source Convert Setting dialog box \[CX\]](#), click the [Browse...] button in the [Backup of project before conversion.] area.
- In the [Add File dialog box](#), click the [Browse...] button in the [Place] area.
- In the Path Edit dialog box, click the [Browse...] button in [Path(One path per one line)] in path edit area.
- On the [Property panel](#), select the following property, and then click the [...] button.
 - The [Common Options] tab
 - The [Intermediate file output folder] of the [Output File Type and Path] category
 - The [Output folder] of the [Frequently Used Options (for Link)] category [CA78K0][CA78K0R]
 - The [Output folder for hex file] of the [Frequently Used Options (for Object Convert)] category [CA78K0][CA78K0R]
 - The [Output folder for ROMized object file] of the [Frequently Used Options (for ROMization)] category [CA850]
 - The [Output folder for hex file] of the [Frequently Used Options (for Hex Convert)] category [CA850]
 - The [Output folder for section file] of the [Frequently Used Options (for Section File Generate)] category [CA850]
 - The [Output folder] of the [Frequently Used Options (for Link)] category [CX]

- The [Error message file output folder] of the [Error Output] category [CX]
- The [Temporary folder] of [Others] category
- The [Link Options] tab [CA78K0][CA78K0R]
 - The [Output folder] of the [Output File] category
- The [Object Convert Options] tab [CA78K0][CA78K0R]
 - The [Output folder for hex file] of the [Hex File] category
- The [Create Library Options] tab [CA78K0][CA78K0R]
 - The [Output folder] of the [Output File] category
- The [Memory Bank Relocation Options] tab [CA78K0]
 - The [Output folder for function information file], the [Output folder for replacement information file], the [Output folder for object information file] and the [Output folder for reference information file] of the [Output File] category
- The [Variables/Functions Relocation Options] tab [CA78K0R]
 - The [Output folder for variables/functions information file] of the [Output File] category
- The [Compile Options] tab [CA850]
 - The [Output folder for assembly file], the [Output folder for assemble list] and the [Output folder for frequency information file] of the [Output File] category
- The [Assemble Options] tab [CA850]
 - The [Output folder for assemble list file] of the [Assemble List] category
- The [Link Options] tab [CA850]
 - The [Output folder] of the [Output File] category
 - The [Output folder for link map file] of the [Link Map] category
- The [ROMization Process Options] tab [CA850]
 - The [Output folder for ROMized object file] of the [Output File] category
 - The [Output folder for ROMization section file] of the [Section List] category
 - The [Output folder for memory map file] of the [Memory Map] category
- The [Hex Convert Options] tab [CA850]
 - The [Output folder for hex file] of the [Output File] category
- The [Archive Options] tab [CA850]
 - The [Output folder] of the [Output File] category
- The [Section File Generate Options] tab [CA850]
 - The [Output folder for section file] of the [Output File] category
- The [Individual Compile Options] tab [CA850]
 - The [Output folder for assembly file], the [Output folder for assemble list] and the [Output folder for frequency information file] of the [Output File] category
- The [Individual Assemble Options] tab [CA850]
 - The [Output folder for assemble list file] of the [Assemble List] category
- The [Dump Options] tab [CA850]
 - The [Output folder] of the [Output File] category
- [Compile Options] tab [CX]
 - The [Output folder for assembler source file] of the [Output File] category
 - The [Output folder for assemble list file] of the [Assemble List] category
- [Link Options] tab [CX]
 - The [Output folder] of the [Output File] category
 - The [Output folder for link map file] of the [Link Map] category
 - The [Output folder for symbol information file] of the [Symbol Information] category
- [Hex Output Options] tab [CX]
 - The [Output folder for hex file] of the [Output File] category
- [Create Library Options] tab [CX]
 - The [Output folder] of the [Output File] category

- [Individual Compile Options] tab [CX]
 - The [Output folder for assembler source file] of the [Output File] category
 - The [Output folder for assemble list file] of the [Assemble List] category
- The [Generation] tab, the [Output folder]
- Click the [Copy] button in the [History page](#) in the [CubeSuite Update Manager window](#).

[Description of each area]

(1) Message area

Shows messages related to folders selected in this dialog box.

(2) Folder location area

Select a folder to set in the caller of this dialog box, or a folder to which to output files (e.g. source code and report files).

The folder selected by default differs depending on the caller.

(a) [Project] menu and Project Tree panel

"C:\Documents and Settings\user name\My Documents" is selected for the first time. The previously selected folder is selected after the second time.

(b) Search and Replace dialog box, Source Convert Setting dialog box, and Add File dialog box

The folder set in the caller is selected.

When the field is blank or the path which does not exist is entered, the project folder is selected instead.

(c) Create Project dialog box

The folder set in the caller is selected.

When the field is blank or the path which does not exist is entered, "C:\Documents and Settings\user name\My Documents" is selected instead.

(d) Path Edit dialog box and Property panel

The project folder is selected.

(e) CubeSuite Update Manager window

"C:\Documents and Settings\user name\Desktop" is selected.

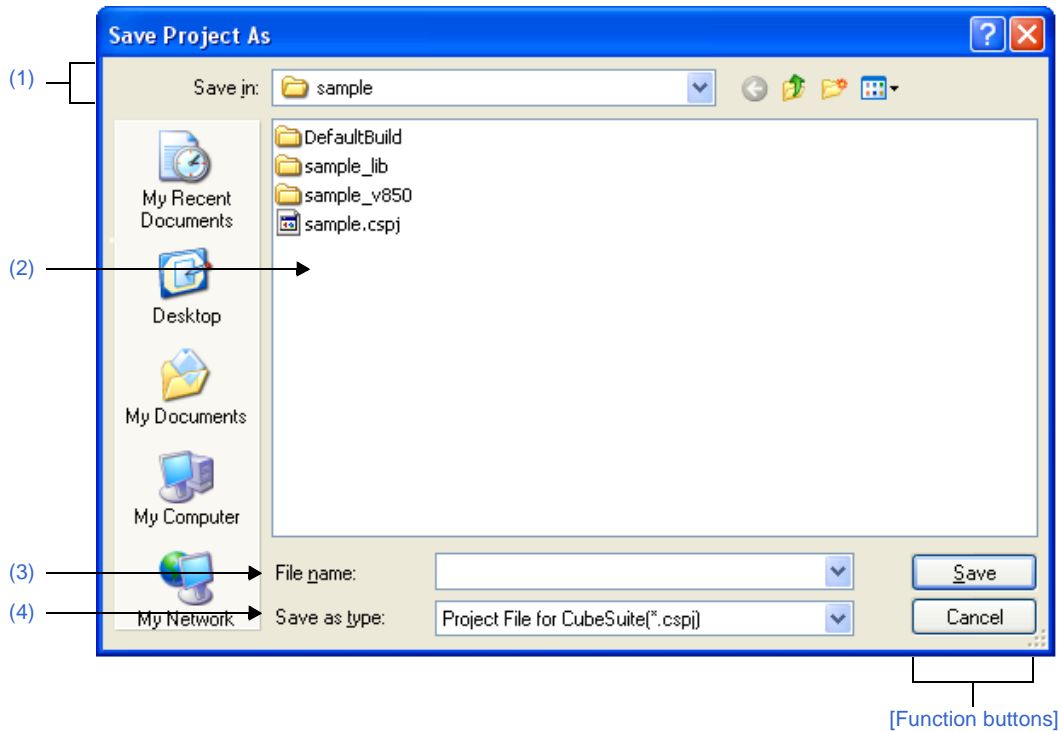
[Function buttons]

Button	Function
Make New Folder	Creates a new folder in the root of the selected folder. The default folder name is "New folder".
OK	The designated folder path is set to the path that this dialog box is called from. The folder selected in the Folder location area is set as the file output destination.
Cancel	Closes this dialog box.

Save Project As dialog box

This dialog box is used to save project files as different names.

Figure A-66. Save Project As Dialog Box



The following items are explained here.

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

[How to open]

- From the [File] menu or [Project] menu, select [Save Project As...].

[Description of each area]

(1) [Save in] area

Select a folder to save the project file.
The project folder is selected by default.

(2) List of files area

File list that matches to the selections in [Save in] and [Files of type] is shown.

(3) [File name] area

Specify the project file name to save.

(4) [Save as type] area

The following file types (file type) are displayed.

Project file for CubeSuite (*.cspj)	Project file for CubeSuite.
-------------------------------------	-----------------------------

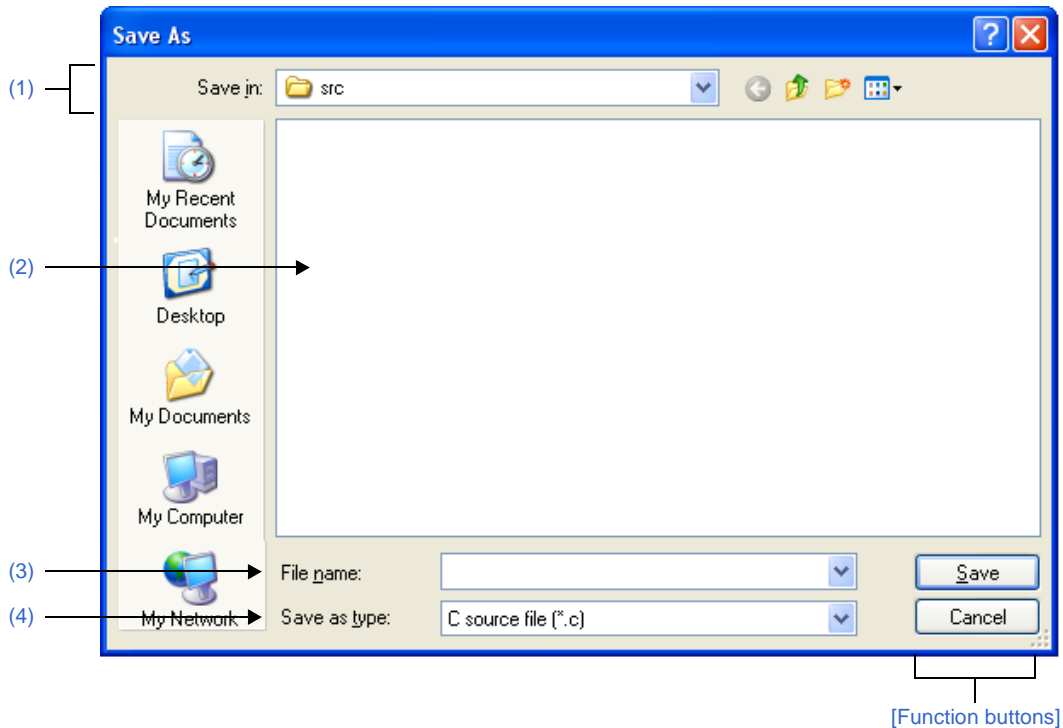
[Function buttons]

Button	Function
Save	Saves the project file as the designated file name.
Cancel	Closes this dialog box.

Save As dialog box

This dialog box is used to save the editing file or contents of each panel to a file with a name.

Figure A-67. Save As Dialog Box



The following items are explained here.

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

[How to open]

- Focus Editor panel and from the [File] menu, select [Save *file name* As...].
- Focus CPU Register panel and from the [File] menu, select [Save CPU Register Data As...].
- Focus Watch panel and from the [File] menu, select [Save Watch Data As...].
- Focus SFR panel and from the [File] menu, select [Save SFR Data As...]. [78K0][78K0R]
- Focus IOR panel and from the [File] menu, select [Save IOR Data As...]. [V850]
- Focus Call Stack panel and from the [File] menu, select [Save Call Stack Data As...].
- Focus Local Variables panel and from the [File] menu, select [Save Local Variables Data As...].
- Focus Output panel and from the [File] menu, select [Save *tab name* As...].
- Focus Device Pin List panel and from the [File] menu, select [Save Device Pin List As...].
- Focus Device Top View panel and from the [File] menu, select [Save Device Top View As...].
- Focus Function List panel and from the [File] menu, select [Save Function List Data As...].
- Focus Variable List panel and from the [File] menu, select [Save Variable List Data As...].
- Focus Analysis Chart panel and from the [File] menu, select [Save Analysis Chart Data As...].

[Description of each area]**(1) [Save in] area**

Select the folder to save the panel contents in the file.

(2) List of files area

File list that matches the selections in [Save in] and [Files of type] area is shown.

(3) [File name] area

Specify the file name to save.

(4) [Save as type] area**(a) In the Editor panel**

The following file types (file type) are displayed depend on the file type of the currently editing file.

Text file (*.txt)	Text format
C source file (*.c)	C language source file
Header file (*.h;*.inc)	Header file
Assemble file (*.asm) [CA78K0][CA78K0R][CX]	Assembler source file
Assemble file (*.s) [CA850][CX]	Assembler source file
Assemble file (*.asm; *.s) ^{Note 1}	Assembler source file
Link directive file (*.dr; *.dir) [CA78K0][CA78K0R]	Link directive file
Link directive file (*.dir; *.dr) [CA850][CX]	Link directive file
Function information file (*.fin) ^{Note 2} [CA78K0]	Function information file
Variable and function information file (*.vfi) [CA78K0R]	Variable and function information file
Section file (*.sf) [CA850]	Section file
Symbol information file (*.sfg) [CX]	Symbol information file
Map file (*.map)	Map file
Symbol table file (*.sym) [CA78K0][CA78K0R]	Symbol table file
Hex file (*.hex; *.hxb; *.hxf) [CA78K0][CA78K0R]	Hex file
Hex file (.hex) [CA850][CX]	Hex file

Notes 1. This file type is only shown for a debug-dedicated project.

2. This file type is only shown for microcontrollers with a memory bank.

(b) In the CPU Register panel/Watch panel/SFRpanel [78K0][78K0R]/IOR panel [V850]/Call Stack panel/Local Variables panel

The following file types (file type) are displayed.

This saves the contents of the panel in the file format selected from the drop-down list.

Text files (*.txt)	Text format (default)
CSV (Comma-Separated Variables) (*.csv)	CSV format ^{Note}

Note 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) In the Output panel

The following file types (file type) are displayed.

The file can only be saved in text format.

Text file (*.txt)	Text format
-------------------	-------------

(d) In the Device Pin List panel

The following file types (file type) are displayed.

The file can only be saved in Excel format.

Microsoft Office Excel Book (*.xls)	Microsoft Office Excel Book format
-------------------------------------	------------------------------------

(e) In the Device Top View panel

The following file types (file type) are displayed.

The file can only be saved in bit map format.

Bitmap (*.bmp)	Bitmap format
----------------	---------------

(f) In the Function List panel

The following file types (file type) are displayed.

Text files (*.txt)	Text format
CSV (Comma-Separated Variables) (*.csv)	CSV format
Microsoft Office Excel Workbook (*.xls)	Microsoft Excel 97 - Excel 2003 and 5.0/95 book (*.xls) format
CubeSuite Function List File (*.csfl)	File format for importing function information

(g) In the Variable List panel

The following file types (file type) are displayed.

Text files (*.txt)	Text format
CSV (Comma-Separated Variables) (*.csv)	CSV format
Microsoft Office Excel Workbook (*.xls)	Microsoft Excel 97 - Excel 2003 and 5.0/95 book (*.xls) format
CubeSuite Variable List File (*.csvl)	File format for importing variable information

(h) In the Analysis Chart panel

The following file types (file type) are displayed.

Text files (*.txt)	Text format
CSV (Comma-Separated Variables) (*.csv)	CSV format
Microsoft Office Excel Workbook (*.xls)	Microsoft Excel 97 - Excel 2003 and 5.0/95 book (*.xls) format
Bitmaps (*.bmp)	Bitmap format
JPEG Files (*.jpg)	JPEG format
PNG Files (*.png)	PNG format

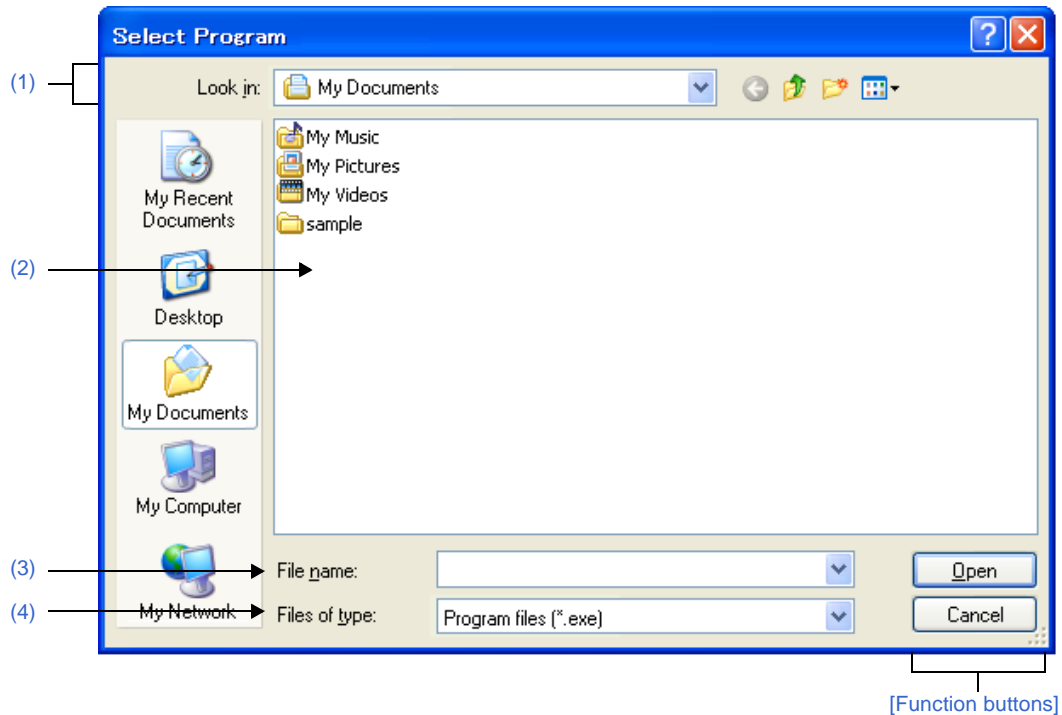
[Function buttons]

Button	Function
Save	Saves the file as the designated file name.
Cancel	Closes this dialog box.

Select Program dialog box

This dialog box is used to select the executable file of an external tool.

Figure A-68. Select Program Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- In the [General - External Tools] category of the Option dialog box, click the [...] button in the new registration area.

[Description of each area]

(1) [Look in]

Select the location (folder) of the executable for the external tool to register from the drop-down list.

(2) List of files

This area displays a list of files matching the conditions selected in [Look in] and [Files of type].

(3) [File name]

Specify the name of the executable file for the external tool to register.

(4) [Files of type]

Select the type of the executable file for the external tool to register from the following drop-down list.

Program files (*.exe)	Executable format (default)
All files (*.*)	All formats

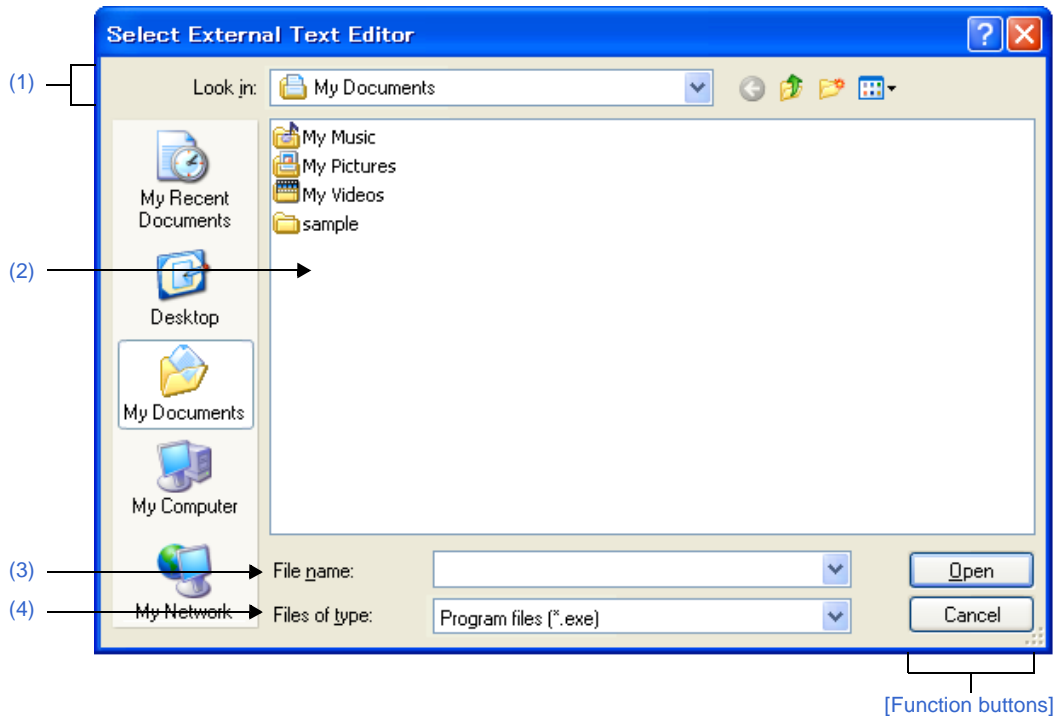
[Function buttons]

Button	Function
Open	Specifies the selected file in the Option dialog box .
Cancel	Ignores the setting and closes this dialog box.

Select External Text Editor dialog box

This dialog box is used to select the executable file of an external text editor.

Figure A-69. Select External Text Editor Dialog Box



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- In the [General - Text Editor] category of the Option dialog box, click the [Browse...] button in the [External text editor] area.

[Description of each area]

(1) [Look in]

Select the location (folder) of the executable file for the external text editor to register from the drop-down list.

(2) List of files

This area displays a list of files matching the conditions selected in [Look in] and [Files of type].

(3) [File name]

Specify the name of the executable file for the external text editor.

(4) [Files of type]

Select the type of the executable file for the external text editor to register from the following drop-down list.

Program files (*.exe)	Executable format (default)
All files (*.*)	All formats

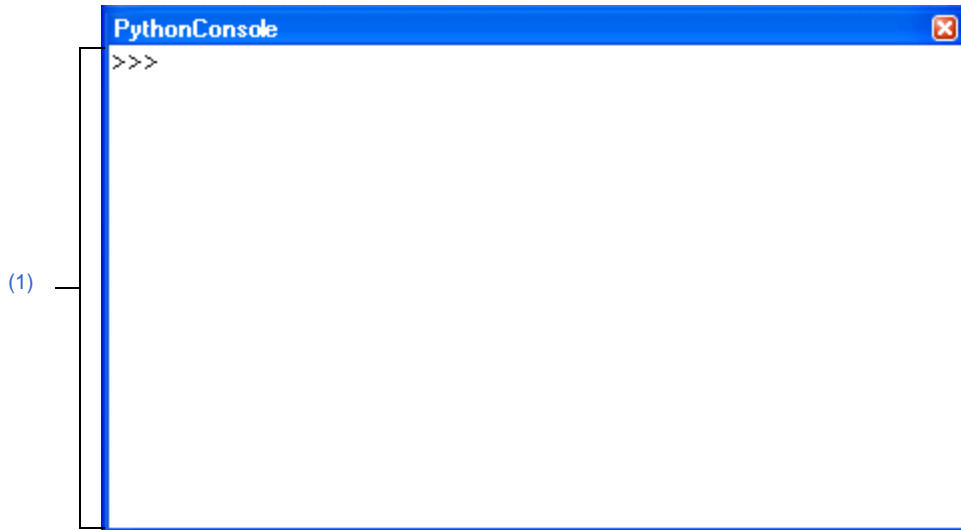
[Function buttons]

Button	Function
Open	Specifies the selected file in the Option dialog box .
Cancel	Ignores the setting and closes this dialog box.

Python Console panel

This panel is used to use IronPython to control CubeSuite and the debug tool via the command input method.

Figure A-70. Python Console Panel



The following items are explained here.

- [How to open]
- [Description of each area]
- [[File] menu (Python Console panel-dedicated items)]
- [Context menu]

[How to open]

- From the [View] menu, select [Python Console].

[Description of each area]

(1) I/O area

Enter and run IronPython functions and control statements, and CubeSuite Python functions.
The results of function execution and errors are also displayed.

[[File] menu (Python Console panel-dedicated items)]

The following items are exclusive for [File] menu in the Python Console panel (other items are common to all the panels).

Save Python Console	Saves the content displayed in the current panel in the last text file (*.txt) to be saved. Note that if this item is selected first after the program starts, then the behavior is the same as selecting [Save Python Console As...].
Save Python Console As ...	Opens the Save As dialog box to save the contents currently displayed on this panel in the designated text file (*.txt).

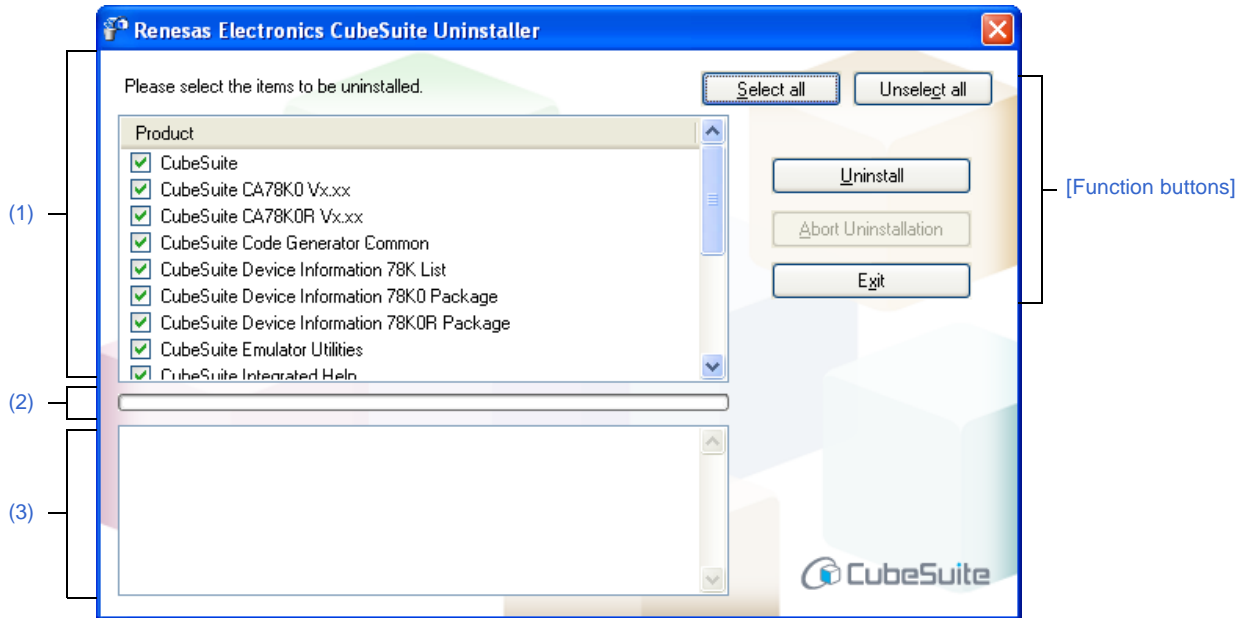
[Context menu]

Cut	Cuts the selected characters and copies them to the clip board.
Copy	Copies the selected characters to the clip board.
Paste	Inserts the contents of the clipboard into the caret position.
Select All	Selects all characters displayed on this panel.
Abort	Forces the currently running command to stop.

CubeSuite Uninstaller window

This window is used to specify one or more installed CubeSuite products to uninstall at once.

Figure A-71. CubeSuite Uninstaller Window



The following items are explained here.

- [How to open]
- [Description of each area]
- [Function buttons]

[How to open]

- From the Windows [Start] menu, select [Programs] >> [NEC Electronics CubeSuite] >> [CubeSuite Uninstaller].

[Description of each area]

(1) Select Tools area

The CubeSuite products that are installed appear here. Select the check boxes of the tools to uninstall.

(2) Progress Graph area

This area displays a progress bar of the uninstallation process.

(3) Progress Details area

This area displays information about ongoing and completed uninstalls for each tool to be uninstalled.

[Function buttons]

Button	Function
Select all	Selects all check boxes.
Unselect all	Clears all check boxes.
Uninstall	Uninstalls the selected tools.
Abort Uninstallation	Cancels the uninstallation.
Exit	Closes this window.

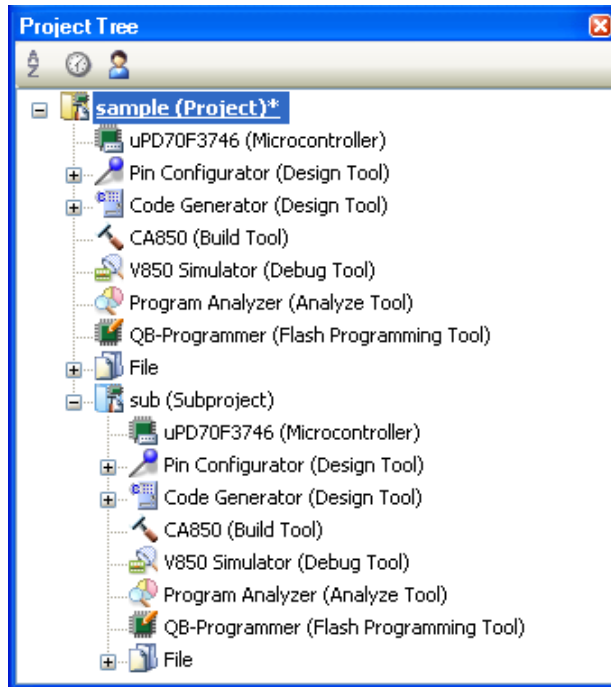
APPENDIX B GLOSSARY

This section defines terms used in the CubeSuite manual.

(1) Node

This refers to an element representing a branch or leaf on a tree view, connected with lines to show a hierarchy in the [Project Tree panel](#) and other views.

Figure B-72. Project Tree Panel



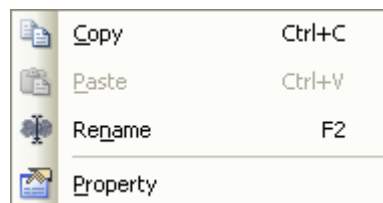
(2) List control

This displays a widget such as in the panel settings areas. Click the to display a list of available selections.

(3) Context menu

This refers to the menu that appears when you right click in a window, over an icon, or other object on the screen. The menu displays a list of operations currently available for the object you clicked.

Figure B-73. Sample Context Menu

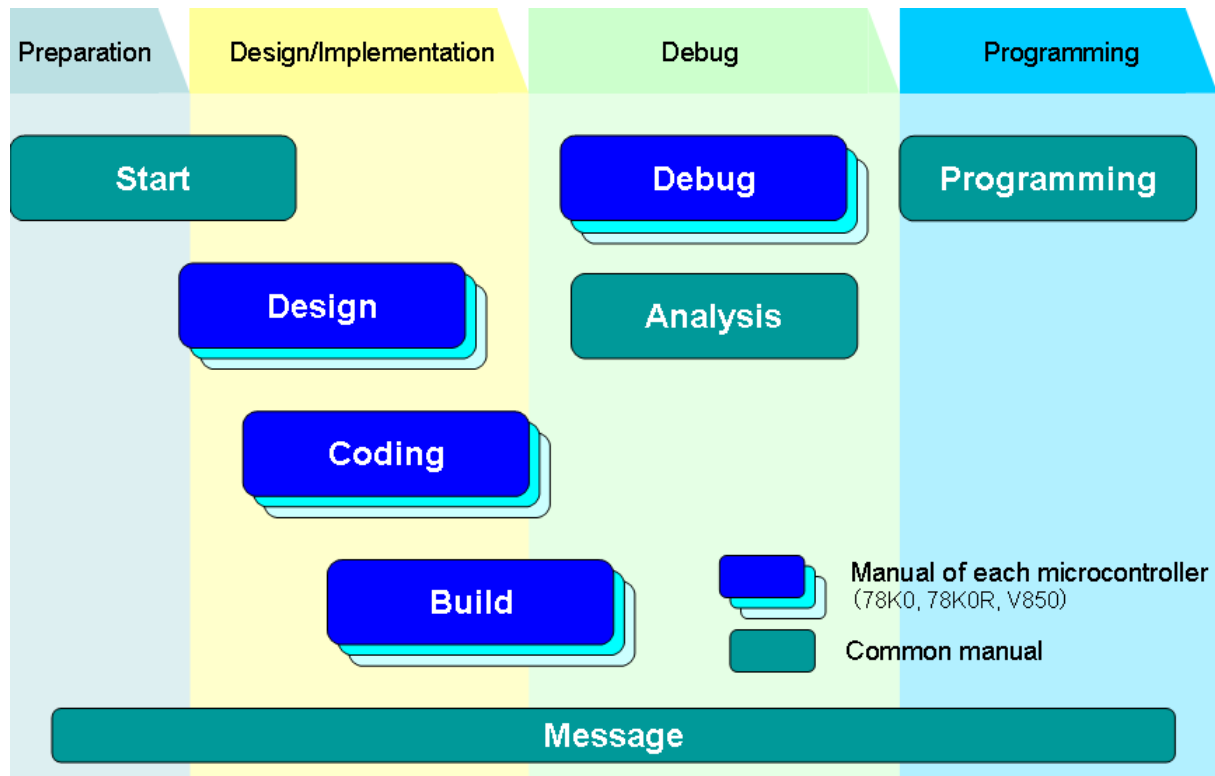


APPENDIX C HOW THE MANUALS ARE ORGANIZED

This section describes the manuals included with the CubeSuite integrated development environment for developing applications and systems for the 78K0 microcontroller, 78K0R microcontroller and V850 microcontroller.

The manuals are organized as follows, in accordance with the software development phases.

Figure C-1. List of Manuals



- Notes**
1. The manuals provided will differ depending on which products are installed.
 2. The "CubeSuite Start" includes instructions on builds in the design/implementation processes when using an external build tool.

(1) Start

This manual describes an outline of CubeSuite. It also describes installation, updates, license settings, and other information. It describes the operations from launching CubeSuite to creating a project.

(2) Design

This manual describes the design tool (pin assignment and code generation).

(3) Coding

This manual describes the roles and functions of the commands of the C compiler, assembler, and other tools included with CubeSuite. It provides information necessary for development using the C compiler and assembler, as well as expert knowledge on effective coding techniques.

(4) Build

This manual describes the build tools. The build tool component is part of CubeSuite. It enables various types of information to be configured via a GUI tool. This enables you to generate load module files, hex-format object module files, or library files from your source files, according to your objectives.

(5) Debug

This manual describes how to control the connection and execution settings of your debug tool. It also provides information for when you use the connected debug tool to debug your programs.

(6) Analysis

This manual describes the analyze tools. The analysis tool analyzes the source program and information while the program is executing, and provides information about the functions and variables.

(7) Programming

This manual describes how to program to the internal Flash memory.

(8) Message

This manual describes output messages.

APPENDIX D INPUT CONVENTIONS

This section describes input conventions.

D.1 Input Conventions

Below are input conventions for configuring information in the panels and dialog boxes provided by CubeSuite.

(1) Character sets

The following character sets are allowed for input.

Table D-1. List of Character Sets

Character Sets	Outline
ASCII	Single-byte numbers, letters, and punctuation
Shift-JIS	Double-byte letters, numbers, punctuation, Hiragana, Katakana, and Kanji, and single-byte Katakana
EUC-JP	Double-byte letters, numbers, punctuation, Hiragana, Katakana, and Kanji, and single-byte Katakana
UTF-8	Double-byte letters, numbers, punctuation, Hiragana, Katakana, and Kanji (including Chinese), and single-byte Katakana


(2) Numbers

Numbers may be input in the following base formats.

Table D-2. List of Number Bases

Number Bases	Outline
Decimal number	Starts with 1 to 9, followed by a sequence of the digits 0 to 9, and 0
Hexadecimal number	Starts with 0x, followed by a sequence of the digits 0 to 9 and the letters a to f (As for capital letter/small letter of the alphabet, it is pretermisson.)

D.2 Displaying Icons at Locations of Input Errors

Some of the panels and dialog boxes provided by CubeSuite display warnings when there is an input error or a required item is missing: they display  icons at the locations of input errors where illegal characters were entered, indicating that the necessary information was not entered.


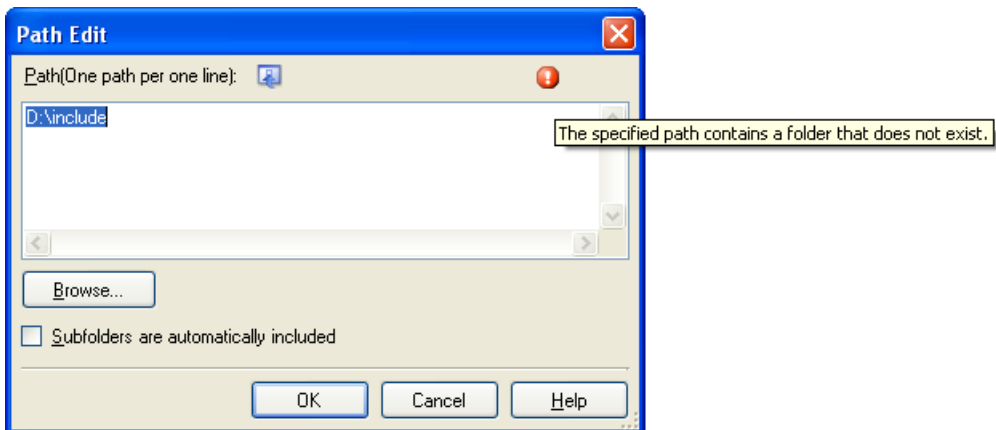
Remark Move the mouse cursor over the  icon to view information about the required input string (tips to fix the error) in a popup message.

Figure D-1. Sample Icons Shown at Locations of Input Errors



APPENDIX E USING AN EXTERNAL BUILD TOOL

This section describes how to create a project when debugging files generated with a build tool other than the one provided by CubeSuite (CA850/CX/CA78K0/CA78K0R).

E.1 Overview

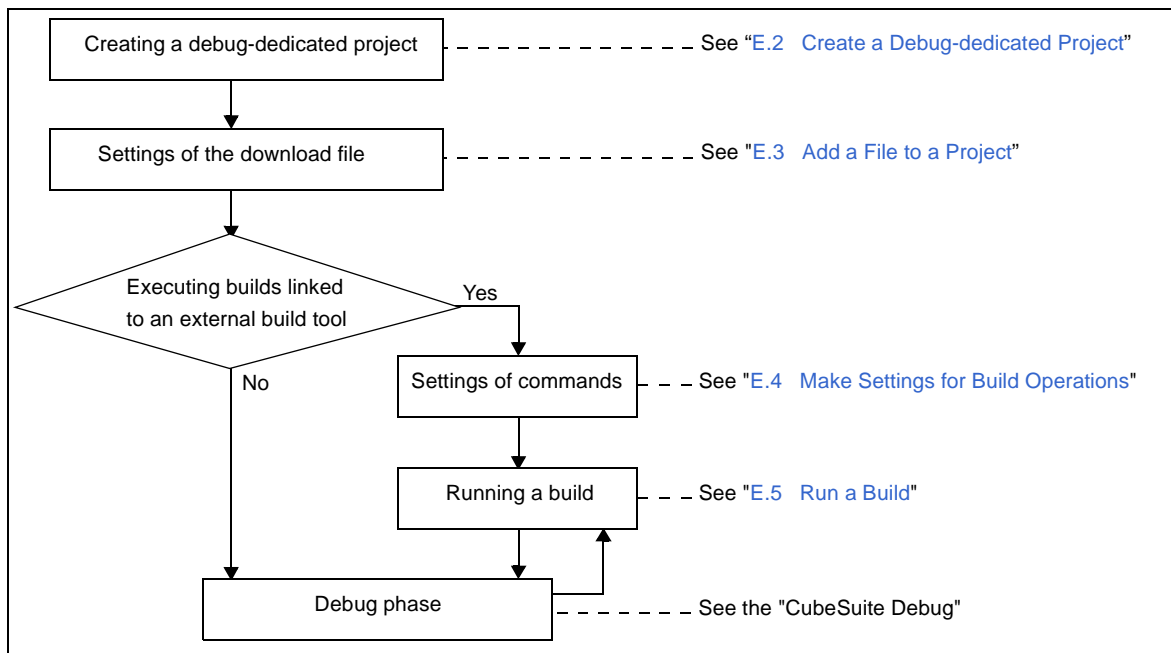
CubeSuite automatically determines the load module file or hex file output by the build tool provided by CubeSuite (CA850/CX/CA78K0/CA78K0R) as the download file for debugging. For this reason, you must create a dedicated project (hereafter referred to as a "debug-dedicated project") in order to debug a load module file or hex file created by an external build tool (e.g. a compiler or assembler other than the build tool provided by CubeSuite) as the download file.

Creating a debug-dedicated project enables you to perform debugging without going through the build tool provided by CubeSuite.

A debug-dedicated project also enables you to configure commands to execute build processes (e.g. "make") suited to your execution environment, and you can perform builds linked to an external build tool by executing these commands from CubeSuite.

The operational sequence for using a debug-dedicated project is described below.

Figure E-1. Operational Sequence for Using a Debug-dedicated Project



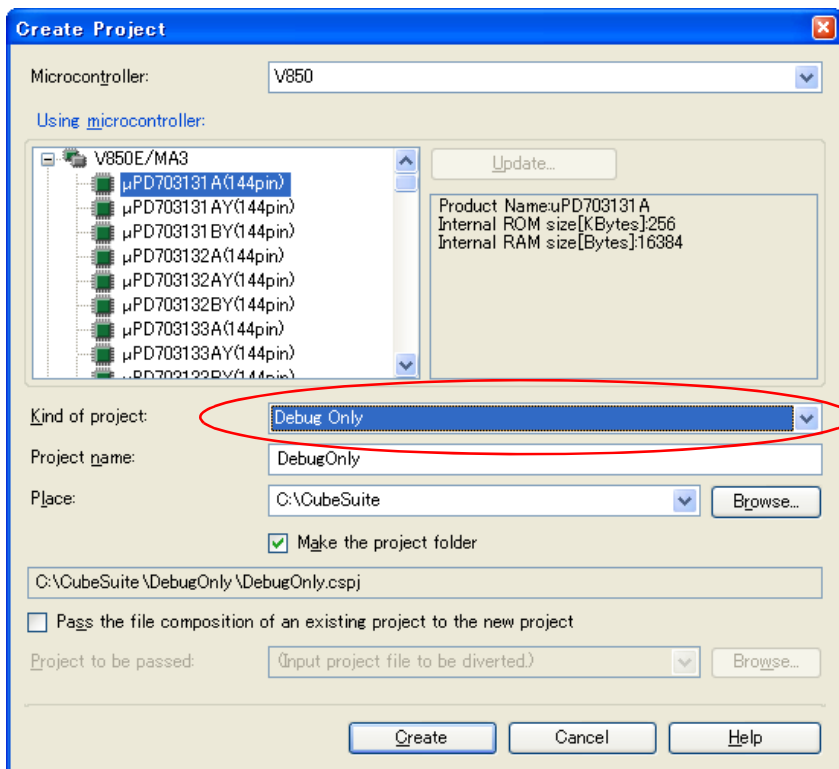
Caution The design tool (code generator) and the analyze tool are not available for a debug-dedicated project.

E.2 Create a Debug-dedicated Project

The creation of a debug-dedicated project is performed with the [Create Project dialog box](#) that is opened by selecting [Create New Project...] from the [Project] menu or [Add] >> [Add New Subproject...] from the context menu after selecting the Project node on the project tree, as well as the method of creating a project described in "2.6.2 [Create a new project](#)" or "2.6.3 [Add a new subproject](#)".

To create a debug-dedicated project, however, specify [Debug Only] with the [Kind of project] item on the [Create Project dialog box](#).

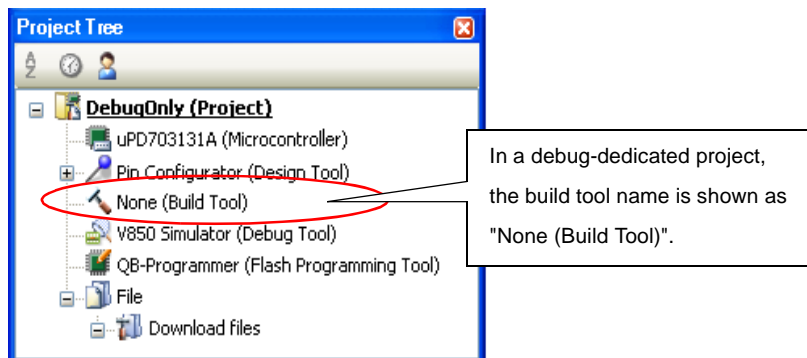
Figure E-2. Create Project Dialog Box (Creating a Debug-dedicated Project)



When you click the [Create] button after configuring all settings^{Note}, the project file of the debug-dedicated project is created in the location specified in the [Place] item and the structure of the created debug-dedicated project is displayed as a tree in the [Project Tree panel](#).

Note See "2.6.2 [Create a new project](#)" or "2.6.3 [Add a new subproject](#)", for details on how to configure each item.

Figure E-3. Project Tree Panel (After Creating a Debug-dedicated Project)



E.3 Add a File to a Project

Add a download file to debug to the created debug-dedicated project.

Furthermore, to perform source level debugging, add C source files to be used for creation of the download file to the project.

This section describes how to add these files to the debug-dedicated project.

E.3.1 Add a download file

Add a load module file to the debug-dedicated project as the download file to debug.

A load module file can be added by the following methods.

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

Caution To perform source level debugging (step execution in source level units, etc.), a load module file with the symbol information must be added to the debug-dedicated project.

Remarks 1. The specified load module file is reflected automatically on the [Download File Settings] tab in the Property panel of the debugging tool.

Also use this [Download File Settings] tab to configure the load module file type and download options, add a hex file or binary file to download additionally as well as the load module file added by this method below.

See the "CubeSuite Debug", for details on the downloading.

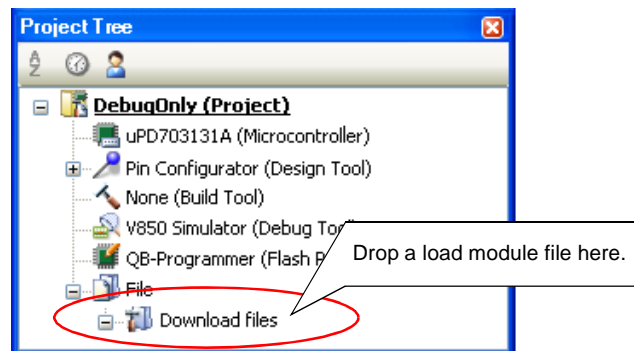
2. [V850]

Multiple load module files can be added to the debug-dedicated project.

(1) Adding an existing file

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

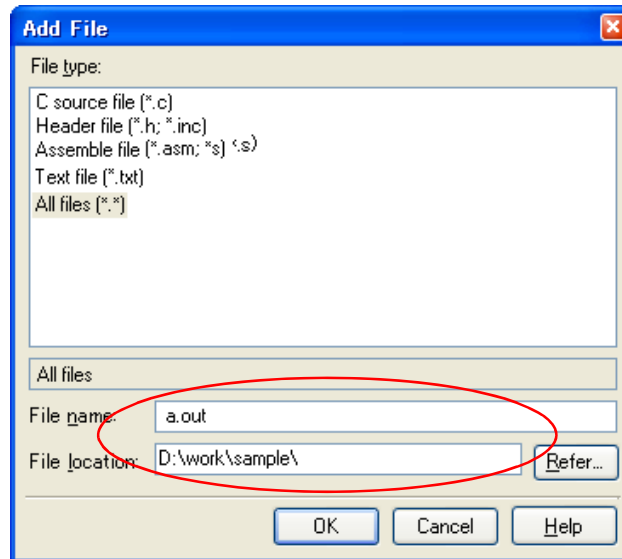
Figure E-4. Project Tree Panel (Adding a Load Module File)



(2) Creating and adding an empty file

Select the Download files node on the project tree, and then select [Add] >> [Add New File...] from the context menu. The [Add File dialog box](#) will open.

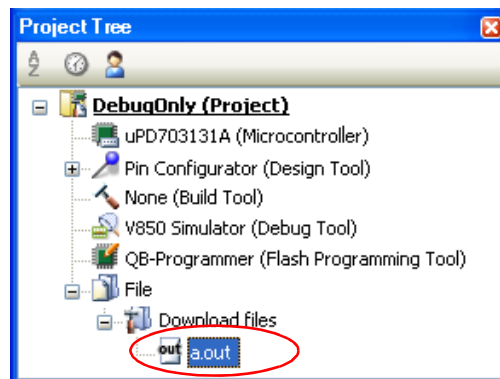
Figure E-5. Add File Dialog Box (Adding a Load Module File)



In the dialog box, specify the file name to be created newly and the location in which it is created, and then click the [OK] button.

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

Figure E-6. Project Tree Panel (After Adding Load Module File "a.out")



E.3.2 Add C source files and other files

Add C source files to be used for creation of the download file to the debug-dedicated project.

By adding the C source file to the project, you can perform source level debugging (step execution in source level units, etc.).

You can also add files other than C source files as necessary. By double-clicking the added file name on the project tree, you can open the [Editor panel](#) and edit the contents of the file directly.

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

- C source file (*.c)
- Assembler source file (*.asm, *.s)
- Header file (*.h, *.inc)
- Symbol information file (*.sfg)
- Link directive file (*.dir, *.dr)
- Link map file (*.map)
- Hex file (*.hex)
- Text file (*.txt)

Caution To perform source level debugging, a load module file with the symbol information must be added as a download file to debug (see "E.3.1 Add a download file").

- Remarks 1.** By dragging a file and dropping it onto the [Editor panel](#), you can open files other than those listed above in the [Editor panel](#).
- 2.** When the environment is set to use an external editor on the [Option dialog box](#), the file is opened with the external editor that has been set.
Other files are opened with the applications associated by the host OS.

C source files and other files can be added by the following methods.

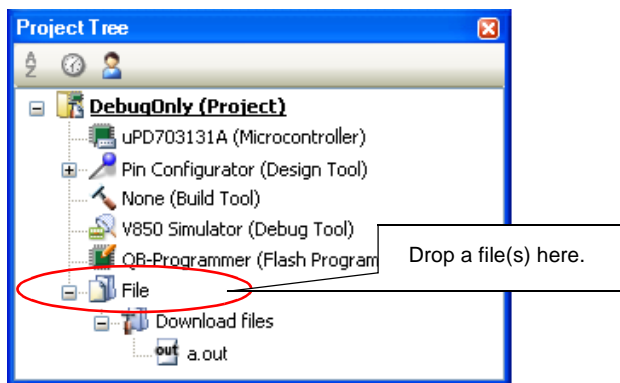
- [Adding an existing file](#)
- [Creating and adding an empty file](#)

(1) Adding an existing file

(a) Add an individual files

Drag a file(s) from Explorer or the like, and drop it onto the File node on the project tree.

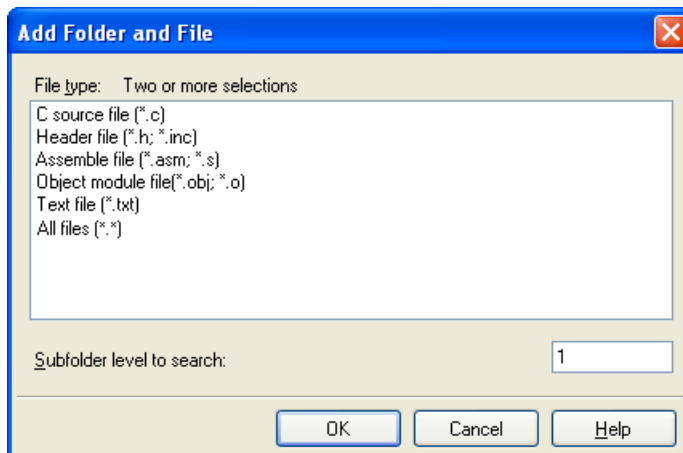
Figure E-7. Project Tree Panel (Adding a File)



(b) Add a folder

Drag a folder(s) from Explorer or the like, and drop it onto the File node on the project tree. The [Add Folder and File dialog box](#) will open.

Figure E-8. Add Folder and File Dialog Box

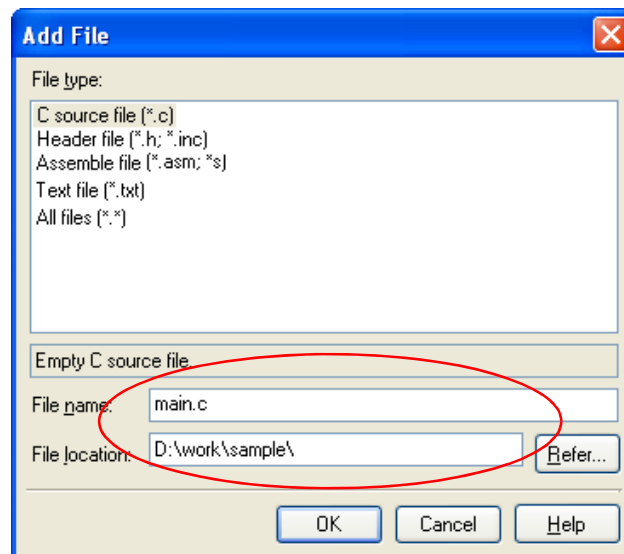


In the dialog box, select the types of the files to be added and specify the number of levels of the subfolder to be added to the debug-dedicated project. At this time, you can select multiple file types by left clicking while holding down the [Ctrl] or [Shift] key. If nothing is selected, it is assumed that all types are selected. And then click the [OK] button.

(2) Creating and adding an empty file

Select the File node on the project tree, and then select [Add] >> [Add New File...] from the context menu. The [Add File dialog box](#) will open.

Figure E-9. Add File Dialog Box (Adding a File)



In the dialog box, specify the file name to be created newly and the location in which it is created, and then click the [OK] button.

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

Note that the location of the file added below the File node depends on the current settings of [\[Toolbar\]](#) in the [Project Tree panel](#).

Figure E-10. Project Tree Panel (After Adding File "main.c")

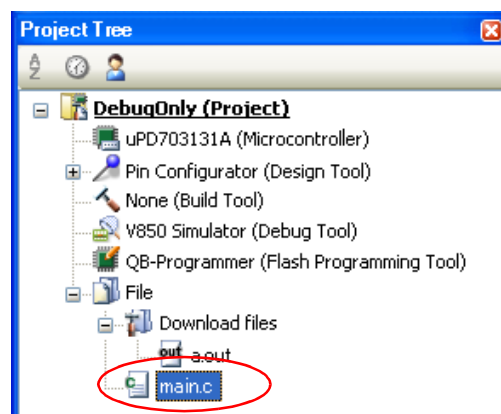
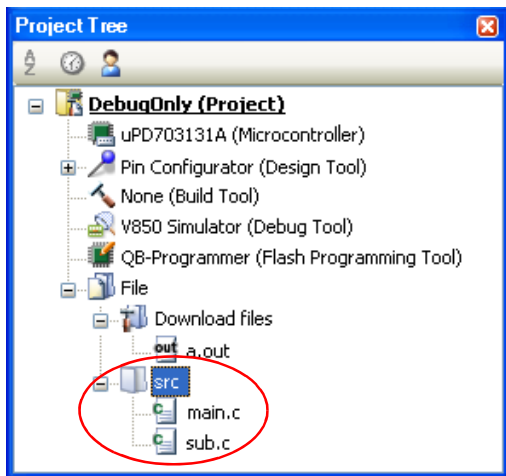


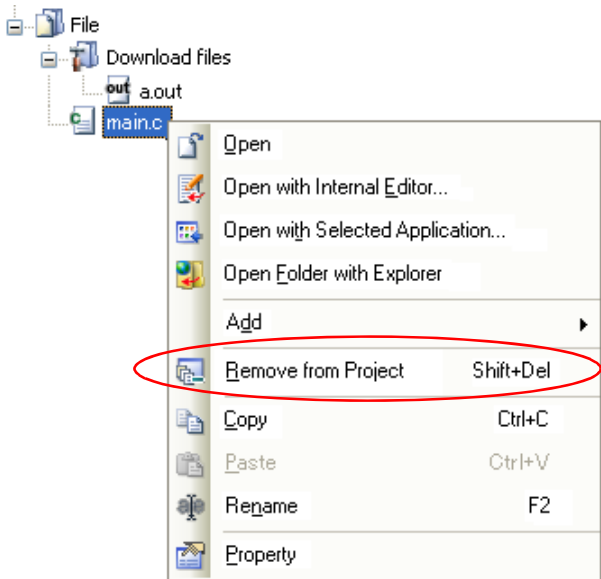
Figure E-11. Project Tree Panel (After Adding Folder "src")



E.3.3 Remove an added file from a project

To remove a file(s) that has been added to the debug-dedicated project by using the method above, select the file(s) on the project tree and then select [Remove from Project] from the context menu.

Figure E-12. [Remove from Project] Item



E.4 Make Settings for Build Operations

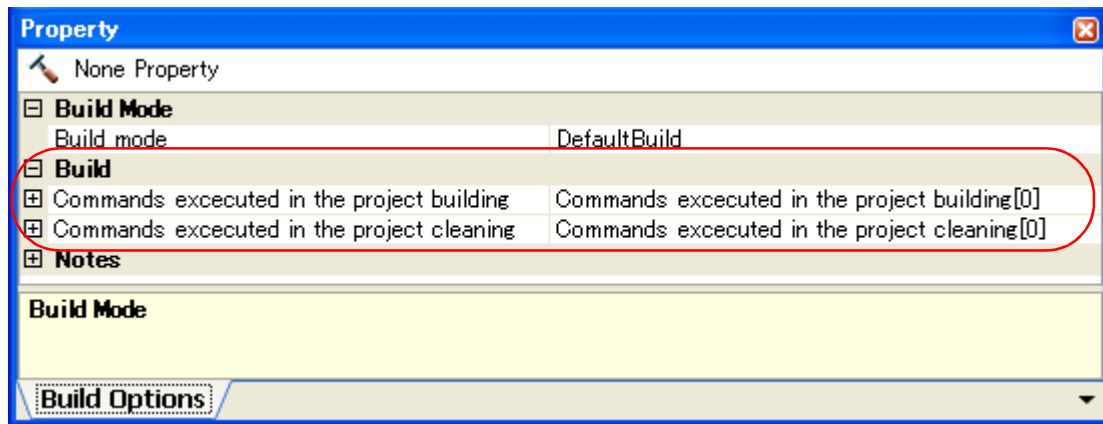
Configure CubeSuite to execute builds linked to an external build tool (e.g. a compiler/assembler other than the build tool provided by CubeSuite).

E.4.1 Set the commands

You can link to an external build tool by executing the command set here when performing build-related operations (see "E.5 Run a Build") on CubeSuite.

The command settings are made with the [Build] category on the [Build Options] tab in the Property panel after selecting the Build Tool node of the target debug-dedicated project (main project or subproject) on the project tree.

Figure E-13. Property Panel: [Build Options] Tab (Build Category)



Below are how to set the commands.

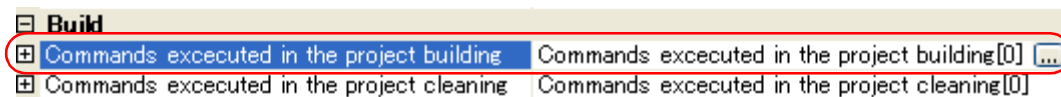
- [Setting the command when running a build](#)
- [Setting the command when running a clean](#)

(1) Setting the command when running a build

Set the command to be executed when running a build (see "E.5.1 Run a build") in the [Commands executed in the project building] property.

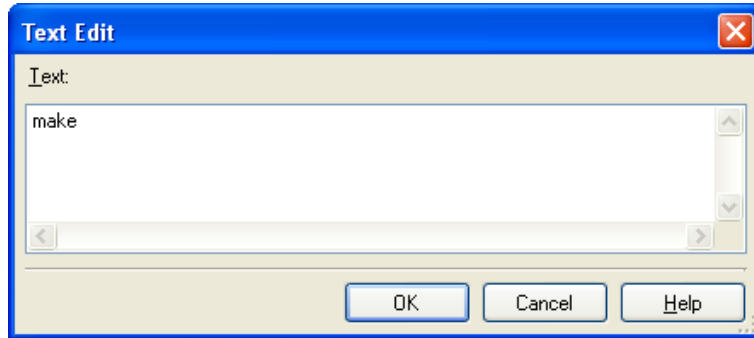
In this property, set the command to execute the required build processing (e.g. "make").

Figure E-14. [Commands executed in the project building] Property



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

Figure E-15. Text Edit Dialog Box



In [Text], enter the command to be executed with one item per line.
 The commands can be specified up to 256 characters per line, up to 256 lines.
 If you click the [OK] button, the entered commands are displayed as subproperties.

Figure E-16. [Commands executed in the project building] Property (After Setting Commands)



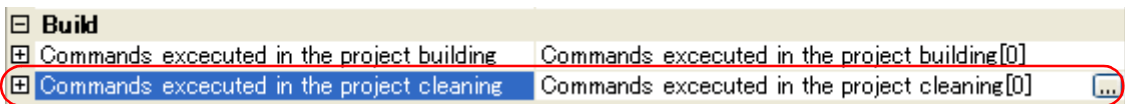
Remark The following macro names are available as embedded macros.

- %BuildModeName%: Replaces with the build mode name.
- %ProjectFolder%: Replaces with the absolute path of the project folder.

(2) Setting the command when running a clean

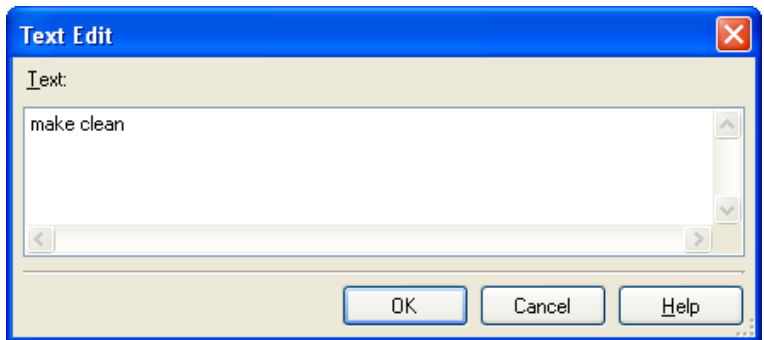
Set the command to be executed when running a clean (see "E.5.3 Run a clean") in the [Commands executed in the project cleaning] property.
 In this property, set the command to delete intermediate files, generated files, and the like output by the build process.

Figure E-17. [Commands executed in the project cleaning] Property



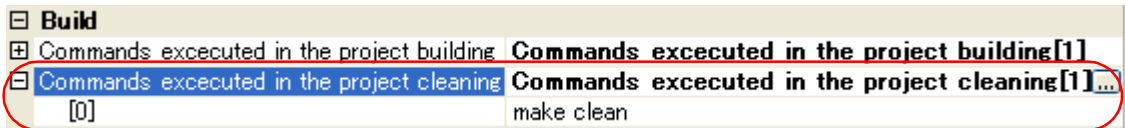
If you click the [...] button, the [Text Edit dialog box](#) will open.

Figure E-18. Text Edit Dialog Box



In [Text], enter the command to be executed with one item per line.
 The commands can be specified up to 256 characters per line, up to 256 lines.
 If you click the [OK] button, the entered commands are displayed as subproperties.

Figure E-19. [Commands executed in the project cleaning] Property (After Setting Commands)



Remark The following macro names are available as embedded macros.

- %BuildModeName%: Replaces with the build mode name.
- %ProjectFolder%: Replaces with the absolute path of the project folder.

To change the specified commands, you can use the [...] button or enter the command directly in the text box of the subproperty.

E.4.2 Set the build mode

You can collectively change the settings of the commands executed according to the purpose of the build.

The commands set in "E.4.1 Set the commands" are organized into what is called "build mode", and by changing the build mode, you eliminate the necessity of changing the command settings every time (i.e. the command settings can be made with each build mode).

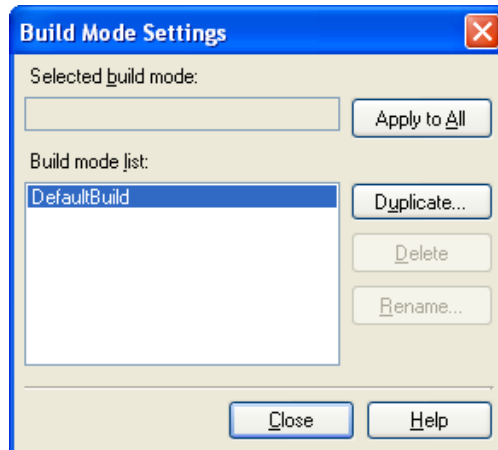
The build mode prepared by default is only "DefaultBuild". Add a build mode according to the purpose of the build. The method to add and change a build mode is shown below.

- [Create a new build mode](#)
- [Change the build mode](#)

(1) Create a new build mode

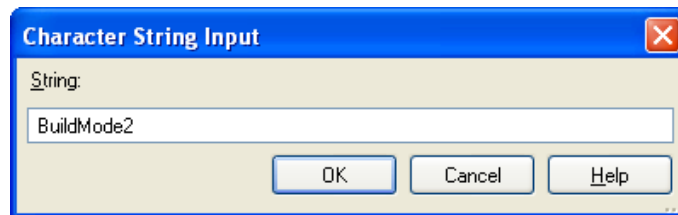
Creating a new build mode is performed with duplicating an existing build mode.
 Select [Build Mode Settings...] from the [Build] menu. The [Build Mode Settings dialog box](#) will open.

Figure E-20. Build Mode Settings Dialog Box



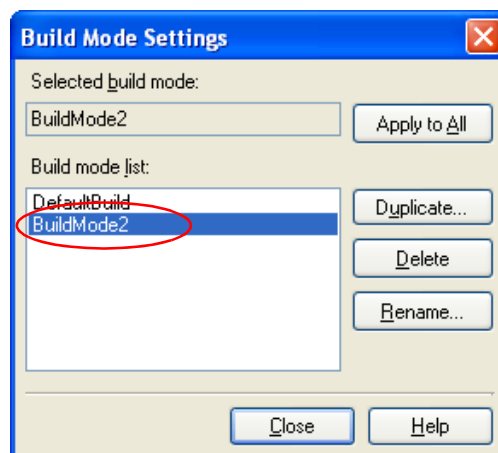
Select the build mode to be duplicated from [Build mode list] and click the [Duplicate...] button. The [Character String Input dialog box](#) will open.

Figure E-21. Character String Input Dialog Box



In the dialog box, enter the name of the build mode to be created and then click the [OK] button. The build mode with that name will be duplicated.
 The created build mode is added to the build modes of the main project and all the subprojects (including projects other than a debug-dedicated project) which currently belong to the project.

Figure E-22. Build Mode Settings Dialog Box (After Adding Build Mode)



Remarks 1. Creating a build mode is regarded a project change.
 When closing the project, you will be asked to confirm whether or not to save the build mode.

2. You can change the name of the build mode by selecting the build mode from the build mode list and clicking the [Rename...] button.
However, you cannot change the name of "DefaultBuild".

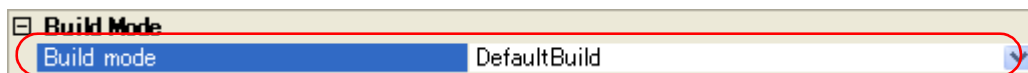
(2) Change the build mode

Change the build mode to the newly created build mode.

(a) When changing the build mode for the main project or subprojects

Select the Build tool node of the target debug-dedicated project on the project tree and select the build mode to be changed to with the [Build mode] property in the [Build Mode] category on the [Build Options] tab in the Property panel.

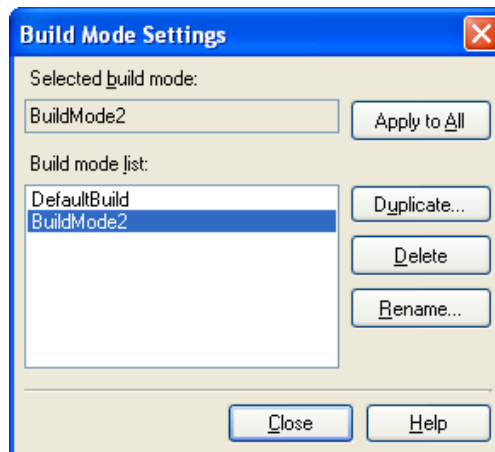
Figure E-23. [Build mode] Property



(b) When changing the build mode for the entire project

Select [Build Mode Settings...] from the [Build] menu. The Build Mode Settings dialog box will open.

Figure E-24. Build Mode Settings Dialog Box



Select the build mode to be changed from [Build mode list]. The selected build mode will be displayed in [Selected build mode].

Click the [Apply to All] button. The build mode for the main project and all the subprojects (including projects other than a debug-dedicated project) which belong to the project will be changed to the build mode selected in the dialog box.

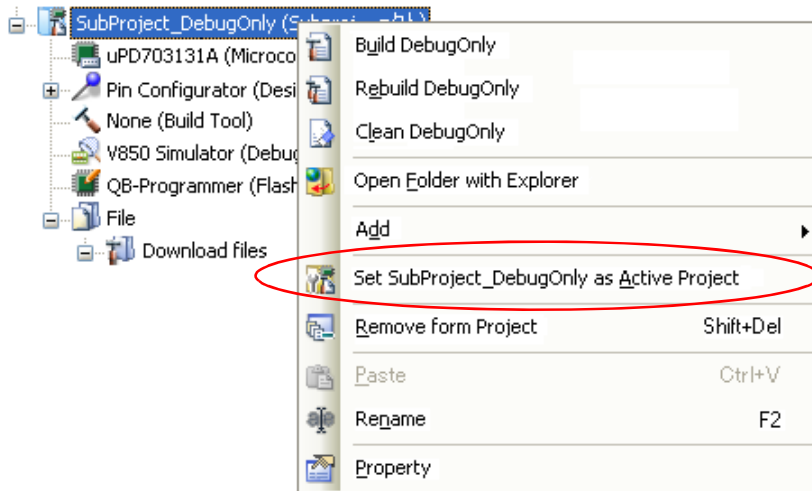
Caution For subprojects that the selected build mode does not exist, the build mode of the subproject is duplicated from the contents of its "DefaultBuild", and then the selected build mode is created newly to the subproject (i.e. the selected build mode with the contents of the subproject is added).

E.4.3 Set the target project for a build

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

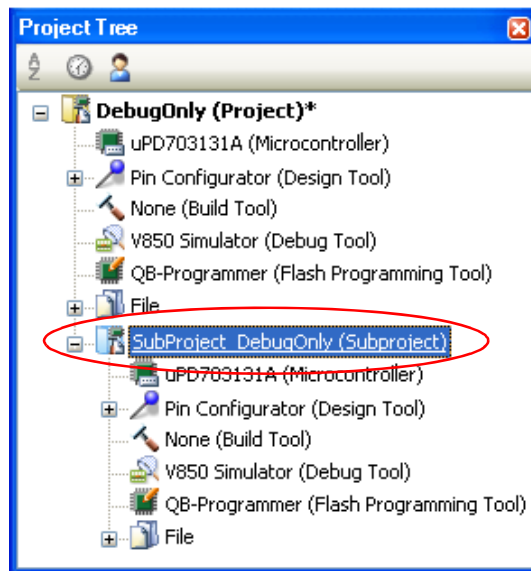
To set the active project, select the Project node to be set as the active project on the project tree and select [Set selected subproject as Active Project] from the context menu.

Figure E-25. [Set selected project as Active Project] Item



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

Figure E-26. Active Project



- Remarks 1.** Immediately after creating a project newly, the main project is the active project.
- 2.** When you remove a subproject that set as the active project from a project, the main project will be the active project.

E.5 Run a Build

This section explains operations related to running a build for a debug-dedicated project.

(1) Build types

The types of operations related to builds which can be linked to an external build tool in a debug-dedicated project are as follows.

Table E-1. Build Types

Type	Description
Build	Executes the command specified with the [Commands executed in the project building] property in the [Build] category in the Property panel . "Build" can run a build of only updated files ^{Note} . -> See " E.5.1 Run a build ".
Rebuild	Executes the command specified with the [Commands executed in the project cleaning] property in the [Build] category in the Property panel , and then executes the command specified with the [Commands executed in the project building] property in the category same as above. "Rebuild" can run a build of all build target files ^{Note} . -> See " E.5.2 Run a rebuild ".
Clean	Executes the command specified with the [Commands executed in the project cleaning] property in the [Build] category in the Property panel . "Clean" can delete all the intermediate files and generated files output by running a build ^{Note} . -> See " E.5.3 Run a clean ".
Rapid build	Runs a build in parallel with the change of the build setting. -> See " E.5.4 Run a rapid build ".
Batch build	Runs builds in batch with the build modes that the project has. -> See " E.5.5 Run a batch build ".

Note It is assumed that a proper command has been set in the [Property panel](#) (see "[E.4.1 Set the commands](#)").

(2) Displaying execution results

The execution results of the build (standard output and standard error from the external build tool used) are displayed in each tab on the [Output panel](#).

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

Figure E-27. Build Execution Results (Build, Rebuild, or Batch Build)

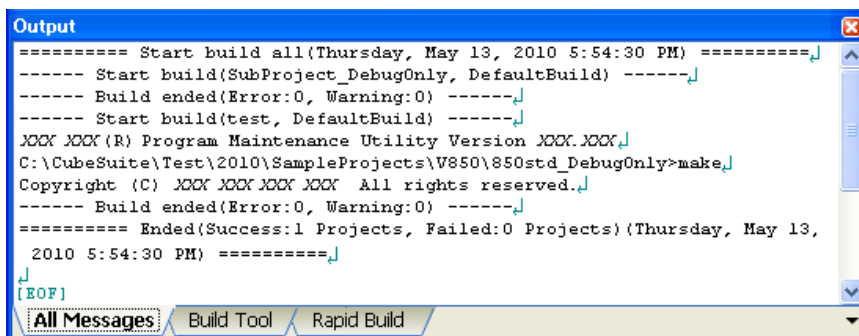


Figure E-28. Build Execution Results (Rapid Build)



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

E.5.1 Run a build

The command specified with the [Commands executed in the project building] property in the [Build] category in the [Property panel](#) is executed (see "(1) [Setting the command when running a build](#)"). This allows you to run a build of only updated files.

The commands being specified in the entire project (main project and subprojects) or active project (see "[E.4.3 Set the target project for a build](#)") can be executed.

Remark If there are files being edited with the [Editor panel](#) when running a build, then all these files are saved.

(1) When running a build of the entire project

The commands being specified in all the debug-dedicated projects that belong to the project are executed.

Click the  button on the toolbar.

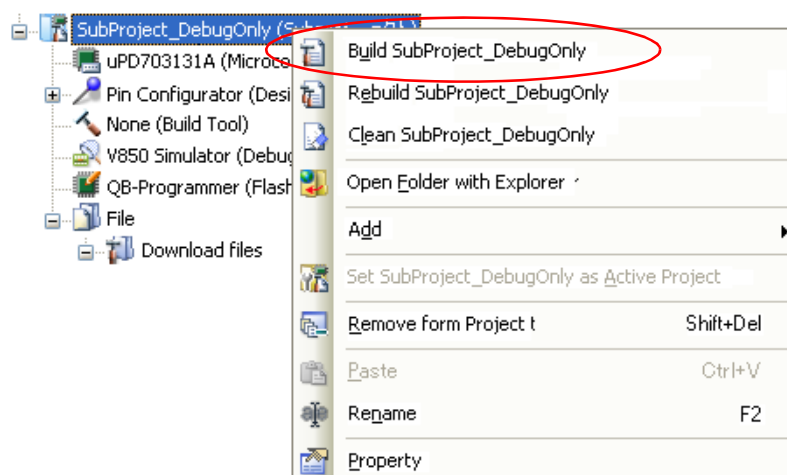
Remarks 1. Builds are run in the order of subproject, main project.

Subprojects are built in the order that they are displayed on the project tree. You can change the display order of the subprojects by dragging the subproject to be moved and dropping it on the desired location.

2. If a project other than a debug-dedicated project is included in your projects, then builds will be run for that project using the build tool provided by CubeSuite (see the "CubeSuite Build").

(2) When running a build of the active project

Select the debug-dedicated project, and then select [Build *active project*] from the context menu.

Figure E-29. [Build *active project*] Item

E.5.2 Run a rebuild

The command specified with the [Commands executed in the project cleaning] property in the [Build] category in the [Property panel](#) is executed (see "(2) [Setting the command when running a clean](#)"), and then the command specified with the [Commands executed in the project building] property in the category same as above is executed (see "(1) [Setting the command when running a build](#)"). This allows you to run a build of all build target files.

The commands being specified in the entire project (main project and subprojects) or active project (see "E.4.3 [Set the target project for a build](#)") can be executed.

Remark If there are files being edited with the [Editor panel](#) when running a rebuild, then all these files are saved.

(1) When running a rebuild of the entire project

The commands being specified in all the debug-dedicated projects that belong to the project are executed.

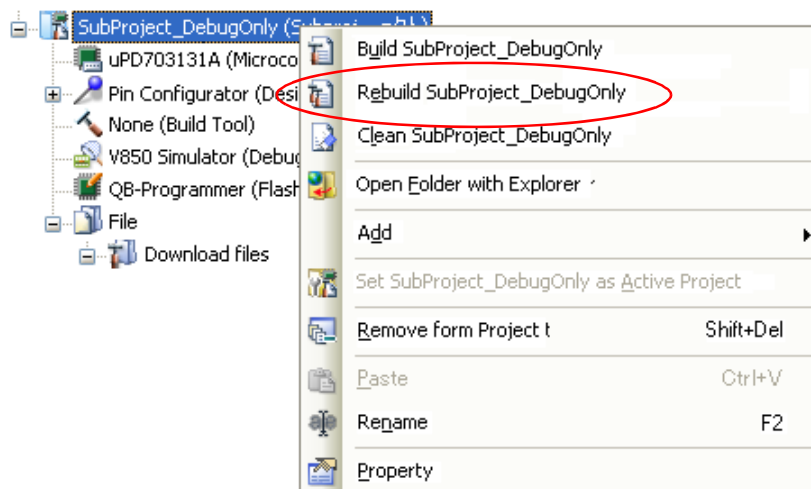
Click the  button on the toolbar.

- Remarks 1.** Rebuilds are run in the order of subproject, main project.
Subprojects are rebuilt in the order that they are displayed on the project tree. You can change the display order of the subprojects by dragging the subproject to be moved and dropping it on the desired location.
- 2.** If a project other than a debug-dedicated project is included in your projects, then rebuilds will be run for that project using the build tool provided by CubeSuite (see the "CubeSuite Build").

(2) When running a rebuild of the active project

Select the debug-dedicated project, and then select [Rebuild *active project*] from the context menu.

Figure E-30. [Rebuild *active project*] Item



E.5.3 Run a clean

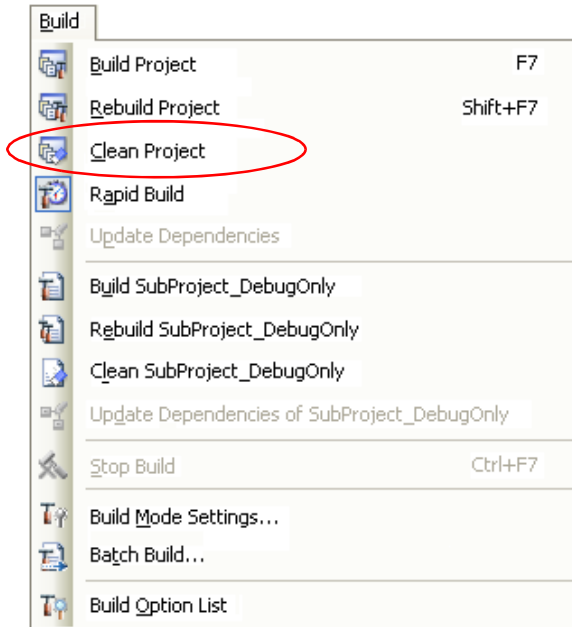
The command specified with the [Commands executed in the project cleaning] property in the [Build] category in the [Property panel](#) is executed (see "(2) [Setting the command when running a clean](#)"). This allows you to delete all the intermediate files and generated files output by running a build.

The commands being specified in the entire project (main project and subprojects) or active project (see "E.4.3 [Set the target project for a build](#)") can be executed.

(1) When running a clean of the entire project

The commands being specified in all the debug-dedicated projects that belong to the project are executed.
 Select [Clean Project] from the [Build] menu.

Figure E-31. [Clean Project] Item



Remarks 1.

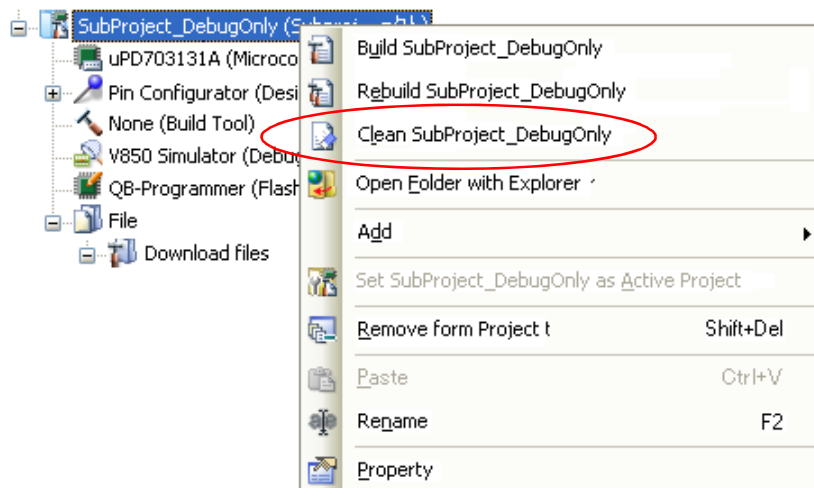
Cleans are run in the order of subproject, main project.
 Subprojects are cleaned in the order that they are displayed on the project tree. You can change the display order of the subprojects by dragging the subproject to be moved and dropping it on the desired location.

2. If a project other than a debug-dedicated project is included in your projects, then cleans will be run for that project using the build tool provided by CubeSuite (see the "CubeSuite Build").

(2) When running a clean of the active project

Select the debug-dedicated project, and then select [Clean active project] from the context menu.

Figure E-32. [Clean active project] Item



E.5.4 Run a rapid build

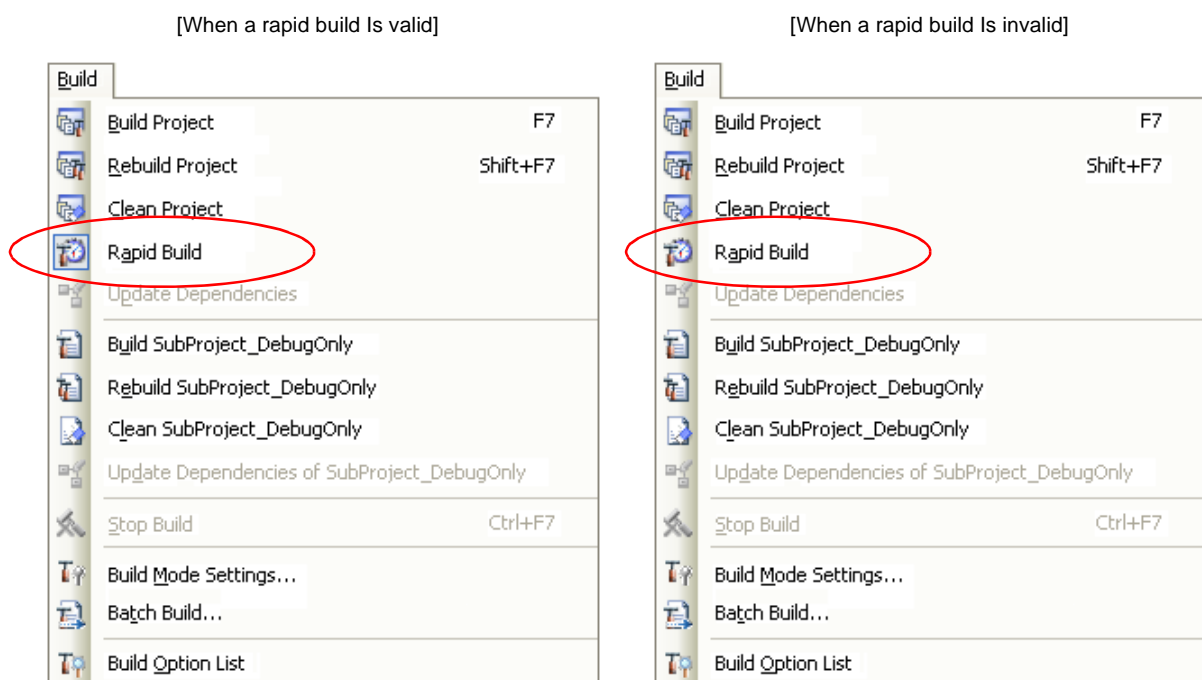
A rapid build is a function that a build (see "E.5.1 Run a build") is started automatically when one of the following events occurs.

- When the C source file, assembler source file, or header file that has been added to the debug-dedicated project is updated
- When the C source file, assembler source file, or header file is added to or removed from the debug-dedicated project
- When the properties in the [Property panel](#) of the debug-dedicated project are changed

If a rapid build is enabled, it is possible to perform a build in parallel with the above operations.

To enable/disable a rapid build, select [Rapid Build] from the [Build] menu. A rapid build is enabled by default.

Figure E-33. [Rapid Build] Item



Caution This function is valid only when editing source files with the [Editor panel](#).

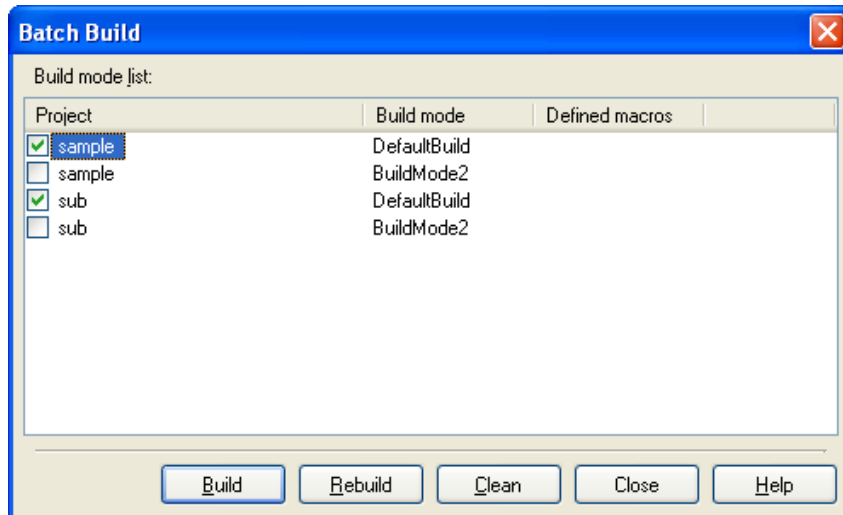
- Remarks 1.** After editing source files, it is recommend to save frequently by pressing the [Ctrl] + [S] key.
2. Enabling/disabling a rapid build is set for the entire project (main project and subprojects).
 3. If you disable a rapid build while it is running, it will be stopped at that time.

E.5.5 Run a batch build

A batch build is a function that builds, rebuilds and cleans are run in batch with the build modes that the project (main project and subproject) has.

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

Figure E-34. Batch Build Dialog Box



In the dialog box, the list of the combinations of the names of the main project and subprojects in the currently opened project and their build modes and macro definitions ([Defined macros] is invalid when the target project is a debug-dedicated project) is displayed.


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

Remarks 1. See the sections below for a build, rebuild, and clean.

- Build: ["E.5.1 Run a build"](#)
- Rebuild: ["E.5.2 Run a rebuild"](#)
- Clean: ["E.5.3 Run a clean"](#)

2. The batch build order follows the project build order, the order of the subprojects, main project. When multiple build modes are selected for a single main project or subproject, after running builds of the subproject with all the selected build modes, the build of the next subproject or main project is run.
3. If there are files being edited with the [Editor panel](#) when running a batch build, then all these files are saved.
4. If a project other than a debug-dedicated project is included in your projects, then builds/rebuilds/cleans will be run for that project using the build tool provided by CubeSuite (see the "CubeSuite Build").

E.5.6 Stop running a build

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

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

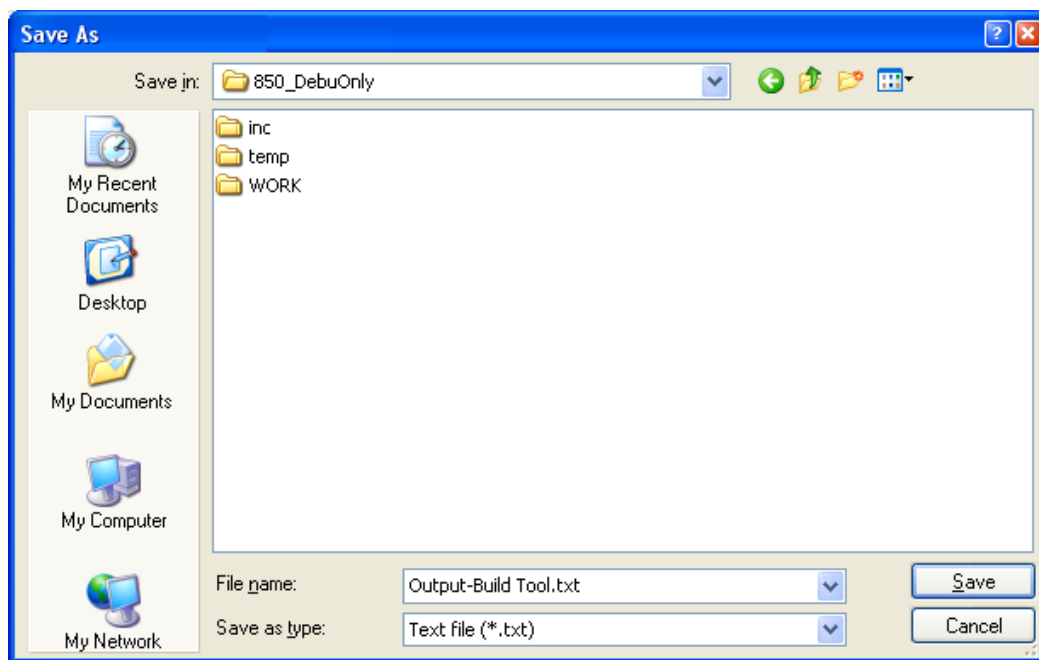
- Build: ["E.5.1 Run a build"](#)
- Rebuild: ["E.5.2 Run a rebuild"](#)
- Batch build: ["E.5.5 Run a batch build"](#)

E.5.7 Save the build results to a file

You can save the execution results of the build (standard output and standard error from the external build tool used) that displayed on the [Output panel](#) as a text file.

Select the [Build Tool] tab on the panel, and then select [Save Output - Build Tool As...] from the [File] menu. The [Save As dialog box](#) will open.

Figure E-35. Save As Dialog Box



In the dialog box, specify a text file name to be saved and the location in which the file is saved, and then click the [Save] button.

APPENDIX F Python CONSOLE/Python FUNCTIONS

This section describes the Python Console and Python functions provided by CubeSuite.

F.1 Overview

The Python Console plug-in is a console tool using the IronPython language.

In addition to the functions and control statements supported by the IronPython language, you can also use Cubesuite Python functions added in order to control CubeSuite.

The functions provided by CubeSuite are shown below.

- On the [Python Console panel](#), you can execute IronPython functions and control statements, and CubeSuite Python functions (see "[F.3 CubeSuite Python Functions](#)" and "[2.10 Execute Python Fuctions](#)").
- When you start CubeSuite from the command line, you can specify and execute a script file (see "[2.11 Manipulate CubeSuite on the Command Line](#)").
- When loading a project file, you can run a script you have prepared in advance (see "[F.2 Related File](#)").

F.2 Related File

Below is a related file of CubeSuite Python functions.

- *project-file-name.py*

If there is a file in the same folder as the project file, and with the same name as the project file but with the "py" extension, then that file is executed automatically when the project file is loaded.

- *download-file-name.py*

If there is a file in the same folder as the download file, and with the same name as the download file but with the "py" extension, then that file is executed automatically after downloading.

F.3 CubeSuite Python Functions

Below is a list of CubeSuite Python functions, classes, and properties.

Table F-1. CubeSuite Python Function List (Common)

Function Name	Function Description
CubeSuiteExit	This function exits from CubeSuite.
Help	This function displays the help for the CubeSuite Python functions.
Hook	This function registers a hook or callback function.
Save	This function saves all editing files and projects.
Source	This function runs a script file.

Table F-2. CubeSuite Python Function List (For Debug Tool)

Function Name	Function Description
debugger.Address	This function evaluates an address expression.
debugger.Assemble.Disassemble	This function performs disassembly.
debugger.Assemble.LineAssemble	This function performs line assembly.
debugger.Breakpoint.Delete	This function deletes a break point.

Function Name	Function Description
debugger.Breakpoint.Disable	This function disables a break point setting.
debugger.Breakpoint.Enable	This function enables a break point setting.
debugger.Breakpoint.Information	This function displays break point information.
debugger.Breakpoint.Set	This function configures a break point.
debugger.Connect	This function connects to the debug tool.
debugger.DebugTool.Change	This function changes the debug tool.
debugger.DebugTool.GetType	This function displays information about the debug tool.
debugger.Disconnect	This function disconnects from the debug tool.
debugger.Download.Binary	This function downloads a binary file.
debugger.Download.Binary64Kb	This function downloads a binary file in within-64 KB format.
debugger.Download.BinaryBank	This function downloads a binary file in memory bank format.
debugger.Download.Coverage	This function downloads coverage data.
debugger.Download.Hex	This function downloads a hex file.
debugger.Download.Hex64Kb	This function downloads a hex file in within-64 KB format.
debugger.Download.HexBank	This function downloads a hex file in memory bank format.
debugger.Download.HexIdTag	This function downloads a hex file with ID tag.
debugger.Download.Information	This function displays download information.
debugger.Download.LoadModule	This function downloads a load module.
debugger.Erase	This function erases the Flash memory.
debugger.GetBreakStatus	This function displays a break condition.
debugger.GetCpuStatus	This function displays the current CPU status.
debugger.GetIeStatus	This function displays the current IE status.
debugger.GetPC	This function displays the PC value.
debugger.Go	This function continues program execution.
debugger.Ie.GetValue debugger.Ie.SetValue	This function sets/references the IE register or DCU register.
debugger.IsConnected	This function checks the connection status of the debug tool.
debugger.IsRunning	This function checks the execution status of the debug tool.
debugger.Jump.File debugger.Jump.Address	This function displays each panel.
debugger.Map.Clear	This function clears the mapping settings.
debugger.Map.Information	This function displays map information.
debugger.Map.Set	This function configures memory mapping.
debugger.Memory.Copy	This function copies the memory.
debugger.Memory.Fill	This function fills the memory.
debugger.Memory.Read	This function references the memory.
debugger.Memory.Write	This function writes to the memory.
debugger.Next	This function performs procedure step execution.

Function Name	Function Description
debugger.Register.GetValue	This function references register/IO register/SFR.
debugger.Register.SetValue	This function sets the value of a register/IO register/SFR.
debugger.Reset	This function resets the CPU.
debugger.ReturnOut	This function runs until control returns to the program that called the current function.
debugger.Run	This function resets and then run the program.
debugger.Step	This function performs step execution.
debugger.Stop	This function stops the execution of the debug tool.
debugger.Upload.Binary	This function saves the memory data in binary format.
debugger.Upload.Coverage	This function saves the coverage data.
debugger.Upload.Intel	This function saves the memory data in Intel format.
debugger.Upload.IntelIdTag	This function saves the memory data in ID-tagged Intel format.
debugger.Upload.Motorola	This function saves the memory data in Motorola format.
debugger.Upload.MotorolaIdTag	This function saves the memory data in ID-tagged Motorola format.
debugger.Upload.Tektronix	This function saves the memory data in Techtronics format.
debugger.Upload.TektronixIdTag	This function saves the memory data in ID-tagged Techtronics format.
debugger.Watch.GetValue	This function references a variable value.
debugger.Watch.SetValue	This function sets a variable value.
debugger.Where	This function displays a stack backtrace.
debugger.Whereami	This function displays a location.
debugger.XCoverage.Clear	This function clears the coverage memory.
debugger.XCoverage.GetCoverage	This function gets the coverage.
debugger.XRunBreak.Delete	This function deletes XRunBreak setting information.
debugger.XRunBreak.Refer	This function displays XRunBreak setting information.
debugger.XRunBreak.Set	This function configures XRunBreak settings.
debugger.XTime	This function displays timing information between Go and Break.
debugger.XTrace.Clear	This function clears the trace memory.
debugger.XTrace.Dump	This function dumps the trace data.

Table F-3. CubeSuite Python Class List

Class Name	Function Description
BreakCondition	This class creates a break condition.

Table F-4. CubeSuite Python Property List

Property Name	Function Description
debugger.Download.CpuReset debugger.Download.FlashErase	This property sets or reference the download property of the debug tool.
debugger.Memory.NoVerify	This property switches the write-time verification setting.

Property Name	Function Description
debugger.Option.Coverage debugger.Option.OpenBreak debugger.Option.Timer debugger.Option.Trace debugger.Option.UseTraceData	This property sets or reference the options of the debug tool.
debugger.XTrace.Addup debugger.XTrace.Complement debugger.XTrace.Mode	This property sets or reference the tracing options of the debug tool.

CubeSuite Python functions have the following limitations.

- If a parameter has a default value, then the [Specification format] parameter is described in the form "*parameter-name=default-value*". You can also specify parameters by value only.

Example If the [Specification format] is "function(*arg1*, *arg2* = 1, *arg3* = True)", then *arg1* has no default value; *arg2* has a default value of 1; and *arg3* has a default value of "True".

The parameters can be specified as follows: "function(*arg1*, *arg2*, *arg3*)".

- Parameters with default values can be omitted.

This is only possible, however, if the parameter can be determined.

Example If the [Specification format] is "function(*arg1*, *arg2* = 1, *arg3* = True)"

```
>>>function("main")           : It is assumed that "function("main", 1, True) "
>>>function("main", 2)        : It is assumed that "function("main", 2, True) "
>>>function("main", arg3 = False) : It is assumed that "function("main", 1, False) "
>>>function("main", False)     : NG because it is assumed that "arg1 = False, arg2 =
"main", arg3 = 3"
```

- You can change the order in which parameters are specified by using the format "*parameter-name=default-value*".

Example If the [Specification format] is "function(*arg1*, *arg2* = 1, *arg3* = True)"

```
>>>function(arg3 = False, arg1 = "main", arg2 = 3)      ...OK
>>>function(False, "main", 3) : NG because it is assumed that "arg1 = False, arg2 =
"main", arg3 = 3"
```


CubeSuiteExit

This function exits from CubeSuite.

[Specification format]

```
CubeSuiteExit ()
```

[Argument(s)]

None

[Return value]

None

[Detailed description]

- This function exits from CubeSuite.

Caution The project file will not be saved, even if it has been modified.

[Example of use]

```
>>>CubeSuiteExit ()
```

Help

This function displays the help for the CubeSuite Python functions.

[Specification format]

```
Help()
```

[Argument(s)]

None

[Return value]

None

[Detailed description]

- This function starts CubeSuite's integrated help, and displays the help for CubeSuite Python functions.

[Example of use]

```
>>>Help()
```

Hook

This function registers a hook or callback function.

Caution Issue this function after connecting the debugging tool.

If it is issued before connecting to the debug tool, then the hook or callback function will not be registered.

[Specification format]

```
Hook(scriptFile)
```

[Argument(s)]

Argument	Description
<i>scriptFile</i>	Specify the script file where the hook or callback function is defined.

[Return value]

None

[Detailed description]

- This function loads *scriptFile*, and registers a hook or callback function.
- Hook functions are called after CubeSuite events occur.
- The types of hook function are shown below.

Note that hook functions do not take parameters.

Hook Function	Event
BeforeDownload	Before download
AfterDownload	After download
AfterCpuReset	After CPU reset
BeforeCpuRun	Before execute
AfterCpuStop	After break

Example Sample script file

```
def BeforeDownload():
    # Processing you want to perform before the download
```

- Hook functions are initialized by the following operations.
 - When a project file is loaded
 - When a new project file is created
 - When the active project is changed
 - When the debugging tool is switched
- Callback functions are called after CubeSuite events occur.

- The name of the callback function is "pythonConsoleCallback".
The parameter of the callback function is the callback trigger.

Argument Value	Callback Trigger
10	After event registration
11	After event deletion
12	Before start of execution
13	After break
14	After CPU reset
18	After debug tool properties are changed
19	Before download
20	After memory or register is changed
63	After time specified by XRunBreak has elapsed

Example Sample script file

```
def pythonConsoleCallback(Id):
    if Id == 63:
        # Processing you want to perform after time specified by XRunBreak has elapsed
```

[Example of use]

```
>>>Hook(".././testScriptFile2.py")
```

Save

This function saves all editing files and projects.

[Specification format]

```
Save ()
```

[Argument(s)]

None

[Return value]

None

[Detailed description]

- This function saves all editing files and projects.

[Example of use]

```
>>>Save ()  
>>>
```

Source

This function runs a script file.

[Specification format]

```
Source(scriptFile)
```

[Argument(s)]

Argument	Description
<i>scriptFile</i>	Specify the script file to run.

[Return value]

None

[Detailed description]

- This function runs the script file specified by *scriptFile*.

[Example of use]

```
>>>Source(".././testScriptFile2.py")
>>>Source("E:/TestFile/TestScript/testScriptFile.py")
>>>
```

debugger.Address

This function evaluates an address expression.

[Specification format]

```
debugger.Address(expression)
```

[Argument(s)]

Argument	Description
<i>expression</i>	Specify an address expression.

[Return value]

Converted address

[Detailed description]

- This function converts the address expression specified by *expression* into the address.

[Example of use]

```
>>>debugger.Address("main")
0x4088
>>>debugger.Address("main + 1")
0x4099
>>>
```

debugger.Assemble.Disassemble

This function performs disassembly.

[Specification format]

```
debugger.Assemble.Disassemble(address, number = 1, code = True)
```

[Argument(s)]

Argument	Description
<i>address</i>	Specify the address at which to start disassembly.
<i>number</i>	Specify the number of lines to display (default: 1).
<i>code</i>	Specify whether to display instruction codes. True: Display instruction codes (default). False: Do not display instruction codes.

[Return value]

None

[Detailed description]

- This function performs disassembly from the address specified by *address*.
- If *number* is specified, the specified number of lines are displayed.
- If *code* is set to "False", then instruction codes are not displayed.

[Example of use]

```
>>>debugger.Assemble.Disassemble("main")
0x00004088 F545 br _TestInit+0x8e
>>>debugger.Assemble.Disassemble("main", 2)
0x00004088 F545 br _TestInit+0x8e
0x0000408A 0A5A mov 0xa, r11
>>>debugger.Assemble.Disassemble("main", 5, False)
0x00004088 br _TestInit+0x8e
0x0000408A mov 0xa, r11
0x0000408C movea 0x19, r0, r13
0x00004090 mov r13, r12
0x00004092 movhi 0xffff, gp, r1
>>>
```


debugger.Assemble.LineAssemble

This function performs line assembly.

[Specification format]

```
debugger.Assemble.LineAssemble(address, code)
```

[Argument(s)]

Argument	Description
<i>address</i>	Specify the address at which to start assembly.
<i>code</i>	Specify the string to assemble.

[Return value]

None

[Detailed description]

- This function performs assembly of the string specified by *code* from the address specified by *address*.
- If "." is specified in *address*, then it is interpreted as the address following the last address assembled.

[Example of use]

```
>>>debugger.Assemble.Disassemble("main")
0x00004088 F545 br _TestInit+0x8e
>>>debugger.Assemble.Disassemble(".")
0x0000408A 0A5A mov 0xa, r11
>>>debugger.Assemble.LineAssemble("main", "mov r13, r12")
>>>debugger.Assemble.Disassemble("main", 1, False)
0x00004088 mov r13, r12
>>>
```

debugger.Breakpoint.Delete

This function deletes a break point.

[Specification format]

```
debugger.Breakpoint.Delete(breakNumber = "")
```

[Argument(s)]

Argument	Description
<i>breakNumber</i>	Specify the break event number to delete.

[Return value]

None

[Detailed description]

- This function deletes the break event specified by *breakNumber*.
- If *breakNumber* is not specified, then breaks of all break event numbers will be deleted.

[Example of use]

```
>>>debugger.Breakpoint.Enable(1)
>>>debugger.Breakpoint.Disable(1)
>>>debugger.Breakpoint.Delete(1)
>>>
```

debugger.Breakpoint.Disable

This function disables a break point setting.

[Specification format]

```
debugger.Breakpoint.Disable(breakNumber = "")
```

[Argument(s)]

Argument	Description
<i>breakNumber</i>	Specify the break event number to disable.

[Return value]

None

[Detailed description]

- This function disables the break event specified by *breakNumber*.
- If *breakNumber* is not specified, then breaks of all break event numbers will be disabled.

[Example of use]

```
>>>debugger.Breakpoint.Enable(1)
>>>debugger.Breakpoint.Disable(1)
>>>debugger.Breakpoint.Delete(1)
>>>
```

debugger.Breakpoint.Enable

This function enables a break point setting.

[Specification format]

```
debugger.Breakpoint.Enable(breakNumber = "")
```

[Argument(s)]

Argument	Description
<i>breakNumber</i>	Specify the break event number to enable.

[Return value]

None

[Detailed description]

- This function enables the break event specified by *breakNumber*.
- If *breakNumber* is not specified, then breaks of all break event numbers will be enabled.

[Example of use]

```
>>>debugger.Breakpoint.Enable(1)
>>>debugger.Breakpoint.Disable(1)
>>>debugger.Breakpoint.Delete(1)
>>>
```

debugger.Breakpoint.Information

This function displays break point information.

[Specification format]

```
debugger.Breakpoint.Information()
```

[Argument(s)]

None

[Return value]

None

[Detailed description]

- This function displays the break point settings in the following format.

```
break-event-number break-name state address-location
```

[Example of use]

```
>>>debugger.Breakpoint.Information()  
1 Break 0001 Disable test.c#Test_Init+32  
3 Break 0003 Enable test.c#Test_Init+10  
4 Break 0004 Disable test.c#Test_Init+44  
5 Break 0005 Enable disp.c#Test_Init+12  
7 Break 0007 Enable disp.c#Test_Init+44  
>>>
```

debugger.Breakpoint.Set

This function configures a break point.

[Specification format]

```
debugger.Breakpoint.Set (BreakCondition)
```

[Argument(s)]

Argument	Description
<i>BreakCondition</i>	Specify a break condition. See " BreakCondition " for details about creating break conditions.

[Return value]

The set break event number

[Detailed description]

- This function sets a break point according to the specifications in *BreakCondition*.

[Example of use]

```
>>>Brean = BreakCondition()
>>>Break.Address = "main"
>>>breakNumber = debugger.Breakpoint.Set (Break)
>>>print breakNumber
1
>>>
```

debugger.Connect

This function connects to the debug tool.

[Specification format]

```
debugger . Connect ()
```

[Argument(s)]

None

[Return value]

None

[Detailed description]

- This function connects to the debug tool.

[Example of use]

```
>>>debugger . Connect ()  
>>>
```

debugger.DebugTool.Change

This function changes the debug tool.

[Specification format]

```
debugger.DebugTool.Change(debugTool)
```

[Argument(s)]

Argument	Description	
<i>debugTool</i>	Specify the debug tool to change. The debug tools that can be specified are shown below.	
	Type	Description
	DebugTool.Simulator	Simulator
	DebugTool.Minicube	MINICUBE
	DebugTool.Minicube2	MINICUBE2
	DebugTool.Iecube	IECUBE
	DebugTool.Iecube2	IECUBE2
	DebugTool.E1Jtag	E1 (JTAG connect)
	DebugTool.E1Serial	E1 (Serial connect)

[Return value]

None

[Detailed description]

- This function changes to the debug tool specified by *DebugTool*.
However, the debug tool that can be changed differs depending on the using device. Select [Debug Tool] on the project tree and select [Using Debug Tool] on the context menu. And then confirm the debug tool that can be changed.

[Example of use]

```
>>>debugger.DebugTool.Change(DebugTool.Simulator)
>>>
```


debugger.DebugTool.GetType

This function displays information about the debug tool.

[Specification format]

```
debugger.DebugTool.GetType()
```

[Argument(s)]

None

[Return value]

Debug tool type

Type	Description
DebugTool.Simulator	Simulator
DebugTool.Minicube	MINICUBE
DebugTool.Minicube2	MINICUBE2
DebugTool.Iecube	IECUBE
DebugTool.Iecube2	IECUBE2
DebugTool.E1Jtag	E1 (JTAG connect)
DebugTool.E1Serial	E1 (Serial connect)

[Detailed description]

- This function displays information about the debug tool.

[Example of use]

```
>>>debugType = debugger.DebugTool.GetType()
DebugTool.Minicube2
>>>if debugType != DebugType.Simulator:
... debugger.DebugTool.Change(DebugTool.Simulator)
...
>>>
```

debugger.Disconnect

This function disconnects from the debug tool.

[Specification format]

```
debugger.Disconnect ()
```

[Argument(s)]

None

[Return value]

None

[Detailed description]

- This function disconnects from the debug tool.

[Example of use]

```
>>>debugger.Disconnect ()
```

debugger.Download.Binary

This function downloads a binary file.

[Specification format]

```
debugger.Download.Binary(fileName, address, append = False)
```

[Argument(s)]

Argument	Description
<i>fileName</i>	Specify a download file.
<i>address</i>	Specify a download start address.
<i>append</i>	Specify whether to make an additional download. True: Perform additional download. False: Perform overwrite download (default).

Caution If two or more parameters are specified, then all three parameters must be specified.
It is not possible to specify only *filename* and *address*.

[Return value]

None

[Detailed description]

- This function downloads data in binary format.

[Example of use]

```
>>>debugger.Download.Binary("C:/test/testModule.bin", 0x1000, False)
>>>debugger.Download.Binary("C:/test/testModule2.bin", 0x2000, True)
>>>
```

debugger.Download.Binary64Kb

This function downloads a binary file in within-64 KB format.

[Specification format]

```
debugger.Download.Binary64Kb(fileName, address, append = False)
```

[Argument(s)]

Argument	Description
<i>fileName</i>	Specify a download file.
<i>address</i>	Specify a download start address.
<i>append</i>	Specify whether to make an additional download. True: Perform additional download. False: Perform overwrite download (default).

Caution If two or more parameters are specified, then all three parameters must be specified.
It is not possible to specify only *filename* and *address*.

[Return value]

None

[Detailed description]

- When using the memory bank, this function downloads binary files in within-64 KB format.

[Example of use]

```
>>>debugger.Download.Binary64Kb("C:/test/testModule.bin", 0x1000, False)
>>>debugger.Download.Binary64Kb("C:/test/testModule2.bin", 0x2000, True)
>>>
```

debugger.Download.BinaryBank

This function downloads a binary file in memory bank format.

[Specification format]

```
debugger.Download.BinaryBank(fileName, address, append = False)
```

[Argument(s)]

Argument	Description
<i>fileName</i>	Specify a download file.
<i>address</i>	Specify a download start address.
<i>append</i>	Specify whether to make an additional download. True: Perform additional download. False: Perform overwrite download (default).

Caution If two or more parameters are specified, then all three parameters must be specified.
It is not possible to specify only *filename* and *address*.

[Return value]

None

[Detailed description]

- When using the memory bank, this function downloads binary files in memory bank format.

[Example of use]

```
>>>debugger.Download.BinaryBank("C:/test/testModule.bin", 0x1000, False)
>>>debugger.Download.BinaryBank("C:/test/testModule2.bin", 0x2000, True)
>>>
```

debugger.Download.Coverage

This function downloads coverage data.

[Specification format]

```
debugger.Download.Coverage(fileName)
```

[Argument(s)]

Argument	Description
<i>fileName</i>	Specify a coverage data file.

[Return value]

None

[Detailed description]

- This function downloads coverage data.

[Example of use]

```
>>>debugger.Download.Coverage("testModule.csrcv")  
>>>
```

debugger.Download.Hex

This function downloads a hex file.

[Specification format]

```
debugger.Download.Hex(fileName, offset = 0, append = False)
```

[Argument(s)]

Argument	Description
<i>fileName</i>	Specify a download file.
<i>offset</i>	Specify an offset (default: 0).
<i>append</i>	Specify whether to make an additional download. True: Perform additional download. False: Perform overwrite download (default).

Caution If two or more parameters are specified, then all three parameters must be specified.
It is not possible to specify only *filename* and *offset*.

[Return value]

None

[Detailed description]

- This function downloads data in hex format.

[Example of use]

```
>>>debugger.Download.Hex("testModule.hex")
>>>
```

debugger.Download.Hex64Kb

This function downloads a hex file in within-64 KB format.

[Specification format]

```
debugger.Download.Hex64Kb(fileName, offset = 0, append = False)
```

[Argument(s)]

Argument	Description
<i>fileName</i>	Specify a download file.
<i>offset</i>	Specify an offset (default: 0).
<i>append</i>	Specify whether to make an additional download. True: Perform additional download. False: Perform overwrite download (default).

Caution If two or more parameters are specified, then all three parameters must be specified.
It is not possible to specify only *filename* and *offset*.

[Return value]

None

[Detailed description]

- When using the memory bank, this function downloads hex files in within-64 KB format.

[Example of use]

```
>>>debugger.Download.Hex64Kb("testModule.hex")
>>>
```


debugger.Download.HexBank

This function downloads a hex file in memory bank format.

[Specification format]

```
debugger.Download.HexBank(fileName, offset = 0, append = False)
```

[Argument(s)]

Argument	Description
<i>fileName</i>	Specify a download file.
<i>offset</i>	Specify an offset (default: 0).
<i>append</i>	Specify whether to make an additional download. True: Perform additional download. False: Perform overwrite download (default).

Caution If two or more parameters are specified, then all three parameters must be specified.
It is not possible to specify only *filename* and *offset*.

[Return value]

None

[Detailed description]

- When using the memory bank, this function downloads hex files in memory-bank format.

[Example of use]

```
>>>debugger.Download.HexBank("testModule.hex")
>>>debugger.Download.HexBank("testModule2.hex", 0x1000, True)
>>>
```

debugger.Download.HexIdTag

This function downloads a hex file with ID tag.

[Specification format]

```
debugger.Download.HexIdTag(fileName, offset = 0, append = False)
```

[Argument(s)]

Argument	Description
<i>fileName</i>	Specify a download file.
<i>offset</i>	Specify an offset (default: 0).
<i>append</i>	Specify whether to make an additional download. True: Perform additional download. False: Perform overwrite download (default).

Caution If two or more parameters are specified, then all three parameters must be specified.
It is not possible to specify only *filename* and *offset*.

[Return value]

None

[Detailed description]

- This function downloads a hex file with ID tag.

[Example of use]

```
>>>debugger.Download.HexIdTag("testModule.hex")
>>>debugger.Download.HexIdTag("testModule2.hex", 0x1000, True)
>>>
```

debugger.Download.Information

This function displays download information.

[Specification format]

```
debugger.Download.Information()
```

[Argument(s)]

None

[Return value]

None

[Detailed description]

- This function displays download information in the following format.

```
download-number download-file-name
```

[Example of use]

```
>>>debugger.Download.Information()  
1 DefaultBuild\test.lmf
```

debugger.Download.LoadModule

This function downloads a load module.

[Specification format]

```
debugger.Download.LoadModule(fileName = "", downloadOption = DownloadOption.Both, append = False)
```

[Argument(s)]

Argument	Description	
<i>fileName</i>	Specify a download file.	
<i>downloadOption</i>	Specify an option. The options that can be specified are shown below.	
	Type	Description
	DownloadOption.NoSymbol	Do not load symbol information.
	DownloadOption.SymbolOnly	Only load symbol information.
	DownloadOption.Both	Load both symbol information and object information (default).
<i>append</i>	Specify whether to make an additional download. True: Perform additional download. False: Perform overwrite download (default).	

Caution If *offset* is specified, then *append* must also be specified.
Similarly, if *append* is specified, then *offset* must also be specified.

[Return value]

None

[Detailed description]

- This function downloads a load module.
- If *fileName* is not specified, the file specified on the [Download File Settings] tab in the Property panel of the debugging tool is downloaded.
- If *downloadOption* is specified, the processing is performed in accordance with the specification.

[Example of use]

```
>>>debugger.Download.LoadModule("testModule.lmf")
>>>debugger.Download.LoadModule("testModule2.lmf", 0x1000, DownloadOption.SymbolOnly, True)
>>>
```

debugger.Erase

This function erases the Flash memory.

[Specification format]

```
debugger.Erase(eraseOption = EraseOption.Code)
```

[Argument(s)]

Argument	Description	
<i>eraseOption</i>	Specify an option. The options that can be specified are shown below.	
	Type	Description
	EraseOption.Code	Erase the code flash memory (default). [MINICUBE] [MINICUBE2]
	EraseOption.Data	Erase the data flash memory.
	EraseOption.External	Erase the flash memory in external space.

Caution IECUBE and the simulator do not have functionality to delete code flash memory. For this reason, if you are using IECUBE or the simulator, you cannot omit *eraseOption*, or specify "EraseOption.Code".

[Return value]

None

[Detailed description]

- This function erases the flash memory, specified by *eraseOption*.

[Example of use]

```
>>>debugger.Erase()
>>>debugger.Erase(EraseOption.External)
>>>
```

debugger.GetBreakStatus

This function displays a break condition.

[Specification format]

```
debugger.GetBreakStatus()
```

[Argument(s)]

None

[Return value]

Break-trigger string

Break-trigger String	Description
None	Break not in effect
Manual	Manual
Event	Event
Software	Software
TraceFull	Trace full
NonMap	Non map
WriteProtect	Write protect
IorIllegal	IOR Illegal
UninitializeMemoryRead	Uninitialize memory read
Temporary	Temporary

- Remarks 1.** Returns the string portion of the "BreakStatus" enum.
2. Determine conditions by writing in the format "BreakStatus.string".

[Detailed description]

- This function displays break-trigger string.
 During execution, this will be "None".

[Example of use]

```
>>>debugger.GetBreakStatus()  
Temporary  
>>>a = debugger.GetBreakStatus()  
Temporary  
>>>print a  
Temporary  
>>>if (debugger.GetBreakStatus() == BreakStatus.Temporary):  
... print "Temporary break"  
...  
Temporary  
Temporary break  
>>>
```

debugger.GetCpuStatus

This function displays the current CPU status.

[Specification format]

```
debugger.GetCpuStatus ()
```

[Argument(s)]

None

[Return value]

Current CPU status

CPU Status	Description
Hold	Path on hold
HoldStopIdle	HOLD/SOFT STOP/HARD STOP/IDLE mode
PowOff	No power supply to target
Reset	Reset state
Standby	Standby mode
Stop	Stop mode
StopIdle	SOFT STOP/HARD STOP/IDLE mode
Wait	Wait state
None	N/A

[Detailed description]

- This function displays the current CPU status.

[Example of use]

```
>>>debugger.GetCpuStatus ()
Stop
>>>
```


debugger.GetIeStatus

This function displays the current IE status.

[Specification format]

```
debugger.GetIeStatus()
```

[Argument(s)]

None

[Return value]

Current IE status

IE Status	Description
Break	Break in effect
Coverage	Coverage running
Timer	Timer running
Tracer	Trace running
Step	Step executing
Run	User program running

[Detailed description]

- This function displays the current IE status.

[Example of use]

```
>>>debugger.GetIeStatus()
None
>>>
```

debugger.GetPC

This function displays the PC value.

[Specification format]

```
debugger.GetPC()
```

[Argument(s)]

None

[Return value]

PC value

[Detailed description]

- This function displays the PC value.

[Example of use]

```
>>>debugger.GetPC()  
0x92B0
```

debugger.Go

This function continues program execution.

[Specification format]

```
debugger.Go (goOption = GoOption.Normal)
```

[Argument(s)]

Argument	Description	
<i>goOption</i>	Specify an option. The options that can be specified are shown below.	
	Type	Description
	GoOption.IgnoreBreak	Execute ignoring breakpoints.
	GoOption.WaitBreak	Wait until program stops.
	GoOption.Normal	Breakpoints enabled; do not wait until program stops (default).

[Return value]

None

[Detailed description]

- This function continues program execution.
- If *goOption* is specified, the processing is performed in accordance with the specification.

[Example of use]

```
>>>debugger.Go ()
>>>debugger.Go (GoOption.WaitBreak)
>>>
```

debugger.Ie.GetValue

debugger.Ie.SetValue

This function sets/references the IE register or DCU register.

[Specification format]

```
debugger.Ie.GetValue(ieType, address)
debugger.Ie.SetValue(ieType, address, value)
```

[Argument(s)]

Argument	Description						
<i>ieType</i>	Specify a register. The registers that can be specified are shown below.						
	<table border="1"> <thead> <tr> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>IeType.Reg</td> <td>IE register [78K0] [78K0R] [V850 [IECUBE]]</td> </tr> <tr> <td>IeType.Dcu</td> <td>DCU register [V850 [IECUBE]]</td> </tr> </tbody> </table>	Type	Description	IeType.Reg	IE register [78K0] [78K0R] [V850 [IECUBE]]	IeType.Dcu	DCU register [V850 [IECUBE]]
	Type	Description					
	IeType.Reg	IE register [78K0] [78K0R] [V850 [IECUBE]]					
IeType.Dcu	DCU register [V850 [IECUBE]]						
IeType.Reg	IE register [78K0] [78K0R] [V850 [IECUBE]]						
IeType.Dcu	DCU register [V850 [IECUBE]]						
<i>address</i>	Specify the address to reference/set.						
<i>value</i>	Specify the setting value.						

[Return value]

debugger.Ie.GetValue is the register value

There is no debugger.Ie.SetValue

[Detailed description]

- debugger.Ie.GetValue displays the value of the register specified by *address*.
The register type is specified by *ieType*.
- debugger.Ie.SetValue writes *value* to the register specified by *address*.
The register type is specified by *ieType*.

Remark When the DCU register is referenced, the register value is reset to 0.

[Example of use]

```
>>>debugger.Ie.GetValue(IeType.Reg, 0x100)
0x12
>>>debugger.Ie.SetValue(IeType.Reg, 0x100, 0x10)
>>>debugger.Ie.GetValue(IeType.Reg, 0x100)
0x10
>>>
```

debugger.IsConnected

This function checks the connection status of the debug tool.

[Specification format]

```
debugger.IsConnected()
```

[Argument(s)]

None

[Return value]

If the debug tool is connected: True

If the debug tool is not connected: False

[Detailed description]

- This function checks the connection status of the debug tool.

[Example of use]

```
>>>if debugger.IsConnected() == True :  
... print "OK"  
...  
OK  
>>>
```

debugger.IsRunning

This function checks the execution status of the debug tool.

[Specification format]

```
debugger.IsRunning()
```

[Argument(s)]

None

[Return value]

If the debug tool is running: True

If the debug tool is not running: False

[Detailed description]

- This function checks the execution status of the debug tool.

[Example of use]

```
>>>if debugger.IsRunning() == True :  
... print "OK"  
...  
OK  
>>>
```

debugger.Jump.File
debugger.Jump.Address

This function displays each panel.

[Specification format]

```
debugger.Jump.File(fileName, lineNumber = 1)
debugger.Jump.Address(jumpType, address = 0)
```

[Argument(s)]

Argument	Description	
<i>fileName</i>	Specify the name of the file to display.	
<i>lineNumber</i>	Specify the line to display (default: 1).	
<i>jumpType</i>	Specify the type of panel to display. The panel types that can be specified are shown below.	
	Type	Description
	JumpType.Source	Editor panel
	JumpType.Assemble	Disassemble panel
	JumpType.Memory	Memory panel
<i>address</i>	Specify the address to display (default: 0).	

[Return value]

None

[Detailed description]

- debugger.Jump.File displays the file specified by *fileName* in the Editor Panel.
If *lineNumber* is specified, then the line specified by *lineNumber* in the file specified by *fileName* is displayed.
- debugger.Jump.Address displays the panel specified by *jumpType*.
If *address* is specified, then the area corresponding to the specified address is displayed.

[Example of use]

```
>>>debugger.Jump.File("testJump.c")
>>>debugger.Jump.File("testJump.h", 25)
>>>debugger.Jump.Address(JumpType.Memory, 0x2000)
>>>
```

debugger.Map.Clear

This function clears the mapping settings.

[Specification format]

```
debugger.Map.Clear()
```

[Argument(s)]

None

[Return value]

None

[Detailed description]

- This function clears the mapping settings.

[Example of use]

```
>>>debugger.Map.Clear()  
>>>
```


debugger.Map.Information

This function displays map information.

[Specification format]

```
debugger.Map.Information()
```

[Argument(s)]

None

[Return value]

None

[Detailed description]

- This function displays map information.

```
number: start-address end-address access-size memory-type
```

[Example of use]

```
>>>debugger.Map.Information()  
1: 0x00000000 0x0005FFFF 32 (Internal ROM area)  
2: 0x00060000 0x03FF6FFF 8 (Non map area)  
3: 0x03FF7000 0x03FF6FFF 32 (Internal RAM area)  
4: 0x03FFF000 0x03FFFFFF 8 (SFR)  
>>>
```

debugger.Map.Set

This function configures memory mapping.

[Specification format]

```
debugger.Map.Set (mapType, address1, address2, accessSize = 8, cs = "")
```

[Argument(s)]

Argument	Description														
<i>mapType</i>	Specify a memory type. The memory types that can be specified are shown below.														
	<table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">Type</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>MapType.EmulationRom</td> <td>Alternative ROM</td> </tr> <tr> <td>MapType.EmulationRam</td> <td>AlternativeRAM</td> </tr> <tr> <td>MapType.Traget</td> <td>Target area</td> </tr> <tr> <td>MapType.TargetRom</td> <td>Target ROM area</td> </tr> <tr> <td>MapType.Stack</td> <td>Stack area</td> </tr> <tr> <td>MapType.Protect</td> <td>I/O protect area</td> </tr> </tbody> </table>	Type	Description	MapType.EmulationRom	Alternative ROM	MapType.EmulationRam	AlternativeRAM	MapType.Traget	Target area	MapType.TargetRom	Target ROM area	MapType.Stack	Stack area	MapType.Protect	I/O protect area
	Type	Description													
	MapType.EmulationRom	Alternative ROM													
	MapType.EmulationRam	AlternativeRAM													
	MapType.Traget	Target area													
	MapType.TargetRom	Target ROM area													
	MapType.Stack	Stack area													
MapType.Protect	I/O protect area														
<i>address1</i>	Specify a map start address.														
<i>address2</i>	Specify a map end address.														
<i>accessSize</i>	Specify an access size (bit) (default: 8). For V850, specify either 8, 16, or 32. For 78K0R [IECUBE], specify either 8 or 16.														
<i>cs</i>	Specify the chip select (default: not specified). When mapping emulation memory (alternative ROM/RAM) in the V850 [IECUBE], specify the one of the following chip selects as a string: cs0, cs1, cs2, cs3, cs4, cs5, cs6, or cs7. For models in the V850ES series, however, the chip select allocation is fixed, or the chip select will not function, so this can be omitted. <i>If chip select is specified, then accessSize cannot be omitted.</i>														

[Return value]

None

[Detailed description]

- This function performs memory mapping with the memory type specified by *mapType*.

[Example of use]

```
>>>debugger.Map.Set (MapType.EmulationRom, 0x100000, 0x10ffff)
>>>
```

debugger.Memory.Copy

This function copies the memory.

[Specification format]

```
debugger.Memory.Copy(address1, address2, address3)
```

[Argument(s)]

Argument	Description
<i>address1</i>	Specify the start address to copy from.
<i>address2</i>	Specify the end address to copy from.
<i>address3</i>	Specify the address to copy to.

[Return value]

None

[Detailed description]

- This function copies the memory from *address1* to *address2* into *address3*.

[Example of use]

```
>>>debugger.Memory.Copy(0x1000, 0x2000, 0x3000)
>>>
```

debugger.Memory.Fill

This function copies the memory.

[Specification format]

```
debugger.Memory.Fill(address1, address2, value, memoryOption = MemoryOption.Byte)
```

[Argument(s)]

Argument	Description	
<i>address1</i>	Specify the start address to fill.	
<i>address2</i>	Specify the end address to fill to.	
<i>value</i>	Specify the fill value.	
<i>memoryOption</i>	Specify the fill unit. The units that can be specified are shown below.	
	Type	Description
	MemoryOption.Byte	Byte unit (8 bits) (default)
	MemoryOption.Halfword	Half-word unit (16 bits) [V850]
	MemoryOption.Word	Word unit (78K: 16 bits, V850: 32 bits)

[Return value]

None

[Detailed description]

- This function fills from *address1* to *address2* with *value*.
- If *memoryOption* is specified, fill according to that specification.

[Example of use]

```
>>>debugger.Memory.Fill(0x1000, 0x2000, 0xFF)
>>>debugger.Memory.Fill(0x2000, 0x3000, 0x0A, MemoryOption.Word)
>>>
```

debugger.Memory.Read

This function fills the memory.

[Specification format]

```
debugger.Memory.Read(address, memoryOption = MemoryOption.Byte)
```

[Argument(s)]

Argument	Description	
<i>address</i>	Specify the address to reference.	
<i>memoryOption</i>	Specify the display unit. The units that can be specified are shown below.	
	Type	Description
	MemoryOption.Byte	Byte unit (8 bits) (default)
	MemoryOption.Halfword	Half-word unit (16 bits) [V850]
	MemoryOption.Word	Word unit (78K: 16 bits, V850: 32 bits)

[Return value]

The referenced memory value

[Detailed description]

- This function displays the address specified by *address*, according to *memoryOption* in hexadecimal format.

[Example of use]

```
>>>debugger.Memory.Read(0x100)
0x10
>>>value = debugger.Memory.Read(0x100)
0x10
>>>print value
16
>>>debugger.Memory.Read(0x100, MemoryOption.HalfWord)
0x0010
>>>
```

debugger.Memory.Write

This function references the memory.

[Specification format]

```
debugger.Memory.Write(address, value, memoryOption = MemoryOption.Byte)
```

[Argument(s)]

Argument	Description	
<i>address</i>	Specify the address to set.	
<i>value</i>	Specify the value to set.	
<i>memoryOption</i>	Specify the unit to set. The units that can be specified are shown below.	
	Type	Description
	MemoryOption.Byte	Byte unit (8 bits) (default)
	MemoryOption.Halfword	Half-word unit (16 bits) [V850]
	MemoryOption.Word	Word unit (78K: 16 bits, V850: 32 bits)

[Return value]

None

[Detailed description]

- This function sets the value at the address specified by *address*, according to *memoryOption*.

[Example of use]

```
>>>debugger.Memory.Read(0x100)
0x10
>>>debugger.Memory.Write(0x100, 0xFF)
>>>debugger.Memory.Read(0x100)
0xFF
>>> debugger.Memory.Write(0x100, MemoryOption.HalfWord)
>>>
```

debugger.Next

This function performs procedure step execution.

[Specification format]

```
debugger.Next(nextOption = NextOption.Source)
```

[Argument(s)]

Argument	Description	
<i>nextOption</i>	Specify the execution unit. The units that can be specified are shown below.	
	Type	Description
	NextOption.Source	Source-line unit (default)
	NextOption.Instruction	Instruction unit

[Return value]

None

[Detailed description]

- This function performs procedure step execution.
- If a function call is being performed, then stop after the function executes.

[Example of use]

```
>>>debugger.Next ()
>>>debugger.Next (NextOption.Instrunction)
>>>
```

debugger.Register.GetValue

This function references register/IO register/SFR.

[Specification format]

```
debugger.Register.GetValue(regName)
```

[Argument(s)]

Argument	Description
<i>regName</i>	Specify the register name to reference.

[Return value]

The referenced value

[Detailed description]

- This function displays the value of the register specified by "regName".

[Example of use]

```
>>>debugger.Register.GetValue("pc")
0x100
>>>debugger.Register.GetValue("p01:RB1")
0x20
>>>debugger.Register.SetValue("pc", 0x200)
>>>debugger.Register.GetValue("pc")
0x200
>>>
```


debugger.Register.SetValue

This function sets the value of a register/IO register/SFR.

[Specification format]

```
debugger.Register.SetValue(regName, value)
```

[Argument(s)]

Argument	Description
<i>regName</i>	Specify the register name to set.
<i>value</i>	Specify the value to set.

[Return value]

None

[Detailed description]

- This function sets the value specified by *value* in the register specified by *regName*.

[Example of use]

```
>>>debugger.Register.GetValue("pc")
0x100
>>>debugger.Register.GetValue("p01:RB1")
0x20
>>>debugger.Register.SetValue("pc", 0x200)
>>>debugger.Register.GetValue("pc")
0x200
>>>
```

debugger.Reset

This function resets the CPU.

[Specification format]

```
debugger.Reset ()
```

[Argument(s)]

None

[Return value]

None

[Detailed description]

- This function resets the CPU.

[Example of use]

```
>>>debugger.Reset ()  
>>>
```

debugger.ReturnOut

This function runs until control returns to the program that called the current function.

[Specification format]

```
debugger.ReturnOut ()
```

[Argument(s)]

None

[Return value]

None

[Detailed description]

- This function runs until control returns to the program that called the current function.

[Example of use]

```
>>>debugger.ReturnOut ()  
>>>
```

debugger.Run

This function resets and then run the program.

[Specification format]

```
debugger.Run(runOption = RunOption.Normal)
```

[Argument(s)]

Argument	Description	
<i>runOption</i>	Specify an option. The options that can be specified are shown below.	
	Type	Description
	RunOption.WaitBreak	Wait until program stops.
	RunOption.Normal	Breakpoints enabled; do not wait until program stops (default).

[Return value]

None

[Detailed description]

- This function resets and then run the program.
- If "RunOption.WaitBreak" is specified in *runOption*, then it will wait until the program stops.

[Example of use]

```
>>>debugger.Run()
>>>debugger.Run(RunOption.WaitBreak)
```

debugger.Step

This function performs step execution.

[Specification format]

```
debugger.Step(stepOption = StepOption.Source)
```

[Argument(s)]

Argument	Description	
<i>stepOption</i>	Specify the execution unit. The units that can be specified are shown below.	
	Type	Description
	StepOption.Source	Source-line unit (default)
	StepOption.Instruction	Instruction unit

[Return value]

None

[Detailed description]

- This function performs step execution.
- If a function call is being performed, then stop at the top of the function.

[Example of use]

```
>>>debugger.Step()
>>>debugger.Step(StepOption.Instrunction)
```

debugger.Stop

This function stops the execution of the debug tool.

[Specification format]

```
debugger.Stop()
```

[Argument(s)]

None

[Return value]

None

[Detailed description]

- This function stops the execution of the debug tool.
Forcibly halt the program.

[Example of use]

```
>>>debugger.Stop()  
>>>
```

debugger.Upload.Binary

This function saves the memory data in binary format.

[Specification format]

```
debugger.Upload.Binary(fileName, address1, address2, force = False)
```

[Argument(s)]

Argument	Description
<i>fileName</i>	Specify a file name.
<i>address1</i>	Specify an upload start address.
<i>address2</i>	Specify an upload end address.
<i>force</i>	Specify whether to overwrite. True: Overwrite False: Do not overwrite (default).

[Return value]

None

[Detailed description]

- This function saves the memory data from *address1* to *address2* in binary format.

[Example of use]

```
>>>debugger.Upload.Binary("testBinary.bin", 0x1000, 0x2000", True)
>>>
```

debugger.Upload.Coverage

This function saves the coverage data.

[Specification format]

```
debugger.Upload.Coverage(fileName, force = False)
```

[Argument(s)]

Argument	Description
<i>fileName</i>	Specify a file name.
<i>force</i>	Specify whether to overwrite. True: Overwrite False: Do not overwrite (default).

[Return value]

None

[Detailed description]

- This function saves the coverage data to a file.

[Example of use]

```
>>>debugger.Upload.Coverage("coverageData.csrcv")  
>>>
```


debugger.Upload.Intel

This function saves the memory data in Intel format.

[Specification format]

```
debugger.Upload.Intel(fileName, address1, address2, force = False)
```

[Argument(s)]

Argument	Description
<i>fileName</i>	Specify a file name.
<i>address1</i>	Specify an upload start address.
<i>address2</i>	Specify an upload end address.
<i>force</i>	Specify whether to overwrite. True: Overwrite False: Do not overwrite (default).

[Return value]

None

[Detailed description]

- This function saves the memory data from *address1* to *address2* in Intel format.

[Example of use]

```
>>>debugger.Upload.Intel("testIntel.hex ", 0x1000, 0x2000, True)
>>>
```

debugger.Upload.IntelIdTag

This function saves the memory data in ID-tagged Intel format.

[Specification format]

```
debugger.Upload.IntelIdTag(fileName, address1, address2, force = False)
```

[Argument(s)]

Argument	Description
<i>fileName</i>	Specify a file name.
<i>address1</i>	Specify an upload start address.
<i>address2</i>	Specify an upload end address.
<i>force</i>	Specify whether to overwrite. True: Overwrite False: Do not overwrite (default).

[Return value]

None

[Detailed description]

- This function saves the memory data from *address1* to *address2* in ID-tagged Intel format.

[Example of use]

```
>>>debugger.Upload.IntelIdTag("testIdTagIntel.hex", 0x1000, 0x2000", True)
>>>
```

debugger.Upload.Motorola

This function saves the memory data in Motorola format.

[Specification format]

```
debugger.Upload.Motorola(fileName, address1, address2, force = False)
```

[Argument(s)]

Argument	Description
<i>fileName</i>	Specify a file name.
<i>address1</i>	Specify an upload start address.
<i>address2</i>	Specify an upload end address.
<i>force</i>	Specify whether to overwrite. True: Overwrite False: Do not overwrite (default).

[Return value]

None

[Detailed description]

- This function saves the memory data from *address1* to *address2* in Motorola format.

[Example of use]

```
>>>debugger.Upload.Motorola("testMotorola.hex", 0x1000, 0x2000", True)
>>>
```

debugger.Upload.MotorolaIdTag

This function saves the memory data in ID-tagged Motorola format.

[Specification format]

```
debugger.Upload.MotorolaIdTag(fileName, address1, address2, force = False)
```

[Argument(s)]

Argument	Description
<i>fileName</i>	Specify a file name.
<i>address1</i>	Specify an upload start address.
<i>address2</i>	Specify an upload end address.
<i>force</i>	Specify whether to overwrite. True: Overwrite False: Do not overwrite (default).

[Return value]

None

[Detailed description]

- This function saves the memory data from *address1* to *address2* in ID-tagged Motorola format.

[Example of use]

```
>>>debugger.Upload.MotorolaIdTag("testIdTagMotorola.hex", 0x1000, 0x2000", True)
>>>
```

debugger.Upload.Tektronix

This function saves the memory data in Techtronics format.

[Specification format]

```
debugger.Upload.Tektronix(fileName, address1, address2, force = False)
```

[Argument(s)]

Argument	Description
<i>fileName</i>	Specify a file name.
<i>address1</i>	Specify an upload start address.
<i>address2</i>	Specify an upload end address.
<i>force</i>	Specify whether to overwrite. True: Overwrite False: Do not overwrite (default).

[Return value]

None

[Detailed description]

- This function saves the memory data from *address1* to *address2* in Techtronics format.

[Example of use]

```
>>>debugger.Upload.Tektronix("testTektronix.hex", 0x1000, 0x2000", True)
>>>
```

debugger.Upload.TektronixIdTag

This function saves the memory data in ID-tagged Techtronics format.

[Specification format]

```
debugger.Upload.TektronixIdTag(fileName, address1, address2, force = False)
```

[Argument(s)]

Argument	Description
<i>fileName</i>	Specify a file name.
<i>address1</i>	Specify an upload start address.
<i>address2</i>	Specify an upload end address.
<i>force</i>	Specify whether to overwrite. True: Overwrite False: Do not overwrite (default).

[Return value]

None

[Detailed description]

- This function saves the memory data from *address1* to *address2* in ID-tagged Techtronics format.

[Example of use]

```
>>>debugger.Upload.TektronixIdTag("testIdTagTektronix.hex", 0x1000, 0x2000", True)
>>>
```

debugger.Watch.GetValue

This function references a variable value.

[Specification format]

```
debugger.Watch.GetValue(variableName, encode = Encoding.Default, watchOption = WatchOption.Auto)
```

[Argument(s)]

Argument	Description	
<i>variableName</i>	Specify the variable name, register name, or I/O register name/SFR register name to reference.	
<i>encode</i>	Specify the encoding to use when displaying strings. By default, the system encoding is used. The encoding name conforms to the .NET specifications. Examples: Encoding.utf-8, Encoding.euc-jp	
<i>watchOption</i>	Specify an option. The options that can be specified are shown below.	
	Type	Description
	WatchOption.Auto	Automatically detect when displaying (default).
	WatchOption.Binary	Display in binary format.
	WatchOption.Octal	Display in octal format.
	WatchOption.Decimal	Display in decimal format.
	WatchOption.Hexdecimal	Display in hexadecimal format.
	WatchOption.String	Display as a string.
	WatchOption.Sizeof	Display the variable size in decimal format.

[Return value]

None

[Detailed description]

- This function displays the value of the variable specified by *variableName*.
- If *encode* is specified, then perform encoding using *encode*.
- If *watchOption* is specified, display according to *watchOption*.

[Example of use]

```
>>>debugger.Watch.GetValue("testVal")
128
>>>debugger.Watch.GetValue("testVal", WatchOption.Hexdecimal)
0x80
>>>debugger.Watch.GetValue("testVal", WatchOption.Binary)
0b10000000
```


debugger.Watch.SetValue

This function sets a variable value.

[Specification format]

```
debugger.Watch.SetValue(variableName, value)
```

[Argument(s)]

Argument	Description
<i>variableName</i>	Specify the variable name, register name, or I/O register name/SFR register name to set.
<i>value</i>	Specify the value to set.

[Return value]

None

[Detailed description]

- This function sets the value specified by *value* in the variable, register, or I/O register/SFR register specified by *variableName*.

[Example of use]

```
>>>debugger.Watch.GetValue("testVal")
128
>>>debugger.Watch.GetValue("testVal", WatchOption.Hexdecimal)
0x80
>>>debugger.Watch.GetValue("testVal", WatchOption.Binary)
0b10000000
>>>debugger.Watch.SetValue("testVal", 100)
>>>debugger.Watch.GetValue("testVal")
100
>>>debugger.Watch.GetValue("testVal", WatchOption.Hexdecimal)
0x64
>>>debugger.Watch.GetValue("testVal", WatchOption.Binary)
0b1100100
>>>debugger.Watch.SetValue("testVal", 0x256)
>>>debugger.Watch.GetValue("testVal", WatchOption.Hexdecimal)
0x256
```

debugger.Where

This function displays a stack backtrace.

[Specification format]

```
debugger.Where()
```

[Argument(s)]

None

[Return value]

None

[Detailed description]

- This function displays a stack backtrace.

Caution If "--- Information below might be inaccurate." is displayed, then the information displayed below may not be reliable. [78K0R]

[Example of use]

```
>>>debugger.Where()  
1: test2.c#sub2(int i)#13  
--- Information below might be inaccurate.  
2:func.c# func(register int _i)#34  
>>>
```

debugger.Whereami

This function displays a location.

[Specification format]

```
debugger.Whereami (address)
```

[Argument(s)]

Argument	Description
<i>address</i>	Specify the address of the location to display.

[Return value]

None

[Detailed description]

- This function displays the location at the address specified by *address*.
- The location is normally displayed in the following format.

```
file-name#function-name at file-name#line-number
```

However, if the function or line number at that address is not found, then the location is displayed in the following format.

```
at symbol-name+offset-value
```

If the symbol is not found, then the location is displayed in the following format.

```
at address-value
```

- If *address* is omitted, then the location of the pc value is displayed.

[Example of use]

```
>>>debugger.Whereami()
foo.c#func at foo.c#100
>>>debugger.Whereami(0x100)
foo.c#main at foo.c#20
>>>
```

debugger.XCoverage.Clear

This function clears the coverage memory.

[Specification format]

```
debugger.XCoverage.Clear()
```

[Argument(s)]

None

[Return value]

None

[Detailed description]

- This function clears the coverage memory.

[Example of use]

```
>>>debugger.XCoverageClear()  
>>>
```

debugger.XCoverage.GetCoverage

This function gets the coverage.

[Specification format]

```
debugger.XCoverage.GetCoverage(funcName, progName = "", fileName = "")
```

[Argument(s)]

Argument	Description
<i>funcName</i>	Specify the function name to retrieve coverage for.
<i>progName</i>	Specify the name of the load module containing the function. If there is only one load module, then this can be omitted (default).
<i>fileName</i>	Specify the name of the file containing the function. If it is a global function, then this can be omitted (default).

Caution If two or more parameters are specified, then all three parameters must be specified.

[Return value]

Number without "%"

Remark The results of function execution are displayed with a "%" sign added.

[Detailed description]

- This function gets coverage for the function specified by *funcName*.
- If there are multiple load modules, specify *progName*.
- In the case of a static function, specify *fileName*.

[Example of use]

```
>>>debugger.XCoverage.GetCoverage("TestInit", "Test.Out", "Test.c")
81.50%
>>>
```

debugger.XRunBreak.Delete

This function deletes XRunBreak setting information.

[Specification format]

```
debugger.XRunBreak.Delete()
```

[Argument(s)]

None

[Return value]

None

[Detailed description]

- This function deletes XRunBreak setting information.

[Example of use]

```
>>>debugger.XRunBreak.Refer()  
None  
>>>debugger.XRunBreak.Set(1, TimeType.S, True)  
>>>debugger.XRunBreak.Refer()  
1Second Periodic  
>>>debugger.XRunBreak.Delete()  
>>>debugger.XRunBreak.Refer()  
None
```

debugger.XRunBreak.Refer

This function displays XRunBreak setting information.

[Specification format]

```
debugger.XRunBreak.Refer()
```

[Argument(s)]

None

[Return value]

None

[Detailed description]

- This function displays the period information (period time [Periodic]) of the set XRunBreak.
- If there is no XRunBreak setting, "None" is displayed.

[Example of use]

```
>>>debugger.XRunBreak.Refer()  
None  
>>>debugger.XRunBreak.Set(1, TimeType.S, True)  
>>>debugger.XRunBreak.Refer()  
1Second Periodic
```

debugger.XRunBreak.Set

This function configures XRunBreak settings.

[Specification format]

```
debugger.XRunBreak.Set(time, timeType = TimeType.Ms, periodic = False)
```

[Argument(s)]

Argument	Description	
<i>time</i>	Specify the break time.	
<i>timeType</i>	Specify the break time unit. The units that can be specified are shown below.	
	Type	Description
	TimeType.Min	Minute unit
	TimeType.S	Second unit
	TimeType.Ms	Millisecond unit (default)
	TimeType.Us	Microsecond unit
TimeType.Ns	Nanosecond unit	
<i>periodic</i>	Specify whether to call the callback every time the specified time elapses. True: Call at every specified time interval. False: Call one time only (default).	

[Return value]

None

[Detailed description]

- This function configures XRunBreak settings.
- The XRunBreak calling interval depends on the simulator.

[Example of use]

```
>>>debugger.XRunBreak.Refer()
None
>>>debugger.XRunBreak.Set(1, TimeType.S, True)
>>>debugger.XRunBreak.Refer()
1Second Periodic
```


debugger.XTime

This function displays timing information between Go and Break.

[Specification format]

```
debugger.XTime()
```

[Argument(s)]

None

[Return value]

None

[Detailed description]

- This function displays timing information between Go and Break in nanoseconds.

[Example of use]

```
>>>debugger.XTime()  
9820214200nsec  
>>>
```

debugger.XTrace.Clear

This function clears the trace memory.

[Specification format]

```
debugger.XTrace.Clear()
```

[Argument(s)]

None

[Return value]

None

[Detailed description]

- This function clears the trace memory.

[Example of use]

```
>>>debugger.XTrace.Clear()  
>>>
```

debugger.XTrace.Dump

This function dumps the trace data.

[Specification format]

```
debugger.XTrace.Dump(frameCount, fileName = "", append = False)
```

[Argument(s)]

Argument	Description
<i>frameCount</i>	Specify the number of dumps.
<i>fileName</i>	Specify the name of the file to dump to (default: not specified).
<i>append</i>	Specify whether to append trace data to the file. True: Append trace data to the file. False: Do not append trace data to the file (default).

[Return value]

None

[Detailed description]

- This function dumps trace data for the number of frames specified by *frameCount*.
- If *fileName* is specified, then the trace data is written to the file.
- If *append* is set to "True", then the trace data is appended to the file.

[Example of use]

```
>>>debugger.XTrace.Dump(3)
004086
004087  4 jarl _XO_Test_Init, lp
004088  2 br_XO_Test_Init+0x9c
>>>debugger.XTrace.Dump(10, "TestTrace.txt")
>>>
```

BreakCondition

This class creates a break condition.

[Specification format]

```
class BreakCondition:
    Address = ""
    Data = None
    AccessSize = None
    BreakType = BreakType.Hardware
```

[Argument(s)]

Argument	Description	
Address	Specify the address at which to set a break. Must be specified.	
Data	Specify the number to set as a break condition for the data. If "None" is specified, then the data condition is ignored.	
AccessSize	Specify the access size (8, 16, or 32). If "None" is specified, then all access sizes will be specified.	
BreakType	Specify the break type. The break types that can be specified are shown below.	
	Type	Description
	BreakType.Software	Software break (except a simulator)
	BreakType.Hardware	Hardware break (default)
	BreakType.Read	Data read break
	BreakType.Write	Data write break
	BreakType.Access	Data access break

[Detailed description]

- "BreakCondition" is in class format, and the break condition is set in the variable.
- In order to create a break condition, create an instance, and set conditions for that instance.

[Example of use]

```
>>>executeBreak = BreakCondition()           ... create instance
>>>executeBreak.Address = "main"
>>>executeBreak.BreakType = BreakType.Software
>>>debugger.Breakpoint.Set(executeBreak)    ... Specify function in which to set the break
point in parameter
>>>
>>>dataBreak = BreakCondition()             ... create instance
>>>dataBreak.Address = "chData"
>>>dataBreak.Data = 0x10
>>>dataBreak.BreakType = BreakType.Access
>>>debugger.Breakpoint.Set(dataBreak)       ... Specify function in which to set the break
point in parameter
>>>
>>>executeBreak.Address = "sub + 0x10"      ... Reuse break condition
>>>debugger.Breakpoint.Set(executeBreak)    ... Specify function in which to set the break
point in parameter
>>>
```

debugger.Download.CpuReset
debugger.Download.FlashErase

This property sets or reference the download property of the debug tool.

[Specification format]

```
debugger.Download.CpuReset = reset
debugger.Download.FlashErase = erase
```

[Argument(s)]

Argument	Description
<i>reset</i>	Specify the CPU reset setting during download in a bool type. Set or reference the debugging tool's [CPU Reset after download] property. True: Set the property to [Yes]. False: Set the property to [No].
<i>erase</i>	Specify the erase Flash memory setting during download in a bool type. Set or reference the debugging tool's [Erase flash ROM before download] property. True: Set the property to [Yes]. False: Set the property to [No].

[Return value]

If the property is set to [Yes]: True
 If the property is set to [No]: False

Caution If a PM+ workspace is converted to a CubeSuite project, then there will be no debugging tool in the main project. For this reason, "None" will be returned if the main project is the active project.

[Detailed description]

- This property sets or reference the download property of the debug tool.

[Example of use]

```
>>>print debugger.Downlaod.CpuReset
False
>>>debugger.Download.CpuReset = True
>>>print debugger.Download.CpuReset
True
>>>print debugger.Downlaod.FlashErase
False
>>>
```

debugger.Memory.NoVerify

This property switches the write-time verification setting.

[Specification format]

```
debugger.Memory.NoVerify = noverify
```

[Argument(s)]

Argument	Description
<i>noverify</i>	Specify whether to verify during writes. True: Verify during writes. False: Do not verify during writes.

[Return value]

The set value

Caution If a PM+ workspace is converted to a CubeSuite project, then there will be no debugging tool in the main project. For this reason, "None" will be returned if the main project is the active project.

[Detailed description]

- This property switches the write-time verification setting.

[Example of use]

```
>>>print debugger.Memory.NoVerify
False
>>>debugger. Memory.NoVerify = True
>>>print debugger. Memory.NoVerify
True
>>>
```

debugger.Option.Coverage
debugger.Option.OpenBreak
debugger.Option.Timer
debugger.Option.Trace
debugger.Option.UseTraceData

This property sets or reference the options of the debug tool.

[Specification format]

```

debugger.Option.Coverage = coverage
debugger.Option.OpenBreak = openBreak
debugger.Option.Timer = timer
debugger.Option.Trace = trace
debugger.Option.UseTraceData = useTraceDataType
    
```

[Argument(s)]

Argument	Description								
<i>coverage</i>	Specify whether to use the coverage function. True: Use the coverage function. False: Do not use the coverage function.								
<i>openBreak</i>	Specify whether to use the open break function. True: Use the open break function. False: Do not use the open break function.								
<i>timer</i>	Specify whether to use the timer function. True: Use the timer function. False: Do not use the timer function.								
<i>trace</i>	Specify whether to use the trace function. True: Use the trace function. False: Do not use the trace function.								
<i>useTraceDataType</i>	Specify which function to use the trace data in. [V850 [IECUBE]] The functions that can be specified are shown below.								
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Type</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>UseTraceDataType.RRM</td> <td>RRM function</td> </tr> <tr> <td>UseTraceDataType.Trace</td> <td>Trace function</td> </tr> <tr> <td>UseTraceDataType.Coverage</td> <td>Coverage function</td> </tr> </tbody> </table>	Type	Description	UseTraceDataType.RRM	RRM function	UseTraceDataType.Trace	Trace function	UseTraceDataType.Coverage	Coverage function
Type	Description								
UseTraceDataType.RRM	RRM function								
UseTraceDataType.Trace	Trace function								
UseTraceDataType.Coverage	Coverage function								

[Return value]

The set value

Caution If a PM+ workspace is converted to a CubeSuite project, then there will be no debugging tool in the main project. For this reason, "None" will be returned if the main project is the active project.

[Detailed description]

- This property sets or reference the options of the debug tool.

[Example of use]

```
>>>print debugger.Option.UseTraceData
Trace
>>>debugger.Option.UseTraceData = UseTraceDataType.Coverage
>>>print debugger.Option.Coverage
False
>>>debugger.Option.Coverage = True
>>>print debugger.Option.Coverage
True
>>>
```

```

debugger.XTrace.Addup
debugger.XTrace.Complement
debugger.XTrace.Mode

```

This property sets or reference the tracing options of the debug tool.

[Specification format]

```

debugger.XTrace.Addup = addup
debugger.XTrace.Complement = complement
debugger.XTrace.Mode = traceMode

```

[Argument(s)]

Argument	Description	
<i>addup</i>	Specify whether to add up times/tags. True: Add up times/tags. False: Do not add up times/tags.	
<i>complement</i>	Specify whether to supplement the trace. True: Supplement the trace. False: Do not supplement the trace.	
<i>traceMode</i>	Specify the trace control mode. The trace control modes that can be specified are shown below.	
	Type	Description
	TraceMode.FullBreak	Stop program execution and writing of trace data after all trace data has been used up.
	TraceMode.FullStop	Stop writing trace data after all trace data has been used up.
	TraceMode.NonStop	Continue writing trace data even if all trace data has been used up.

[Return value]

The set value

Caution If a PM+ workspace is converted to a CubeSuite project, then there will be no debugging tool in the main project. For this reason, "None" will be returned if the main project is the active project.

[Detailed description]

- This property sets or reference the tracing options of the debug tool.

[Example of use]

```
>>>print debugger.XTrace.Addup
False
>>>debugger.XTrace.Addup = True
>>>print debugger.XTrace.Addup
True
>>>
```

APPENDIX G INDEX

A

- Add Existing Subproject dialog box ... 217
- Add File dialog box ... 141
- Add Folder and File dialog box ... 143

B

- Batch build ... 259
- Batch Build dialog box ... 150
- Browse For Folder dialog box ... 219
- Build ... 254
- Build mode ... 250
- Build Mode Settings dialog box ... 148
- [Build Options] tab ... 132

C

- change the project name ... 51
- Character String Input dialog box ... 145
- Checking for Updates dialog box ... 87
- Clean ... 256
- [Commands] tab ... 197
- Contact Information for Technical Support dialog box ...
209
- Create Project dialog box ... 106
- CubeSuite License Manager window ... 92
- CubeSuite Python functions ... 261
- CubeSuite Uninstaller window ... 234
- CubeSuite Update Manager window ... 75
 - Download and install page ... 79
 - Download only page ... 81
 - Error page ... 86
 - Finish page ... 83
 - History page ... 84
 - Restore hidden updates page ... 85
 - Select updates page ... 77
- Customize Keyboard dialog box ... 202

D

- Debug-dedicated project ... 241

- Download and install page ... 79

- Download only page ... 81

E

- Editor panel ... 134
- Error page ... 86

F

- favorites menu ... 50
- Finish page ... 83

G

- [General - Build/Debug] category ... 187
- [General - Display] category ... 173
- [General - External Tools] category ... 183
- [General - Font and Color] category ... 178
- [General - Startup and Exit] category ... 171
- [General - Text Editor] category ... 175
- [General - Update] category ... 189
- Go to the Location dialog box ... 152

H

- History page ... 84

M

- Main window ... 94
- Make settings for build operations ... 248
- [Microcontroller Information] tab ... 130

N

- New Toolbar dialog box ... 199

O

- One Point Advice dialog box ... 212
- Open Project dialog box ... 215
- Option dialog box ... 169
 - [General - Build/Debug] category ... 187
 - [General - Display] category ... 173
 - [General - External Tools] category ... 183

- [General - Font and Color] category ... 178
- [General - Startup and Exit] category ... 171
- [General - Text Editor] category ... 175
- [General - Update] category ... 189
- [Other - User Information] category ... 191
- Other &Windows... dialog box ... 214
- [Other - User Information] category ... 191
- Output panel ... 138

- P**
- PM+ project ... 57
- Progress Status dialog box ... 168
- [Project] tab ... 127
- project ... 39
- Project Tree panel ... 113
- Property panel ... 123
 - [Build Options] tab ... 132
 - [Microcontroller Information] tab ... 130
 - [Project] tab ... 127
 - [Subproject] tab ... 128
- Python Console panel ... 232

- Q**
- [Quick Replace] tab ... 162
- [Quick Search] tab ... 156

- R**
- Rapid build ... 258
- Rearrange Commands dialog box ... 204
- Rebuild ... 256
- Rename Toolbar dialog box ... 201
- Restore hidden updates page ... 85

- S**
- Save As dialog box ... 224
- Save Project As dialog box ... 222
- Save Settings dialog box ... 153
- save the project file ... 60
- Search and Replace dialog box ... 155
 - [Quick Replace] tab ... 162
 - [Quick Search] tab ... 156
 - [Whole Replace] tab ... 165
 - [Whole Search] tab ... 159
- Select External Text Editor dialog box ... 230
- Select Program dialog box ... 228
- Select updates page ... 77
- Source Convert Setting dialog box ... 111
- Start panel ... 104
- [Subproject] tab ... 128
- subproject ... 41

- T**
- tag jump ... 139
- Task Tray ... 91
- Text Edit dialog box ... 146
- [Toolbars] tab ... 195

- U**
- Update in progress dialog box ... 88
- Update Manager Options dialog box ... 89
- User Setting dialog box ... 193
 - [Commands] tab ... 197
 - [Toolbars] tab ... 195

- V**
- Version Information dialog box ... 207

- W**
- [Whole Replace] tab ... 165
- [Whole Search] tab ... 159

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Oct 01, 2010	-	First Edition issued

CubeSuite Ver.1.40 User's Manual: Start

Publication Date: Rev.1.00 Oct 1, 2010

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: +44-1628-585-100, Fax: +44-1628-585-900**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.**7F, No. 363 Fu Shing North Road Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670**Renesas Electronics Singapore Pte. Ltd.**1 harbourFront Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001**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

CubeSuite Ver.1.40



Renesas Electronics Corporation

R20UT0256EJ0100