# CubeSuite+ V2.02.00

Integrated Development Environment

## User's Manual: Start

Target Device

78K0 Microcontroller

RL78 Family

78K0R Microcontroller

V850 Family

RX Family

RH850 Family

# How to Use This Manual

This manual describes the role of the CubeSuite+ integrated development environment for developing applications and systems for RH850 family, RX family, V850 family, RL78 family, 78K0R microcontrollers, 78K0 microcontrollers, and provides an outline of its features.

CubeSuite+ is an integrated development environment (IDE) for RH850 family, RX family, V850 family, RL78 family, 78K0R microcontrollers, 78K0 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   REGULAR EXPRESSIONS SYNTAX
APPENDIX F   USING AN EXTERNAL BUILD TOOL
APPENDIX G   Python CONSOLE/Python FUNCTIONS
APPENDIX H   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+<br>Integrated Development Environment<br>User's Manual | Start | This manual |
| | RX Design | R20UT2862E |
| | V850 Design | R20UT2134E |
| | RL78 Design | R20UT2684E |
| | 78K0R Design | R20UT2137E |
| | 78K0 Design | R20UT2138E |
| | RH850 Coding | R20UT2584E |
| | RX Coding | R20UT2470E |
| | V850 Coding | R20UT0553E |
| | Coding for CX Compiler | R20UT2659E |
| | RL78,78K0R Coding | R20UT2774E |
| | 78K0 Coding | R20UT2141E |
| | RH850 Build | R20UT2585E |
| | RX Build | R20UT2472E |
| | V850 Build | R20UT0557E |
| | Build for CX Compiler | R20UT2142E |
| | RL78,78K0R Build | R20UT2623E |
| | 78K0 Build | R20UT0783E |
| | RH850 Debug | R20UT2866E |
| | RX Debug | R20UT2875E |
| | V850 Debug | R20UT2446E |
| | RL78 Debug | R20UT2867E |
| | 78K0R Debug | R20UT0732E |
| | 78K0 Debug | R20UT0731E |
| | Analysis | R20UT2868E |
| | Message | R20UT2871E |

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

# TABLE OF CONTENTS

This chapter describes the role of the CubeSuite+ integrated development environment for developing applications and systems for the microcontrollers (RH850, RX, V850, R8C (Localised support), RL78, 78K0R, and 78K0), and provides an outline of its features.

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

## 1.1    Introduction

CubeSuite+ is an integrated development environment (IDE)[Note] for the microcontrollers.

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 preparation, design/implementation, 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 OS), 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 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.

**(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 an editor 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 F   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 information while the program is executing, and display information about the functions and variables.

**(7) 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**
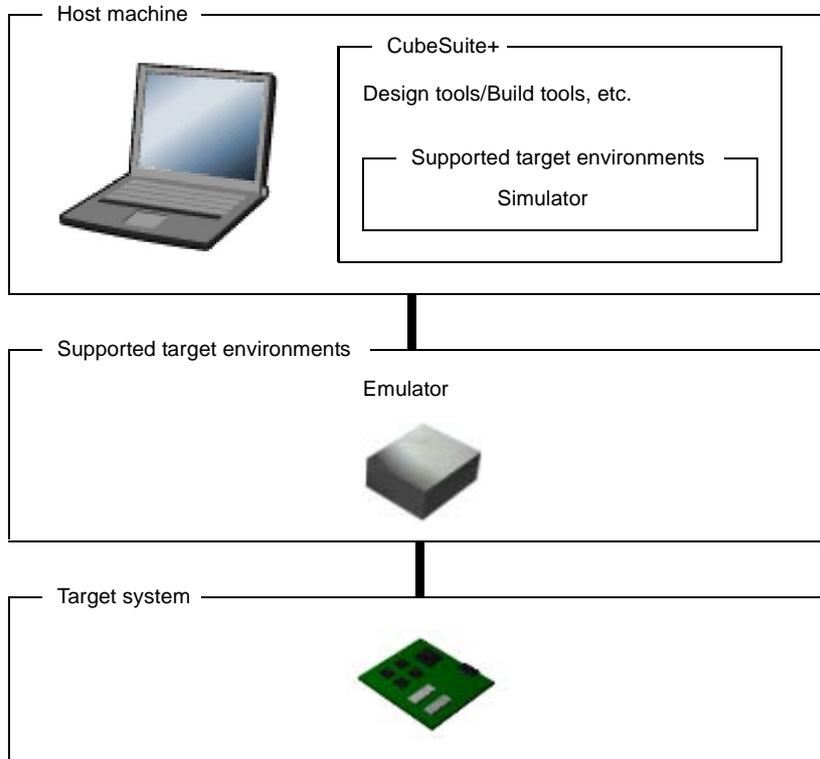


**Remark**    The emulator that can be connected differs depending on the microcontroller used in the project.
See "CubeSuite+ Integrated Development Environment User's Manual: Debug" for details.

## 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 1 GB (2 GB or higher for Windows (64-bit OS)), 2 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 Vista (32-bit OS, 64-bit OS)

- Windows 7 (32-bit OS, 64-bit OS)

- Windows 8 (32-bit OS, 64-bit OS)

- Windows 8.1 (32-bit OS, 64-bit OS)

- Microsoft .NET Framework 4

- Microsoft Visual C++ 2010 SP1 runtime libraries

- Internet Explorer 7 or higher

**(3)  Supported target environments**

- Emulator
    - IECUBE
    - IECUBE2
    - Full-spec emulator
    - MINICUBE
    - MINICUBE2
    - E1
    - E20
    - EZ Emulator
- 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 DVD into the drive.**
     The Preparing to Install page appears automatically.

**Figure 2-1.   Preparing to Install Page**
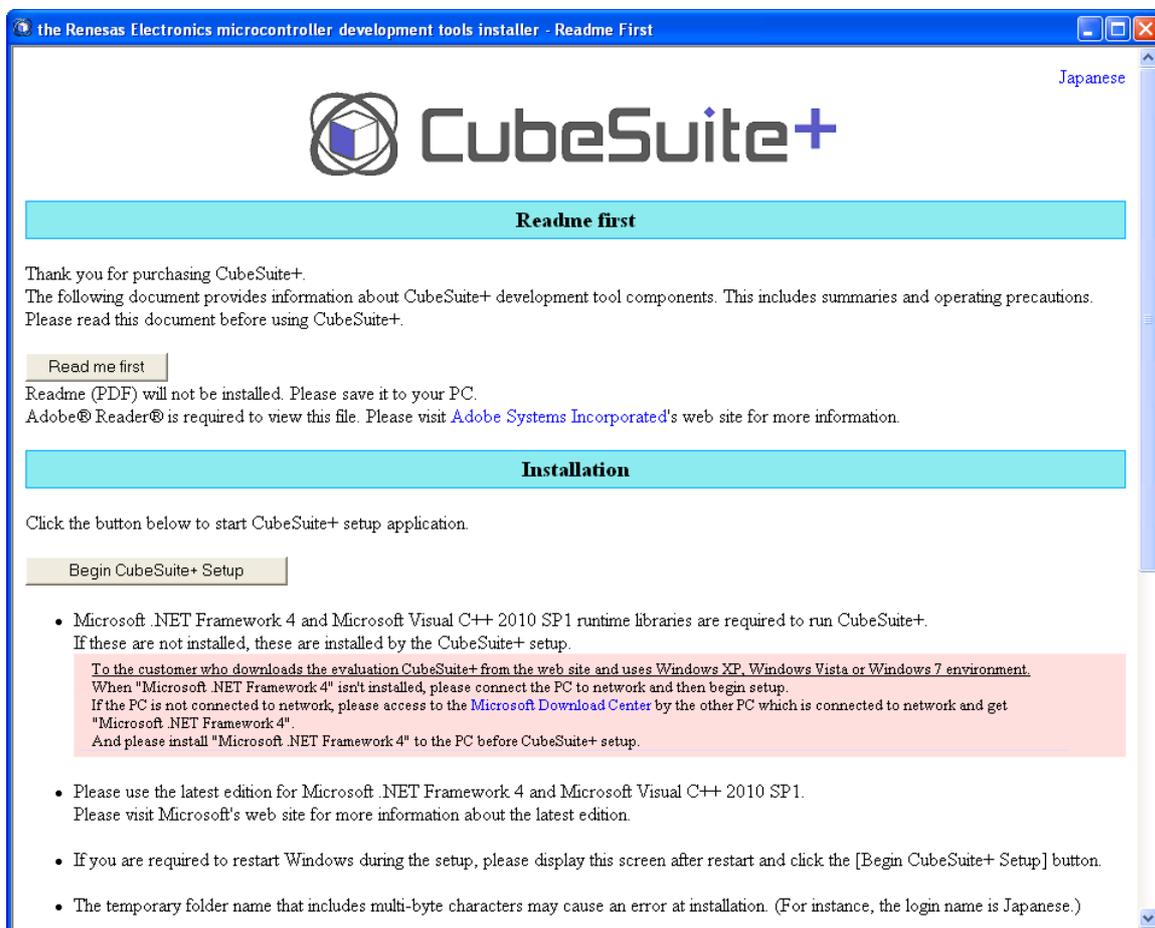


Click the [Begin CubeSuite+ Setup] button, CubeSuite+ setup begins.

     **Caution**     **CubeSuite+ setup must be conducted with administrator privileges.**

     **Remark**     If the page does not appear automatically, open "Install.hta" in the DVD.

**(2) Check the environment required for installation.**

The following must be installed to allow operation of this product; the window shows any that are not yet installed.
- Microsoft .NET Framework 4
- Microsoft Visual C++ 2010 SP1 runtime libraries

**Remark**     If all of the above are installed, this window does not appear.

**Figure 2-2.   Confirm Settings Window**
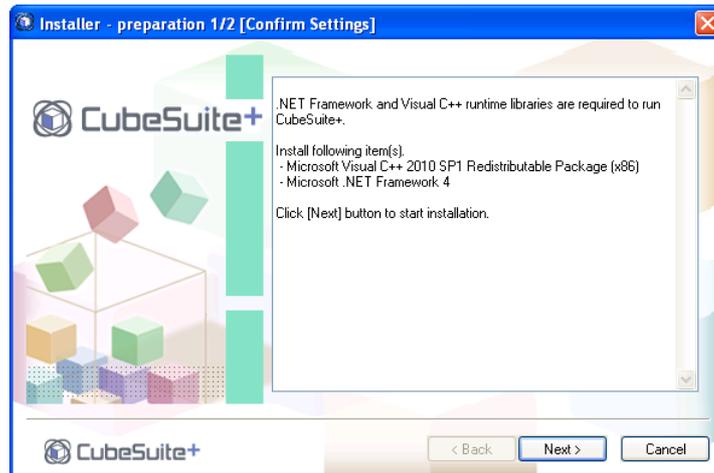


Check the information, then click the [Next] button.

**(3) Check the progress of installation of required software.**

Follow the installation wizard to install the software.

The installation progress appear in the [Install Status] area in this windows.

**Remark**     If the required software is already all installed, this window does not appear.

**Figure 2-3.   Install Window**
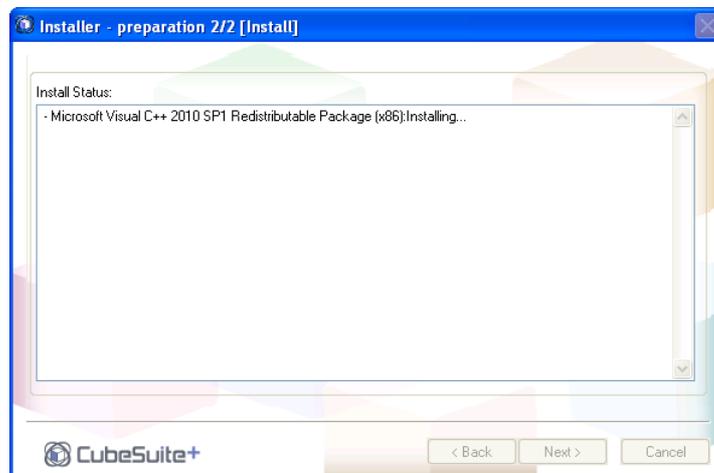


After all required software is installed, click the [Next] button.

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

**Remark**   If this software installation fails, or a restart is required, a message to this effect appears, and the instal-
lation cannot be continued (the [Next] button is disabled).
If a restart is required, restart Windows, then begin the setup again from the [Begin CubeSuite+ Setup]
button in the Preparing to Install page of step (1) above.

**(4)  Confirm the product information, etc.**
Product information, etc. appears.

**Figure 2-4.   Initial Window**



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

**(5)  Confirm the software license agreement.**
The installer asks if you agree to the SOFTWARE LICENSE AGREEMENT.

**Figure 2-5.   Software License Agreement Window**
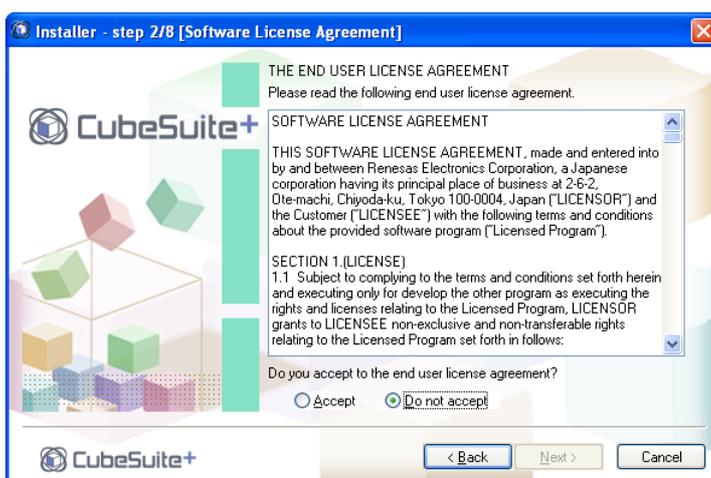


Check the information, and if you agree it, select "Accept", then click the [Next] button.

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

**(6) 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 2-6.  Development Tools Selection Window**
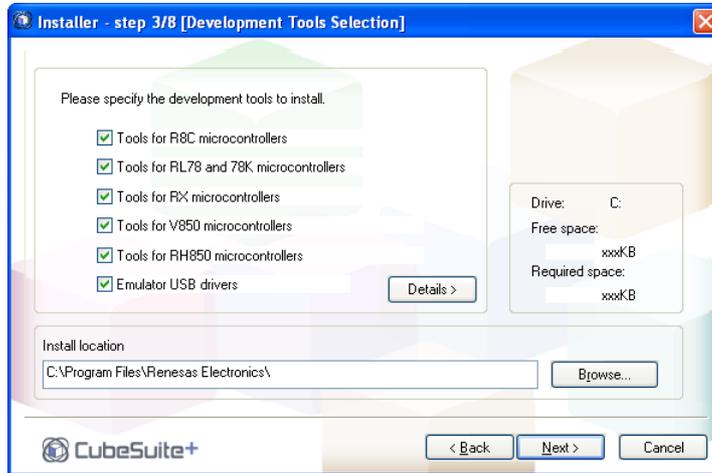


After you select the install target and installation location, click the [Next] button.

**Caution** **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 2-7.  Select Components Window**
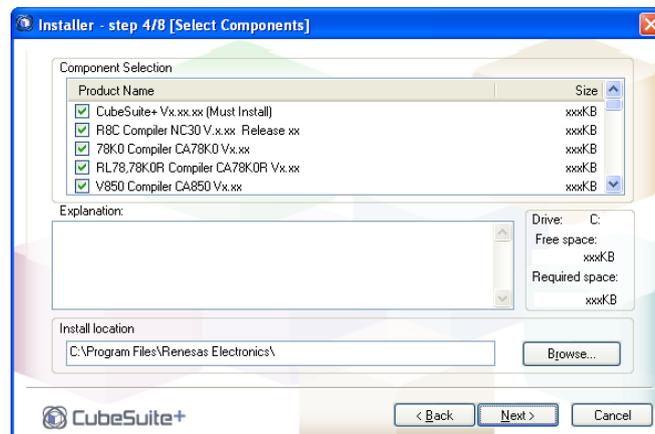


**Caution** **The check boxes of products that must be installed cannot be cleared.**

**(7) Enter your license key.**

Registering a license will remove the following limitations.

- Below are the maximum code sizes that can be developed.

RH850 family: 256 Kbytes

RX family**Note**: 128 Kbytes

V850 family: 256 Kbytes

R8C family (Localised support): 64 Kbytes

RL78 family: 64 Kbytes (The available RAM is restricted to the internal RAM.)

78K0R microcontroller: 64 Kbytes (The available RAM is restricted to the internal RAM.)

78K0 microcontroller: 64 Kbytes (The available RAM is restricted to the internal RAM.)

**Note**   There is a trial period for the first 60 days following building after the first installation of an evaluation version of a software tool.

**Remark**    A license can be registered in the CubeSuite+ Update Manager window after installing this product.

**Figure 2-8.   License Registration Window**



Click the [License Manager...] button.

**Figure 2-9.   License Manager Window**
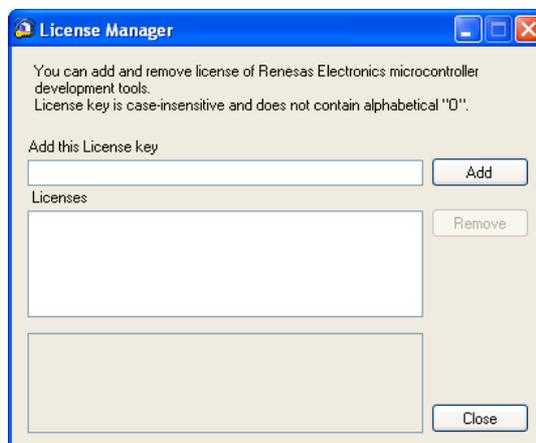
After you register the license in the License Manager window, click the [Next] button in the License Registration window.

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

**(8) Check the installation targets and installation.**
The settings made in the Development Tools Selection window or Select Components window are appeared.

**Figure 2-10. Installation Settings Confirmation Window**
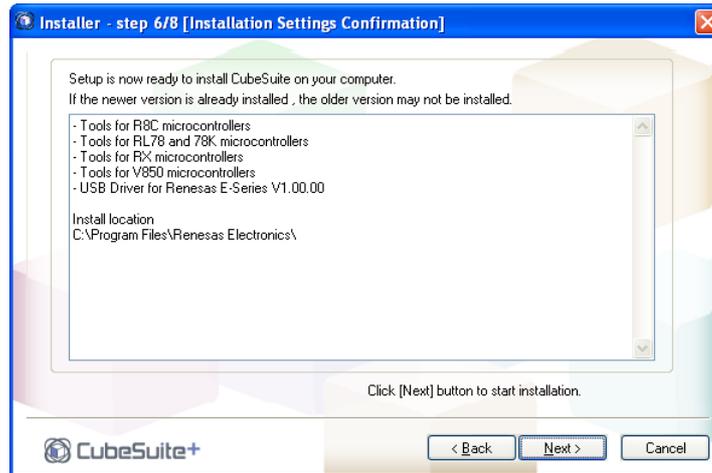


Check the information, and click the [Next] button.

**Caution**     **If an emulator USB driver is installed on Windows Vista, Windows 7, Windows 8, or Windows 8.1, the warning dialog box may appear.**

**(9) 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 2-11. Installation Execution Window**
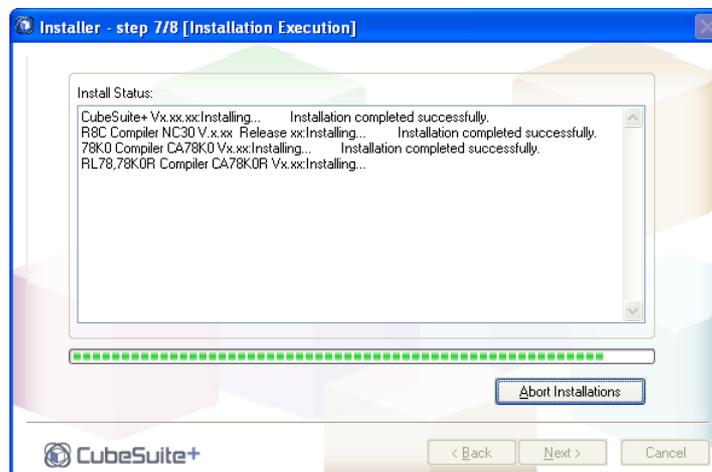


When all installations of the install targets are finished, the [Next] button becomes enabled.

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.

**(10) Confirm the results of the installation.**
When all setup-related tasks are finished, the results of the installation appear.

**Figure 2-12.   Setup Completion Window**
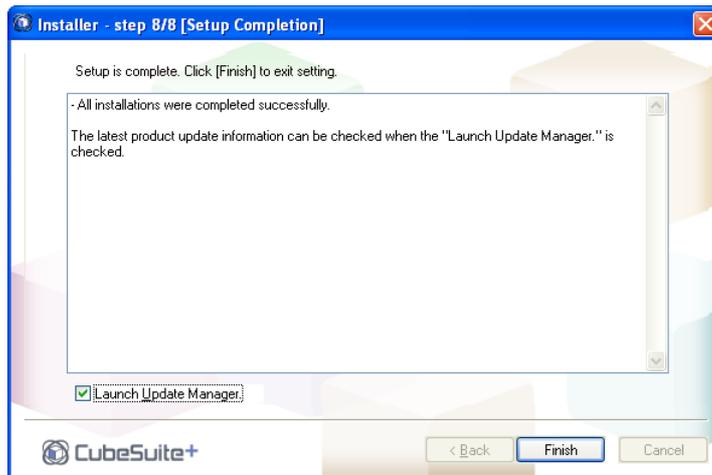


Click the [Finish] button to complete the installation.

**Remarks 1.**   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.**   If required software was installed in steps (2) and (3) above, a message appears, suggesting you update to the latest version.  If a restart is required, a message to this effect also appears.

## 2.2    Uninstalling CubeSuite+

You can uninstall CubeSuite+ using the CubeSuite+ Uninstaller.

**(1)  Launch CubeSuite+ Uninstaller.**
From the Windows the [Start] menu, select [Programs] >> [Renesas Electronics CubeSuite+] >> [Uninstaller].  The CubeSuite+ Uninstaller window is launched.

**Remark**    In Windows 8 or Windows 8.1, double-click on [Uninstaller] on the start screen.

**Figure 2-13.   CubeSuite+ Uninstaller Window**
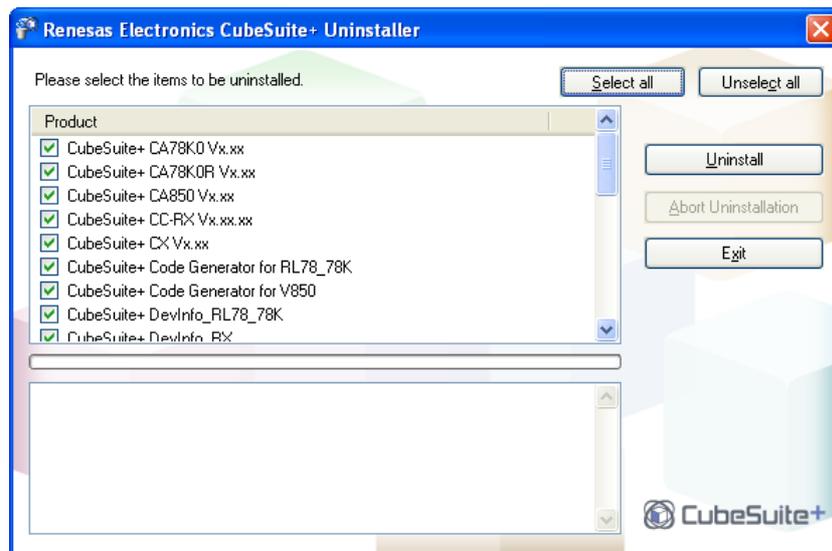


Click the [Select all] button to select all the check boxes.
Click the [Unselect all] button to clear all the check boxes.
Click the [Uninstall] button.

**(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 2-14.   Uninstallation Execution Window (Progress)**
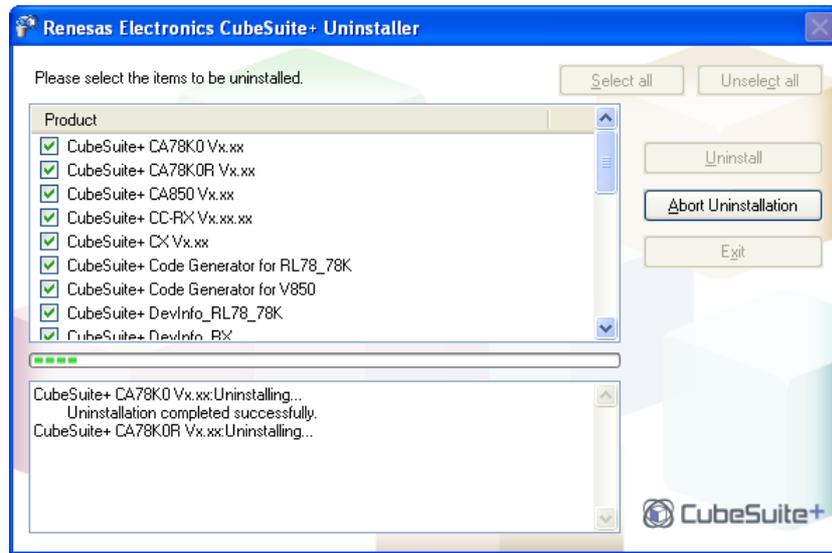


> **Remark**    If you click the [Abort Uninstallations] button, a message stating that the user chose to cancel the uninstallation appears in the progress details area.  The uninstallation will halt when the uninstallation of the tool currently being uninstalled finishes.

**(3)  Confirm the results of the uninstallation.**

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

**Figure 2-15.   Uninstallation Completion Window (Result)**
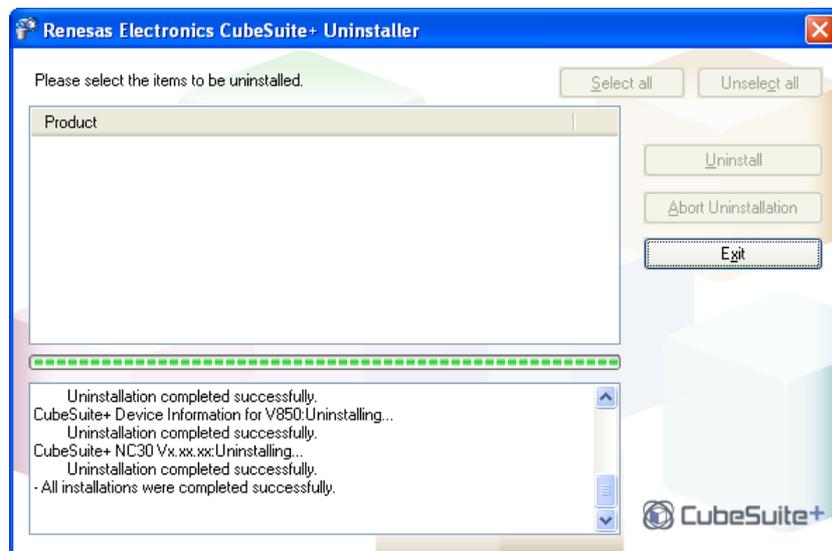


Click the [Exit] button to complete the uninstallation.

> **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 License Manager window using the method below.

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

**Remark**    In Windows 8 or Windows 8.1, double-click on [License Manager] on the start screen.

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

- Below are the maximum code sizes that can be developed.
    - RH850 family: 256 Kbytes
    - RX family**Note**: 128 Kbytes
    - V850 family: 256 Kbytes
    - R8C family (Localised support): 64 Kbytes
    - RL78 family: 64 Kbytes (The available RAM is restricted to the internal RAM.)
    - 78K0R microcontroller: 64 Kbytes (The available RAM is restricted to the internal RAM.)
    - 78K0 microcontroller: 64 Kbytes (The available RAM is restricted to the internal RAM.)

**Note**   There is a trial period for the first 60 days following building after the first installation of an evaluation version of a software tool.

### 2.3.1    Show license information

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

**Figure 2-16.   License Manager Window**

**2.3.2    Add a license**

In the 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 2-17.   License Manager Window**
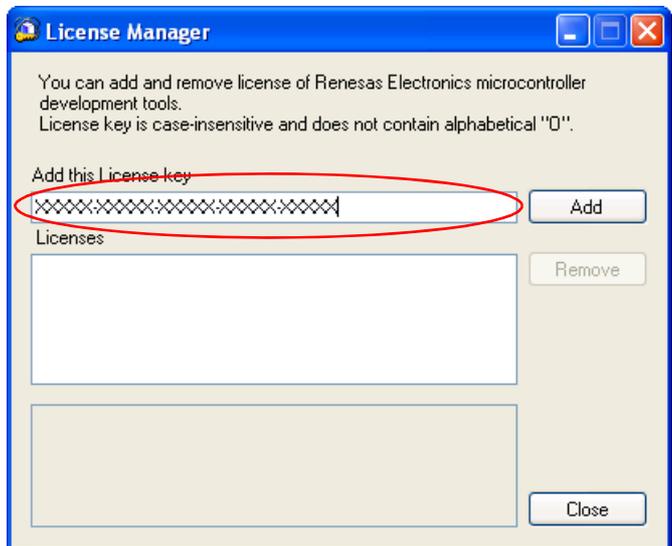


**2.3.3    Delete a license**

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

**Figure 2-18.   License Manager Window**
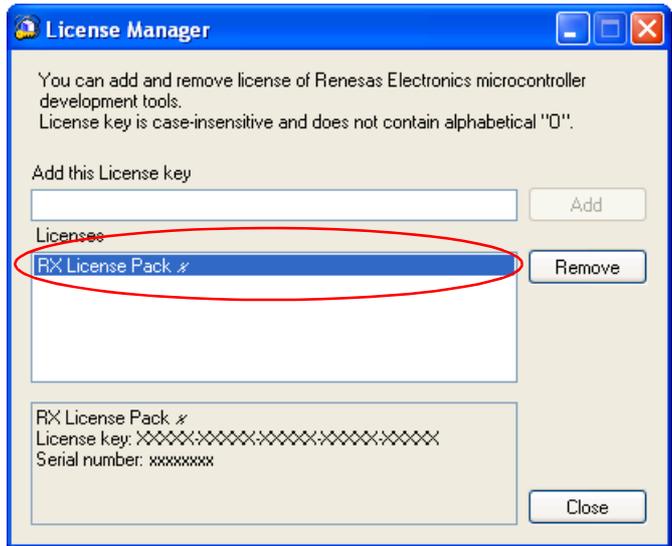
## 2.4   Using the Update Feature

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.

**Caution   When you use the update feature, the host machine must have a connection to the Internet.**

### 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] >> [Renesas Electronics CubeSuite+] >> [Update Manager], or from the CubeSuite+ menu, select [Help] >> [Check for Updates...].  The CubeSuite+ Update Manager window launches.

**Figure 2-19.   CubeSuite+ Update Manager Window**
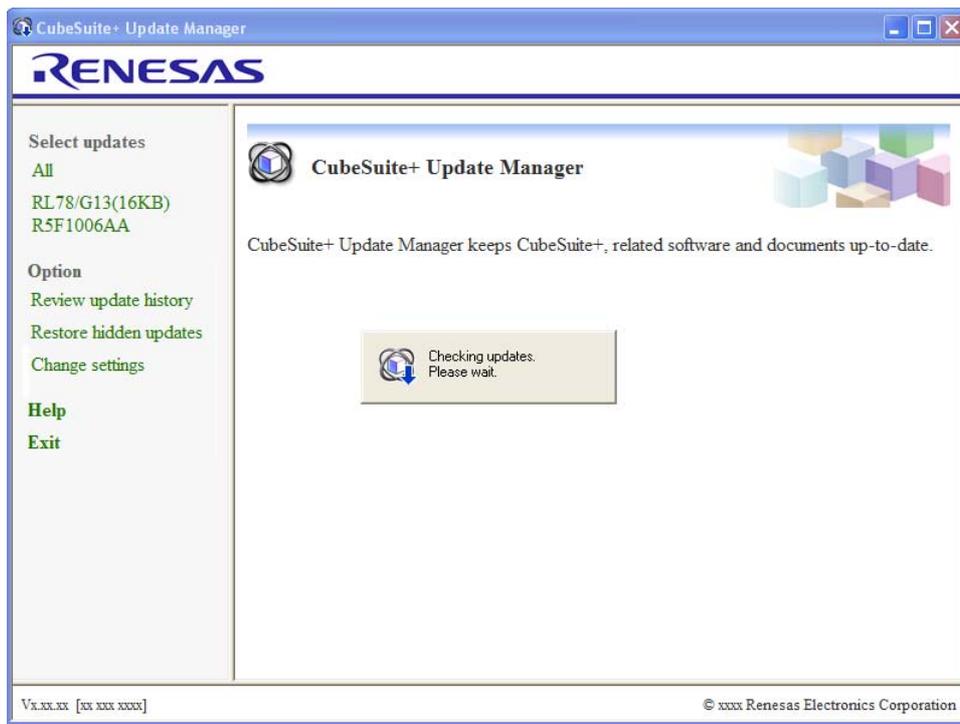


**Remarks 1.**    In Windows 8 or Windows 8.1, double-click on [Update Manager] on the start screen.

**2.**    The Checking for Updates dialog box appears as the manager checks for update information.

**Figure 2-20.   Checking for Updates Dialog Box**
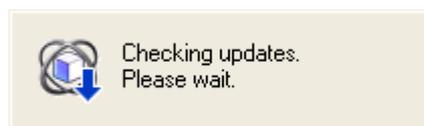
**(2) Select update items.**

A list of appropriate update items appears on the Select updates page.

**Figure 2-21.   Select updates Page**
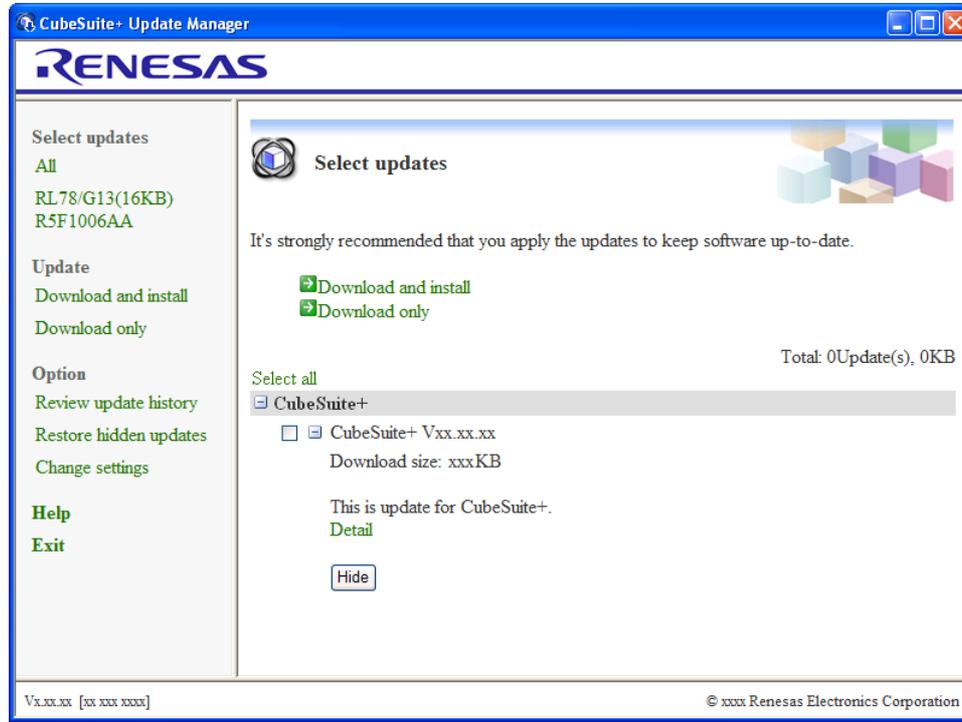


Select the check boxes of updates and click [Download and install] or [Download only] in the update selection area.

- 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.**    Click [Select all] in the update selection area to select the check boxes of all updates.

**3.**    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 2-22.   Download and install Page**
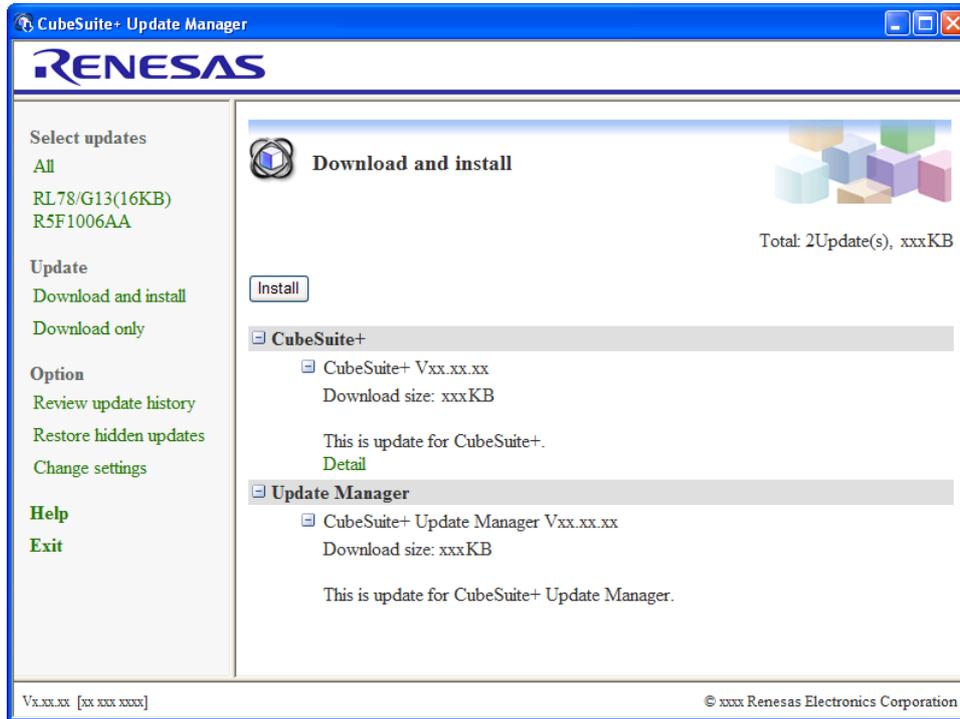


Check the information, and click the [Install] button.

**(4)  Confirm the download and install progress.**
Update items are downloaded, and then the 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 2-23.   Update in progress Dialog Box (Downloading)**
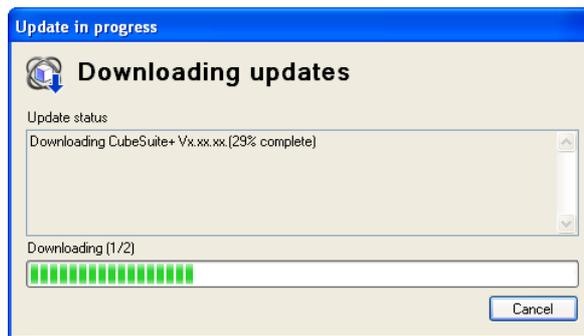
**Figure 2-24.   Update in progress Dialog Box (Installing)**
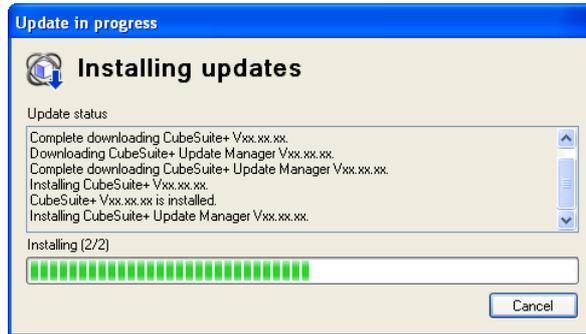


Cautions 1.    **You must have administrator privileges to perform installation.**

　　　　　2.    **If an emulator USB driver is installed on Windows Vista, Windows 7, Windows 8, or Windows 8.1, 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 2-25.   Download only Page**
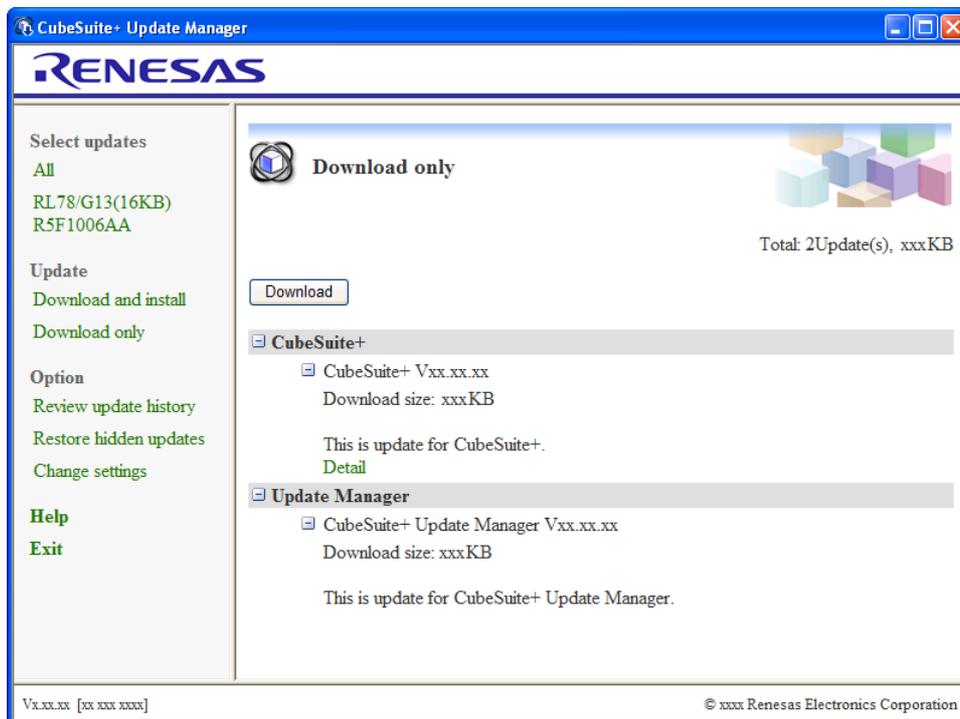


Check the information, and click the [Download] button.

**(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 2-26.   Update in progress Dialog Box**
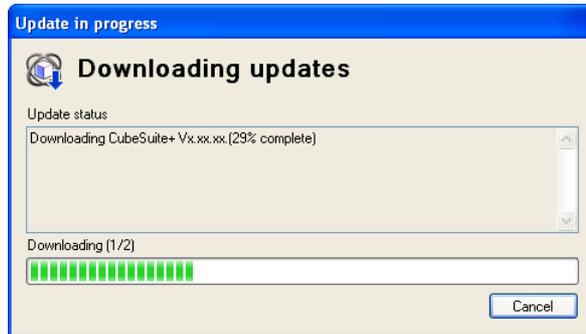


**Remark**    See "2.4.5   Displaying the update history" for instructions on installing download update items later.

**(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 2-27.   Finish Page**
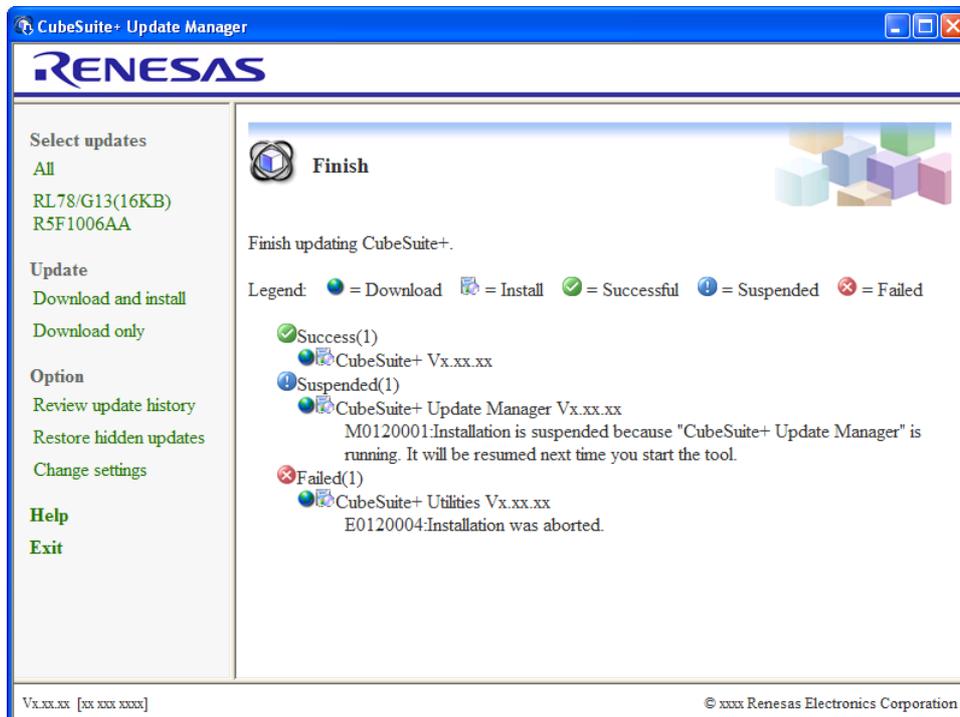


Click the [Exit] to complete updating.

**Remarks 1.**   The icons below are displayed in the head of update titles.

| | |
|---|---|
| 🌐 | Updates are downloaded. |
| 📥 | Updates are installed. |

**2.**   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. |

**2.4.2     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 2-28.   Option Dialog Box ([General - Update] Category)**
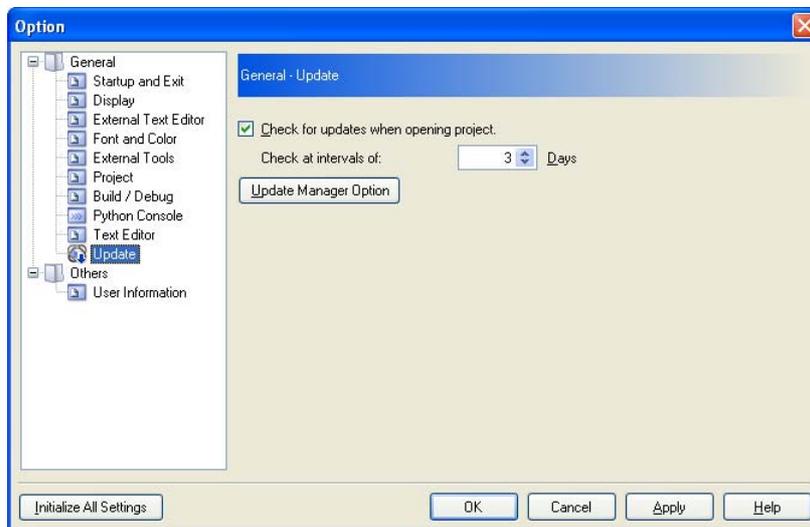


**Remarks 1.**   This function performs the check for updates in parallel with the CubeSuite+ process.  While the check is ongoing, the 🔄 icon appears in the Task Tray.

**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 2-29.   Update Manager Options Dialog Box**
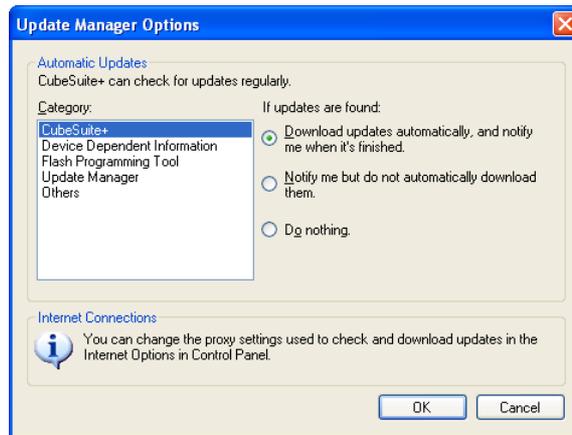


> **Remarks 1.**    During the download, the [icon] icon appears in the Task Tray.
> 
> **2.**    If the check does not find any updates, the auto download function exits.

**(3)  The update notification function**

After downloading the target update in (2), the update notification feature displays an icon [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] 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 2-30.   Update Manager Options Dialog Box**
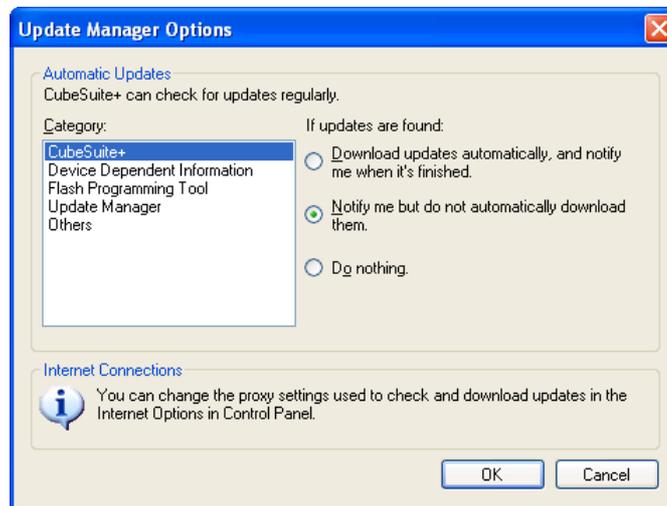


> **Remark**    Click the [icon] or [icon] icon to display the CubeSuite+ Update Manager window.

**Figure 2-31. CubeSuite+ Update Manager Window**
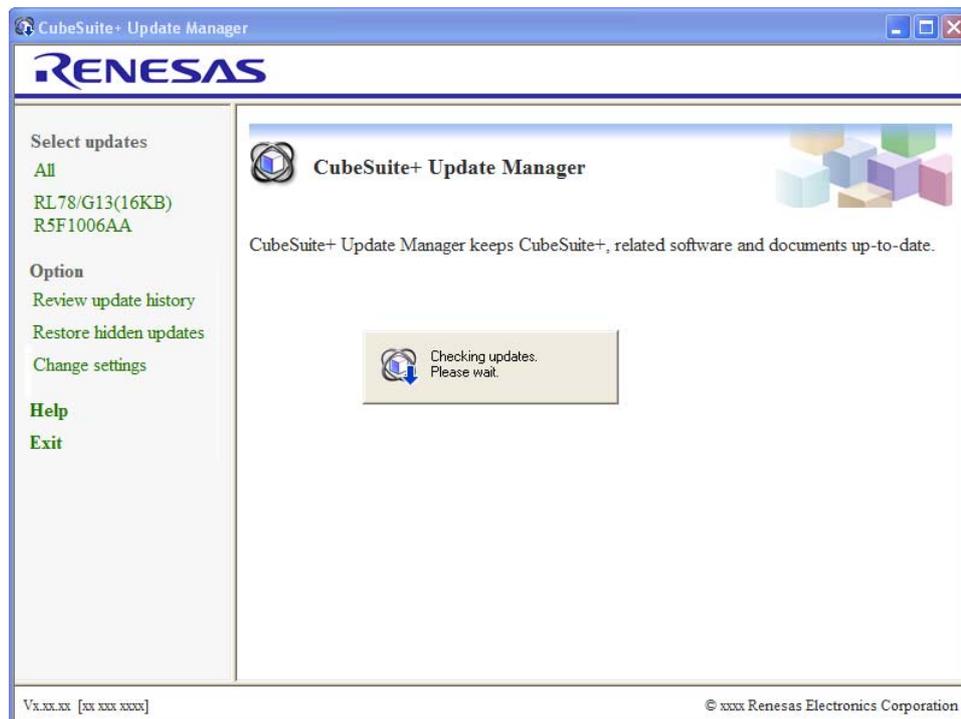


**Remark** If the check does not find any updates, the update notification function exits.

**Remark** Perform the installation in accordance with "2.4.1 The manual update feature", step (3).

### 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 2-32.   History  Page**
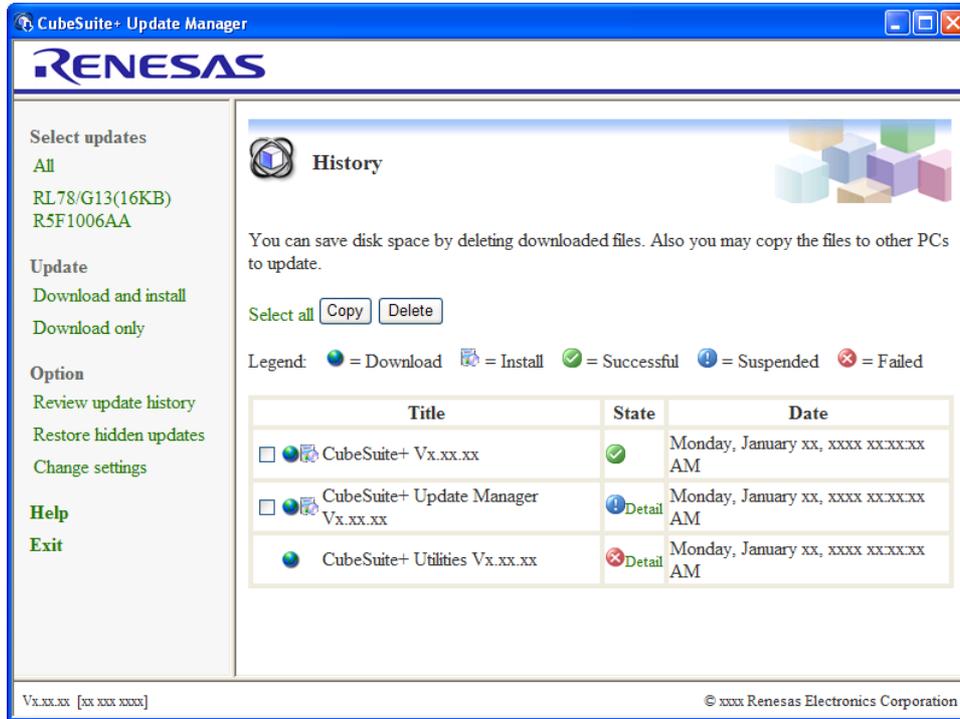


**(1)  Installing download files**

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 2-33.   History Page**
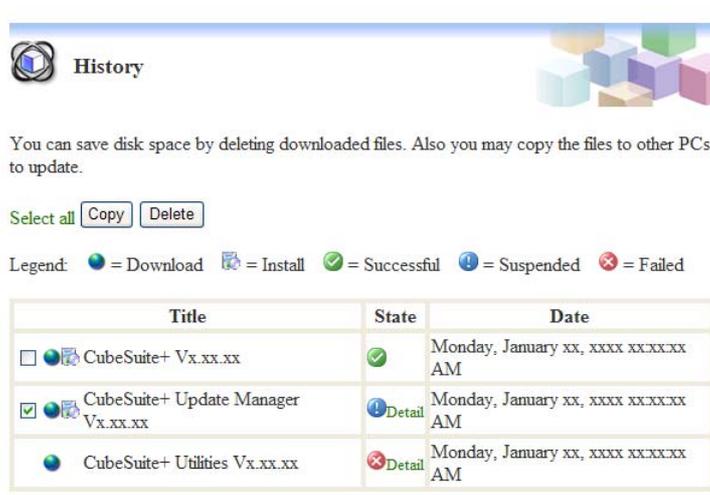
**(b) Select a copy destination folder.**

The Browse For Folder dialog box appears.

**Figure 2-34.   Browse For Folder Dialog Box**
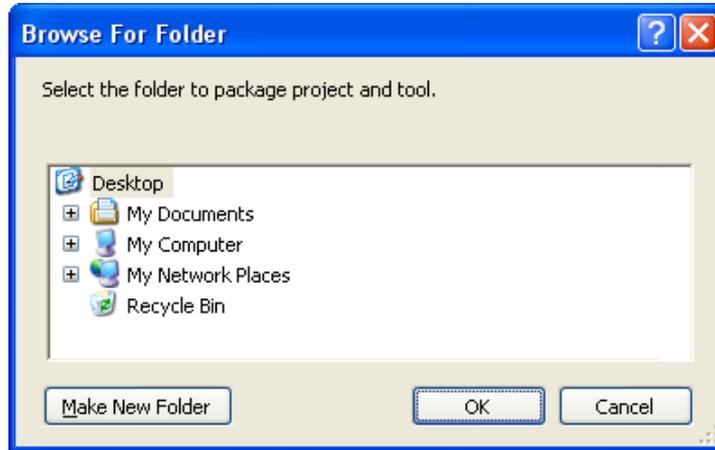


Select a copy destination folder, and click the [OK] button.

**(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 2-35.   History Page**

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

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

**Figure 2-36.   Message Dialog Box**
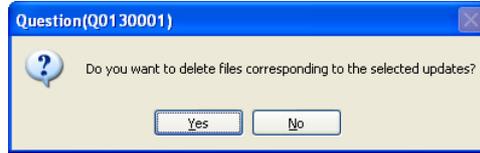


If you click the [Yes] button, the download files are deleted.

### 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 2-37.   Restore hidden updates Page**
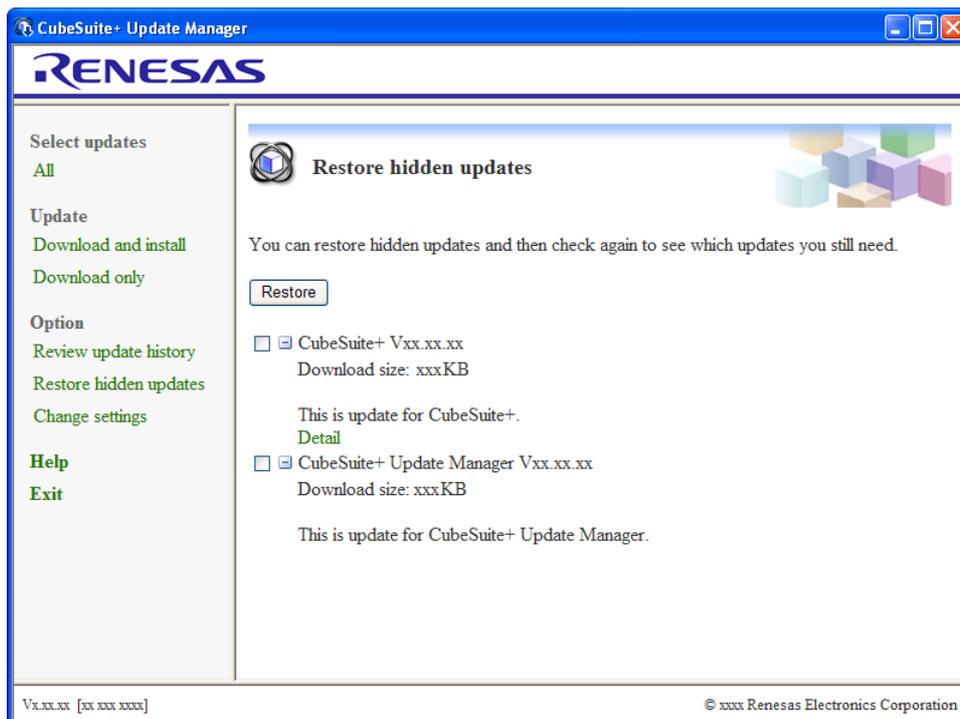
**(1)  Select updates.**

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

**Figure 2-38.   Restore hidden updates Page**
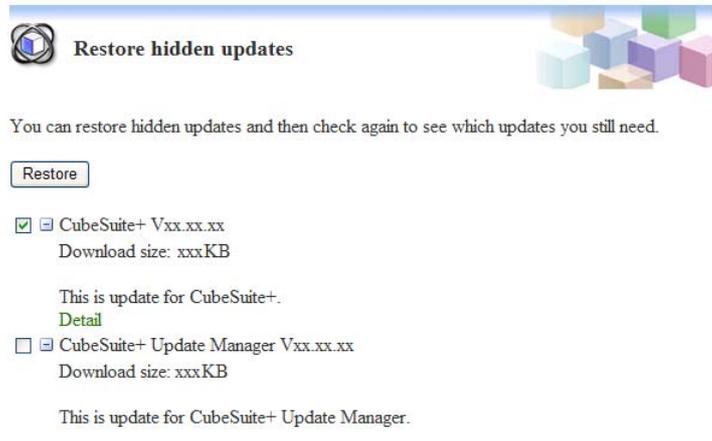


**(2)  Confirm that hidden updates are restored.**

Confirm that the selected updates become unvisible on the Restore hidden updates page and become visible on the Select updates page.

**Figure 2-39.   Restore hidden updates Page**
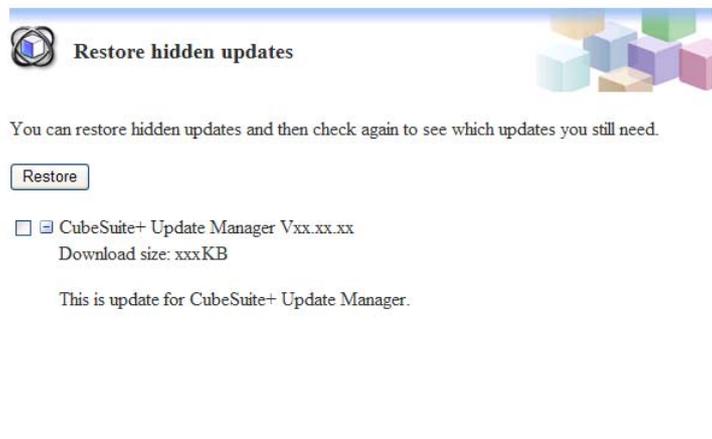
**Figure 2-40.   Select updates Page**
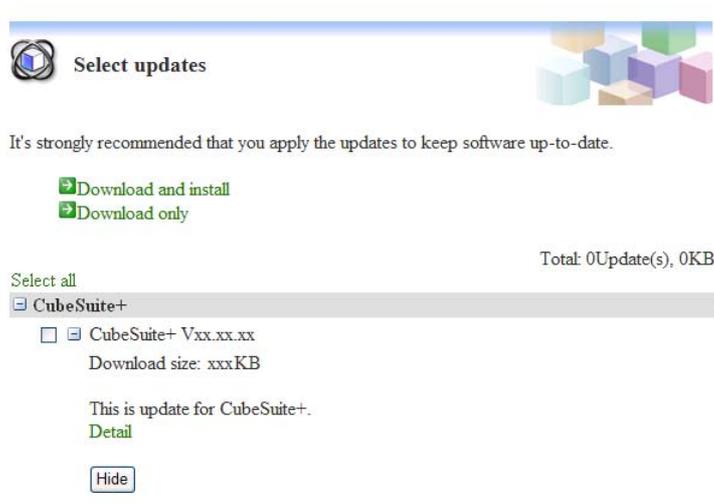
## 2.5   Start CubeSuite+

When CubeSuite+ is installed and started for the first time, select Windows [Start] menu >> [Programs] >> [Renesas Electronics CubeSuite+] >> [CubeSuite+].

Remark    In Windows 8 or Windows 8.1, double-click on [CubeSuite+] on the start screen.

The following Main window will be opened.

**Figure 2-41.   Main Window**
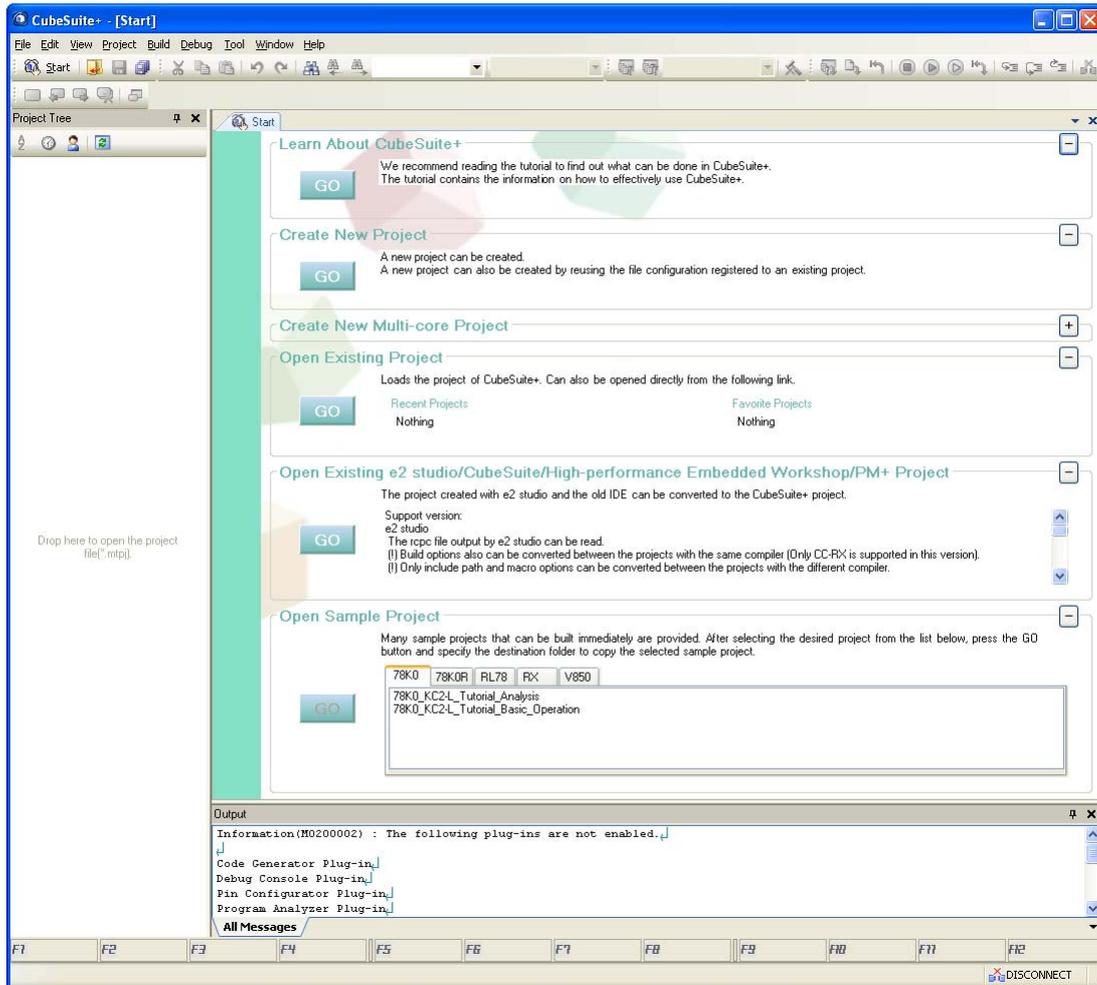


On the second and subsequent occasions, you can startup CubeSuite+ from the icon in the task tray when the rapid start (see "2.10.1   Use rapid start") is enabled.

Remark    The plug-ins other than the build tool, debug tool, and Editor panel are disabled by default (the disabled plug-ins are displayed in the Output panel).
Enable desired plug-ins on the [Additional Function] tab in the Plug-in Manager dialog box opened by selecting [Tool] menu >> [Plug-in Setting...].
To reflect the new settings, restart CubeSuite+.

## 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 (*.mtpj) and references it.

### (1)  Project tree organization and detailed settings

The project's settings are made on the Project Tree panel.

**Figure 2-42.   Project Tree Panel**
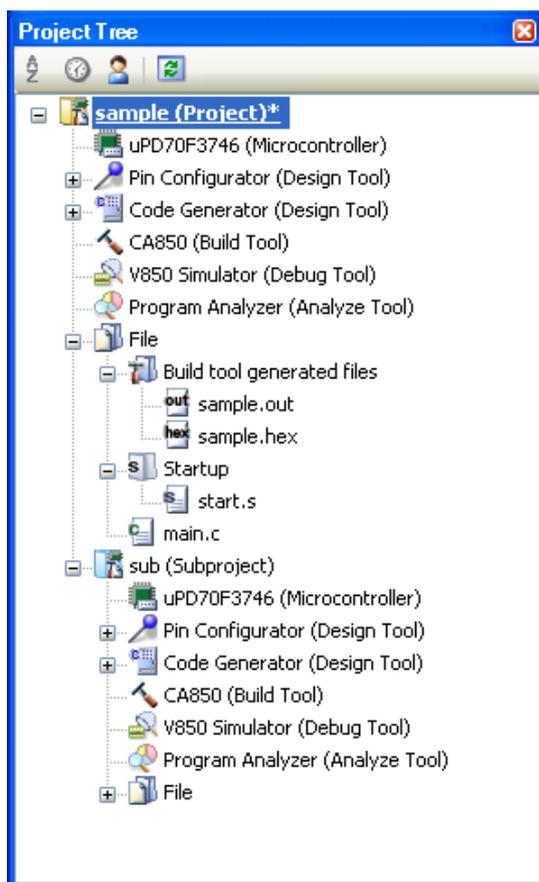


On the project tree, project components are displayed as the nodes below in a tree view.

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 |
|---|---|
| *Project name* (Project)<br>(Hereafter referred to as "Project node") | The project's name. |
| *Microcontroller name* (Microcontroller)<br>(Hereafter referred to as "Microcontroller node") | The microcontroller used in the project. |
| Boot Loader (Configuration Tool for Multi-core node) [RH850]<br>(Hereafter referred to as "Configuration Tool for Multi-core node") | The node for setting application projects which configure a project for multi-core.<br>Note that this node is shown when the project type is a boot loader project. |

| Node | Explanation |
|---|---|
| *Design Tool name* (Design Tool)<br>(Hereafter referred to as "Design Tool node") | The design tool (pin configurator, code generator, etc.) used.<br>Note that Code Generator (Design Tool) node is not shown when the project type is a debug-dedicated project. |
| *Build tool name* (Build tool)<br>(Hereafter referred to as "Build Tool node") | The build tool (compiler, assembler, etc.) used.<br>When the project type is a debug-dedicated project, "None" is shown as *build tool name*. |
| *Debug tool name* (Debug Tool)<br>(Hereafter referred to as "Debug Tool node") | The debug tool (in-circuit emulator, simulator, etc.) used. |
| Program Analyzer (Analyze Tool)<br>(Hereafter referred to as "Analyze Tool node") | The analyze tool used.<br>Note that this node is not shown when the project type is a debug-dedicated project. |
| File<br>(Hereafter referred to as "File node") | Files registered to the project are displayed directly below the File node. |
| Download files<br>(Hereafter referred to as "Download files node") | This is a node for adding download files to the project.<br>Note that this node is shown only when the project type is a debug-dedicated project. |
| Build tool generated files<br>(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).<br>Note that this node is not shown when the project type is a debug-dedicated project. |
| Files for Multi-core [RH850]<br>(Hereafter referred to as "Files for Multi-core node") | Files related to the project for multi-core are displayed directly below the Files for Multi-core node.<br>Note that this node is shown when an application project is related to the boot loader project.**Note** |
| Startup [V850][RL78][78K0R][78K0]<br>(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.<br>Note that this node is not shown when the project type is a debug-dedicated project. |
| *Category name*<br>(Hereafter referred to as "Category node") | These user-defined categories are used to classify files into modules. |
| *Subproject name* (Subproject)<br>(Hereafter referred to as "Subproject node") | Subprojects added to the project.<br>For subprojects, see "(2)   Projects and subprojects". |

**Note**   The boot loader project and the application project are related in the Select Constituent Application Projects dialog box that is opened from the [Constituent application projects] property of the Configuration Tool for Multi-core node.

**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 F   USING AN EXTERNAL BUILD TOOL" for details on a debug-dedicated project.

**(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 2-43.   Project Tree Panel (When Subprojects Added)**
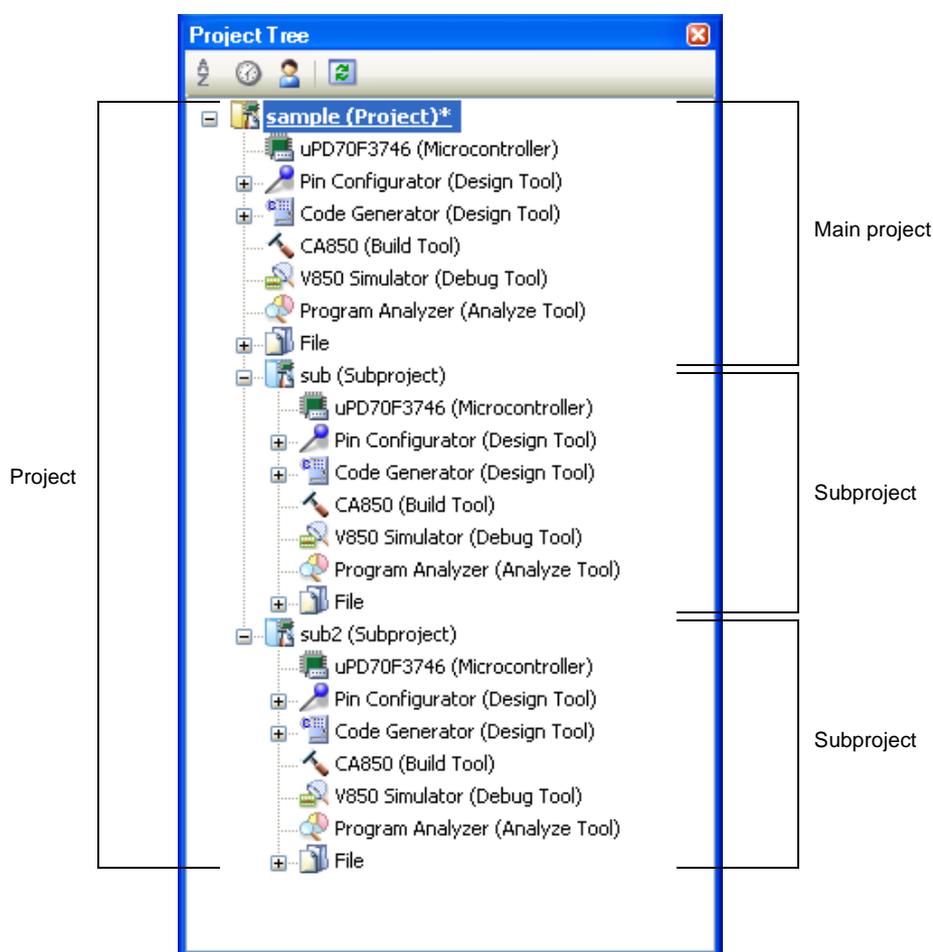


In addition, project settings and the settings for subprojects added to a project are independent and have no effect on each other.  When making the same settings between the main project and subprojects, or between differing subprojects, select multiple nodes to set on the project tree, and make the 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 2-44.   Start Panel**
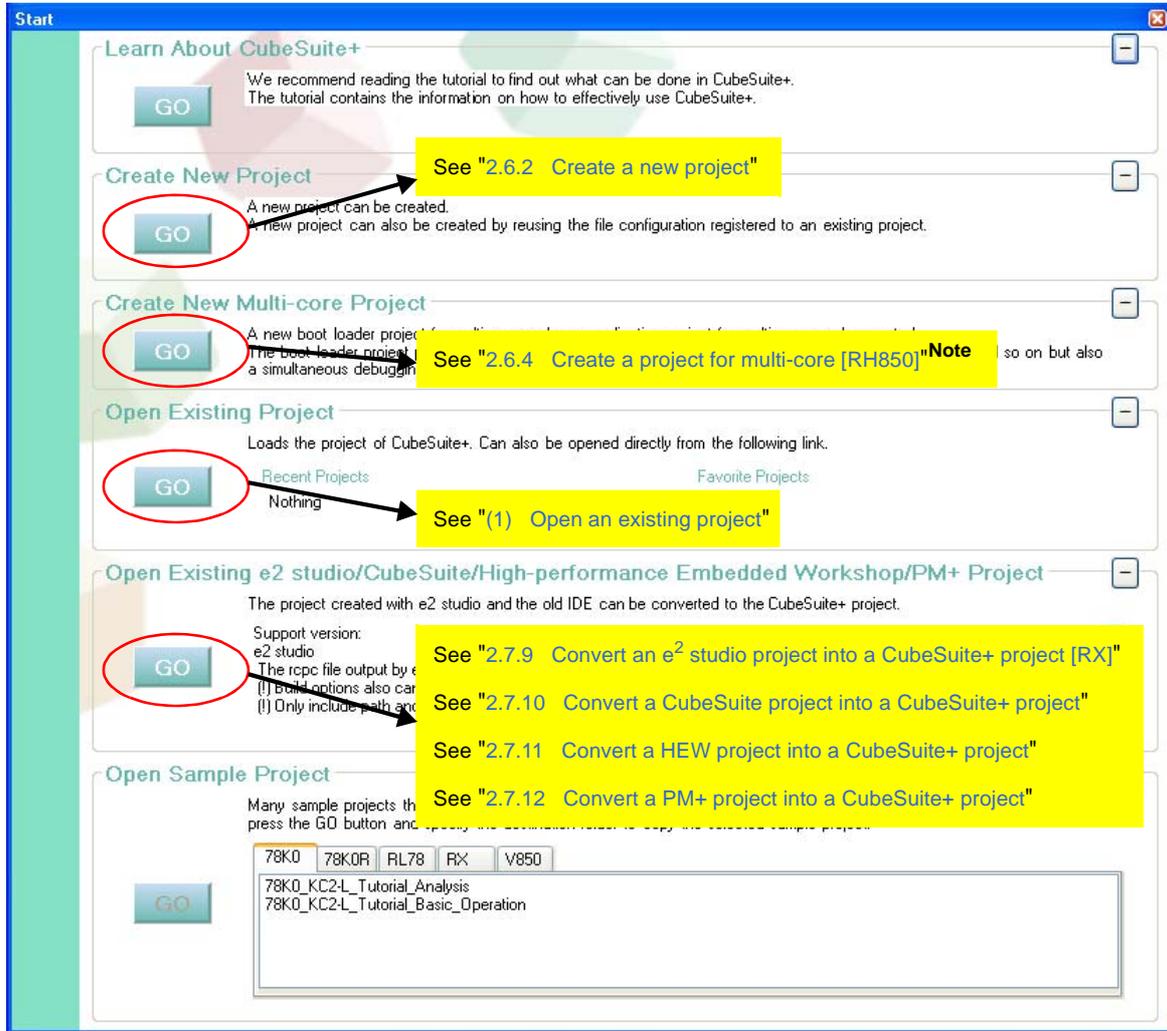


**Note**   This area is minimized by default; click ⊞ to resize the area.

**2.6.2     Create a new project**

This section describes how to create a new project.

>   **Remark**     When a project for RH850 multi-core is created, see "2.6.4   Create a project for multi-core [RH850]".

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

**Figure 2-45.   Create Project Dialog Box (When First Started)**
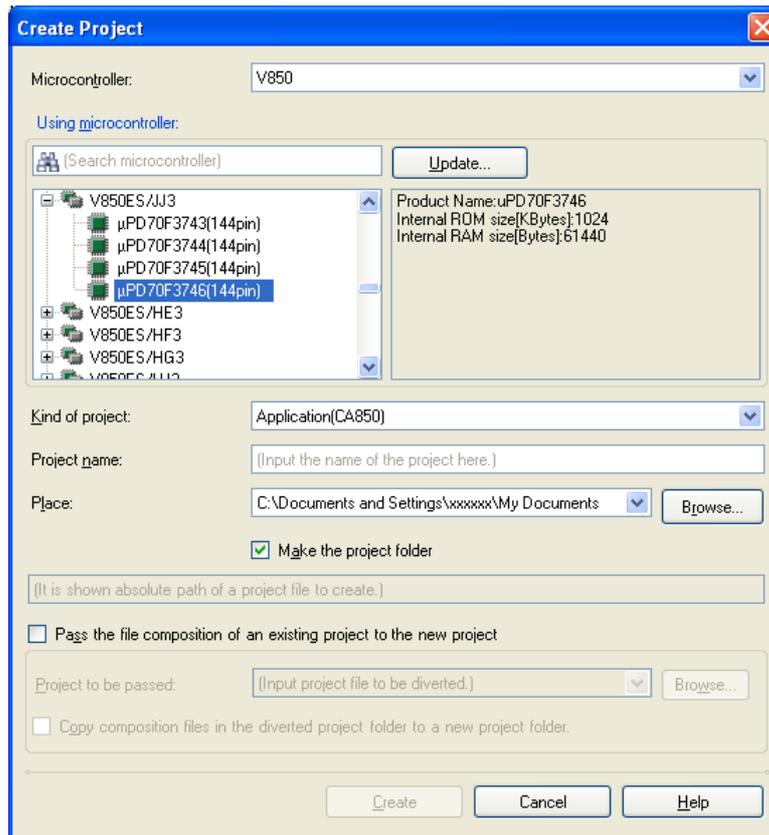


Set the items in the order below.

**(1)  Select the microcontroller type**
>   Select the microcontroller type to use in the project on [Microcontroller].
>   You can select the item below.
>   - RH850
>   - RX
>   - V850
>   - R8C (Localised support)
>   - RL78
>   - 78K0R
>   - 78K0

**(2)  Select the microcontroller**
>   Select the microcontroller to use in the project on the [Using microcontroller] area.
>   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.

> **Caution**    **The [Update...] button is only enabled when this product is installed using the installer.  It is disabled when a packaged item is being used.**

**(3)   Select the project type**

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

You can select the item below.

  - Application(CC-RH/CC-RX/CA850/CX/NC30 (Localised support)/CA78K0R/CA78K0)

Select this to generate the ROMization module file [CA850][CA78K0R], load module file, and hex file from C source files, by using the build tool provided by CubeSuite+.

The generated file will be the debug target.

> **Caution**    **This item is not displayed when microcontrollers for RH850 multi-core are selected in "(2) Select the microcontroller".**

> **Remarks 1.**    If the build tool is CC-RH, the following start-up source files are generated in the project folder. These files are also registered in the project tree.

| File Name | Description |
|---|---|
| cstart.asm | For defining the start-up routine from the occurence of a reset to a branch to the main function |
| iodefine.h | For defining I/O registers |
| main.c | For defining the empty main function |
| vecttbl.asm | For defining the interrupt vector table |

> **2.**    If the build tool is CC-RX, the following start-up source files are generated in the project folder. The files marked with "OK" in the "Registration in Project Tree" column are also registered in the project tree.  When necessary, also register the files marked with "--" in the project tree.

| File Name | Description | Registration in Project Tree |
|---|---|---|
| *ProjectName*.c | For defining the main function | OK |
| dbsct.c | For setting up standard sections | OK |
| hwsetup.c | For initializing hardware | -- |
| intprg.c | For defining interrupt functions | OK |
| iodefine.h | For defining I/O registers | OK |
| lowlvl.src | For defining low-level I/O functions (source file for assembler) | -- |
| lowsrc.c | For defining low-level I/O functions | -- |
| lowsrc.h | Prototypes for low-level I/O functions | -- |
| resetprg.c | For defining initialization for C language | OK |
| sbrk.c | For defining the function for allocating the heap memory | OK |
| sbrk.h | For defining the heap size | OK |

| File Name | Description | Registration in Project Tree |
|---|---|---|
| stacksct.h | For defining pragma for the stack | OK |
| typedefine.h | For defining typedefine for the types used in sbrk.c, etc. | OK |
| vect.h | Prototypes for interrupt vector functions | OK |
| vecttbl.c | For defining the interrupt vector table | OK |

**3.** If the build tool is NC30 (Localised support), the following start-up source files are generated in the project folder.
These files are also registered in the project tree.

| File Name | Description |
|---|---|
| device.c | For defining a constant or initialization of the standard input/output |
| firm.c | For defining a firmware area for OCD |
| fvector.c | For defining the fixed vector table |
| heap.c | For defining the static variable for the heap area |
| init.c | For defining initialization of individual standard inputs/outputs (stdin, stdout, etc.) |
| initsct.c | For defining initialization of each section |
| initsct.h | For defining a macro for initializing sections |
| intprg.c | For defining the address of the interrupt vector |
| resetprg.c | For defining initialization of the C language |
| resetprg.h | For defining a constant for resetprg.c, a macro, and include |
| sfr_r8xx.h | For defining SFR (for C language) |
| sfr_r8xx.inc | For defining SFR (for assembly language) |
| typedefine.h | For defining the scalar (integer) type |

- C++ Application(CC-RX)
  Select this to generate the load module file and hex file from a C++ source file (only a file that has a main function) and C source files, by using build tool CC-RX provided by CubeSuite+.
  The generated file will be the debug target.

**Remark** The following start-up source files are generated in the project folder.
The files marked with "OK" in the "Registration in Project Tree" column are also registered in the project tree. When necessary, also register the files marked with "--" in the project tree.

| File Name | Description | Registration in Project Tree |
|---|---|---|
| *ProjectName*.c | For defining the main function | OK |
| dbsct.c | For setting up standard sections | OK |
| hwsetup.c | For initializing hardware | -- |
| intprg.c | For defining interrupt functions | OK |
| iodefine.h | For defining I/O registers | OK |

| File Name | Description | Registration in Project Tree |
|---|---|---|
| lowlvl.src | For defining low-level I/O functions (source file for assembler) | -- |
| lowsrc.c | For defining low-level I/O functions | -- |
| lowsrc.h | Prototypes for low-level I/O functions | -- |
| resetprg.c | For defining initialization for C language | OK |
| sbrk.c | For defining the function for allocating the heap memory | OK |
| sbrk.h | For defining the heap size | OK |
| stacksct.h | For defining pragma for the stack | OK |
| typedefine.h | For defining typedefine for the types used in sbrk.c, etc. | OK |
| vect.h | Prototypes for interrupt vector functions | OK |
| vecttbl.c | For defining the interrupt vector table | OK |

- Empty Application(CC-RH/CC-RX/NC30 (Localised support))
  Select this to generate the load module file, by using build tool CC-RH/CC-RX/NC30 (Localised support) provided by CubeSuite+.
  The generated file will be the debug target.
  Sample startup programs are not generated when the project is created.

- Boot Loader for Multi-core(CC-RH)
  Select this to create a boot loader project for multi-core, by using build tool CC-RH provided by CubeSuite+.
  A boot loader project is used to set application projects configuring a project for multi-core.

**Caution**   **This item is displayed only when microcontrollers for RH850 multi-core are selected in "(2) Select the microcontroller".**

**Remark**   The following start-up source files are generated in the project folder.
  These files are also registered in the project tree.

| File Name | Description |
|---|---|
| boot.asm | For defining the processing from the occurence of a reset to a branch to each application project |
| iodefine.h | For defining I/O registers |
| vecttbl.asm | For defining the interrupt vector table |

- Application for Multi-core(CC-RH)
  Select this to create an application project for multi-core, by using build tool CC-RH provided by CubeSuite+.

**Caution**   **This item is displayed only when microcontrollers for RH850 multi-core are selected in "(2) Select the microcontroller".**

**Remark**   The following start-up source files are generated in the project folder.
  These files are also registered in the project tree.

| File Name | Description |
|-----------|-------------|
| cstartm.asm | For defining the start-up routine for each application |
| iodefine.h | For defining I/O registers |
| main.c | For defining the empty main function |

- Library(CC-RH/CC-RX/CA850/CX/NC30 (Localised support)/CA78K0R/CA78K0)
  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 F   USING AN EXTERNAL BUILD TOOL" for details on how to create and use the debug-dedicated 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.

**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 an e$^2$ studio, CubeSuite, High-performance Embedded Workshop, or PM+ project file.**
**If you wish to copy an existing e$^2$ studio, CubeSuite, High-performance Embedded Workshop, or PM+ project, open the project in CubeSuite+, then save it as a CubeSuite+ project (see "2.7.9 Convert an e$^2$ studio project into a CubeSuite+ project [RX]", "2.7.10   Convert a CubeSuite project into a CubeSuite+ project", "2.7.11   Convert a HEW project into a CubeSuite+ project", "2.7.12   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 "2.7.8   Convert a CA850 project into a CX project").

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

**Figure 2-46. Create Project Dialog Box (After Setting Items)**
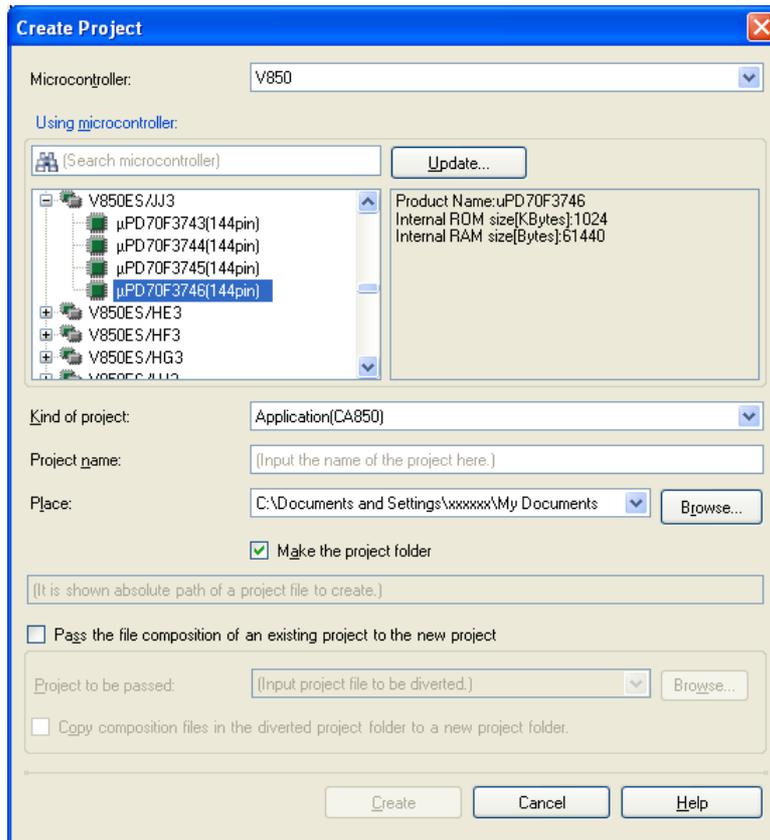


When you click the [Create] button, the project file is created in the location specified in (4) and the structure of the created project is displayed as a tree in the Project Tree panel.

**Figure 2-47. Project Tree Panel (After Creating a New Project)**
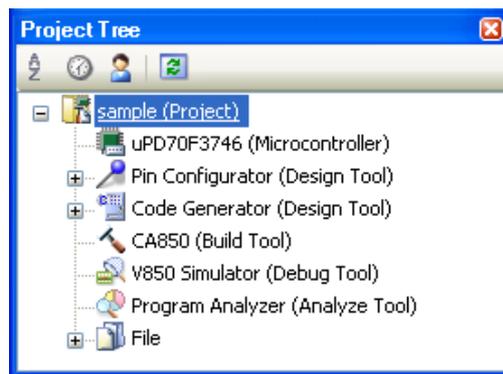


**Remark** After creating a project, you must add target files to the project in order to perform building or debugging.

For details on how to add these files, see the following.

- When "Application" or "Library" is selected on [Kind of project]
  -> See the "CubeSuite+ Integrated Development Environment User's Manual: Build"
- When "Debug Only" is selected on [Kind of project]
  -> See "F.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 2-48.   Create Project Dialog Box (When Adding a New Subproject)**
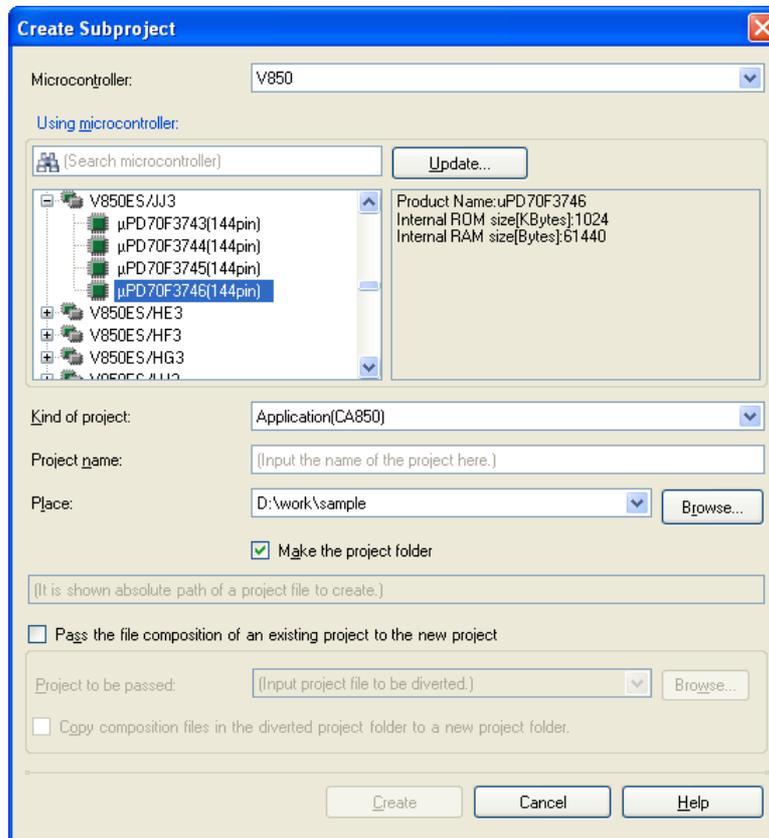


After setting each item on the dialog box, click the [Create] button (For more on the settings for each item, see "2.6.2 Create a new project").

**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 "2.7.8   Convert a CA850 project into a CX project").

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

**Figure 2-49.   Project Tree Panel (After Adding a Subproject)**
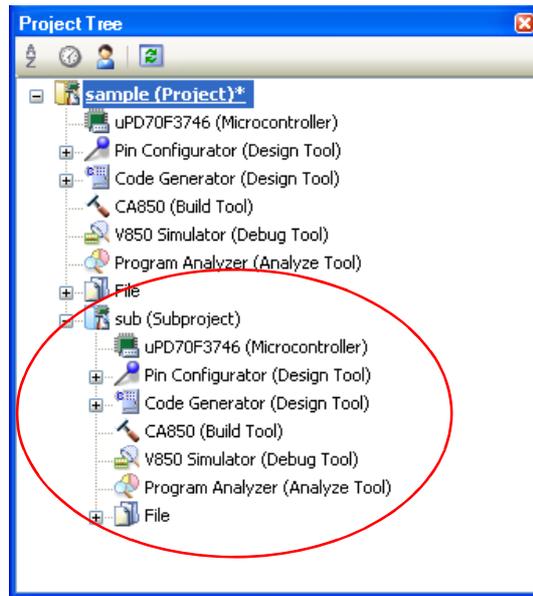
**2.6.4    Create a project for multi-core [RH850]**

A project for multi-core consists of a boot loader project and an application project.  The application project creates pro-
grams for each CPU core, and the boot loader project manages activation of those programs.

The following shows how to create a project for multi-core by specifying the boot loader project and the application
project as the main and subprojects, respectively.

On the tool bar, click &#x1F50D; Start to open the Start panel.

**Figure 2-50.   Start Panel**
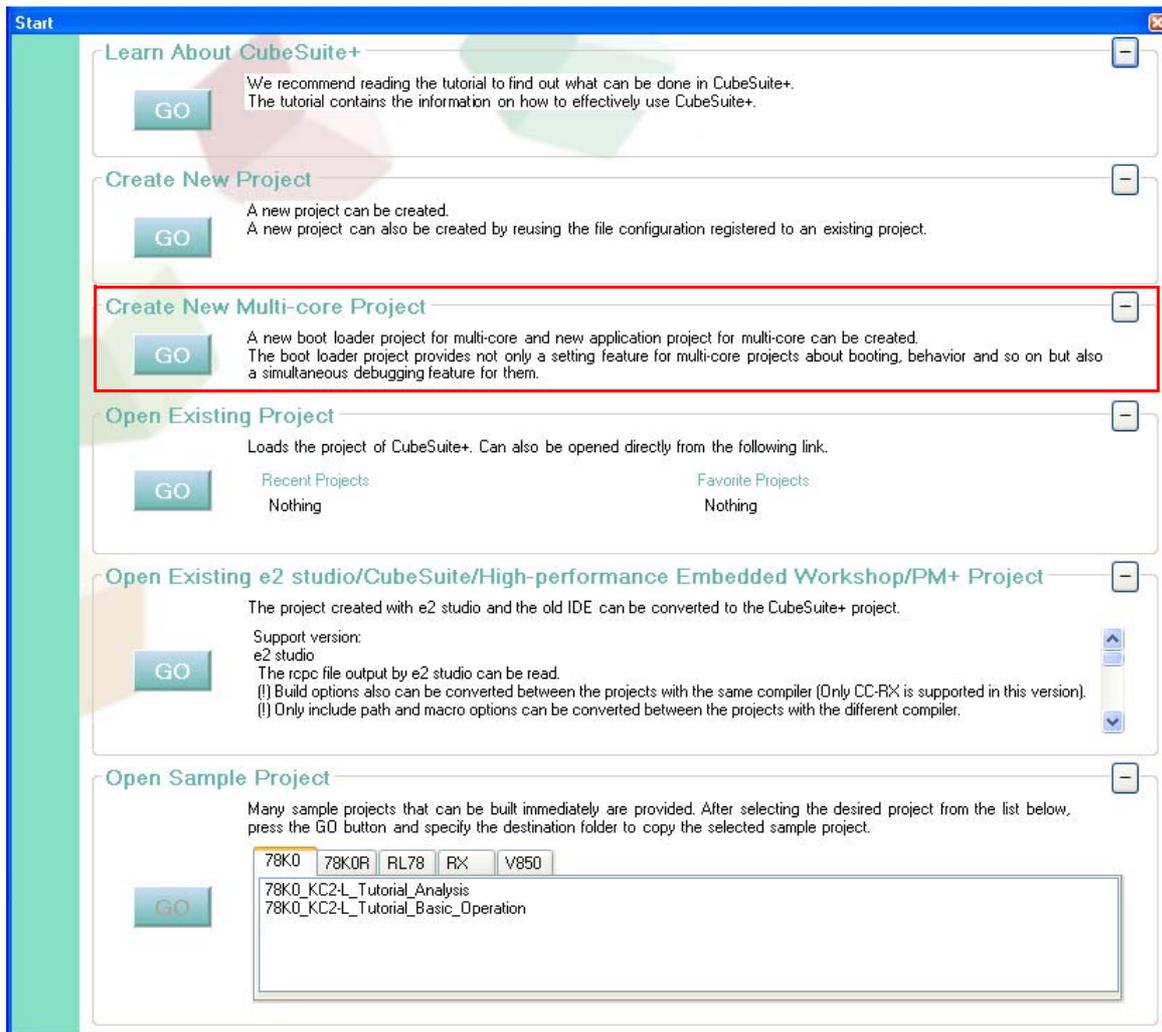


Click the [GO] button in the [Create New Multi-core Project] area to open the Create Project dialog box.

**Remark**    The [Create New Multi-core Project] area is minimized by default; click $\boxed{+}$ to resize the area.

**Figure 2-51.   Create Project Dialog Box (When First Started)**
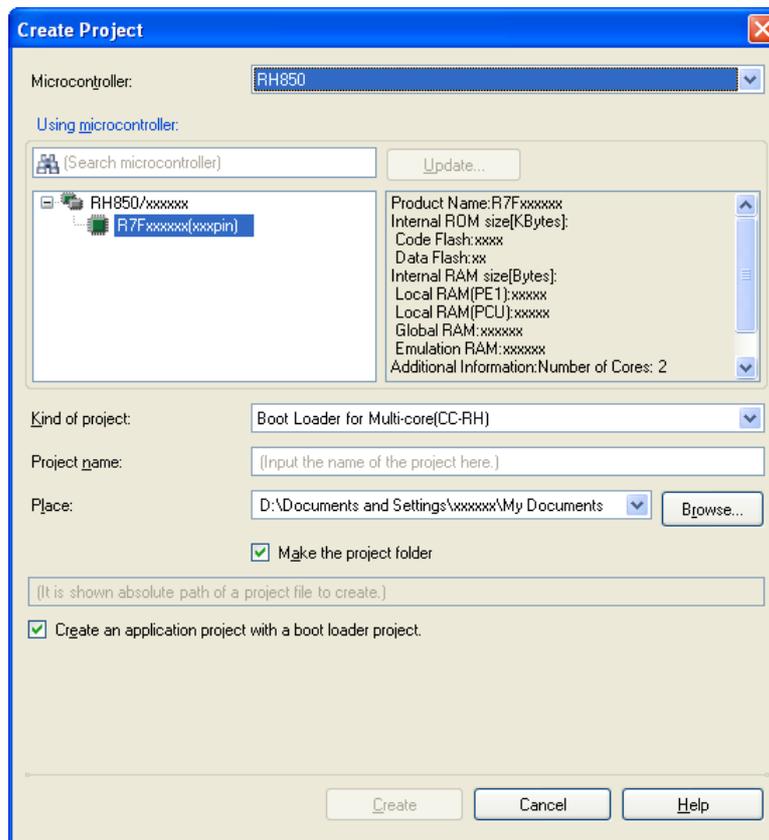


Set the items in the order below.


**(1)  Confirm the microcontroller type**

[RH850] is selected on [Microcontroller].


**(2)  Select the microcontroller**

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

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.


Caution   **The [Update...] button is only enabled when this product is installed using the installer.  It is dis-abled when a packaged item is being used.**


**(3)  Confirm the project type**

[Boot Loader for Multi-core(CC-RH)] is selected on [Kind of 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.


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

**(5) Specify simultaneous creation of the application project**

When an application project is created at the same time as a subproject for the boot loader project, select [Create an application project with a boot loader project].

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

**Figure 2-52.   Create Project Dialog Box (After Setting Items)**
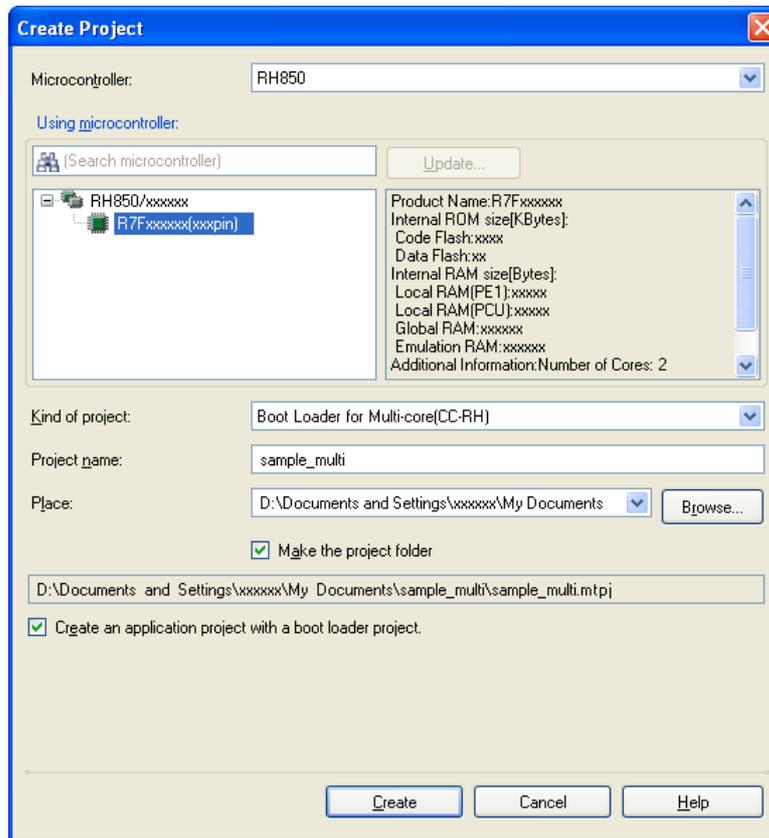


When you click the [Create] button, the project file is created in the location specified in (4) and the structure of the created project is displayed as a tree in the Project Tree panel.

**Figure 2-53.   Project Tree Panel (After Creating a New Project)**
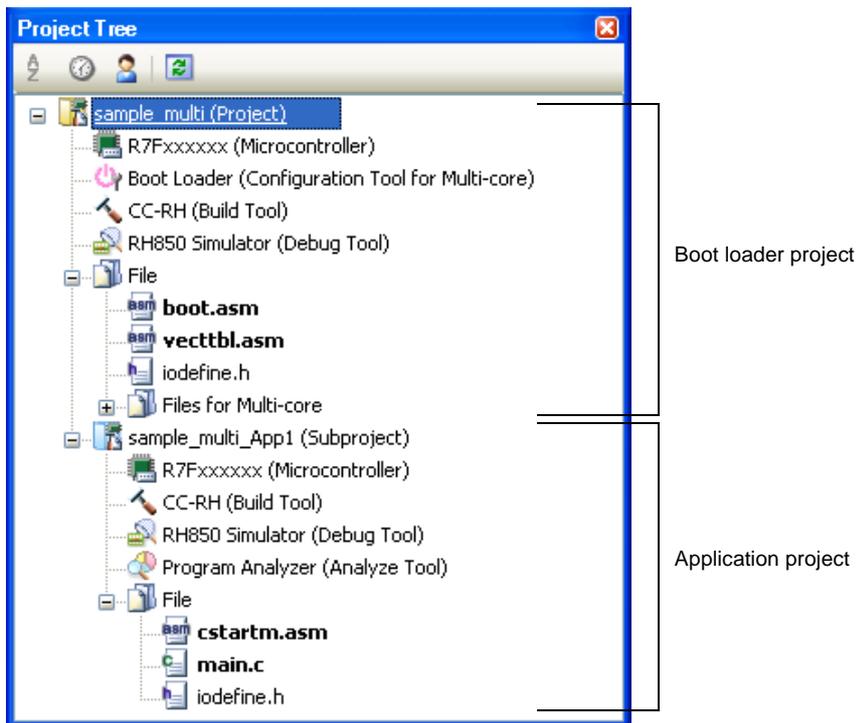


The start-up source files for each project are also created in the project folder and registered in the project tree.

- Start-up source files for the boot loader project

| File Name | Description |
|---|---|
| boot.asm | For defining the processing from the occurence of a reset to a branch to each application project |
| iodefine.h | For defining I/O registers |
| vecttbl.asm | For defining the interrupt vector table |

- Start-up source files for the application project

| File Name | Description |
|---|---|
| cstartm.asm | For defining the start-up routine for each application |
| iodefine.h | For defining I/O registers |
| main.c | For defining the empty main function |

**Remarks 1.** The application project name is "*boot-loader-project-name*_App1".

**2.** Add further application projects as subprojects.

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

## 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 2-54.   Open Project Dialog Box**
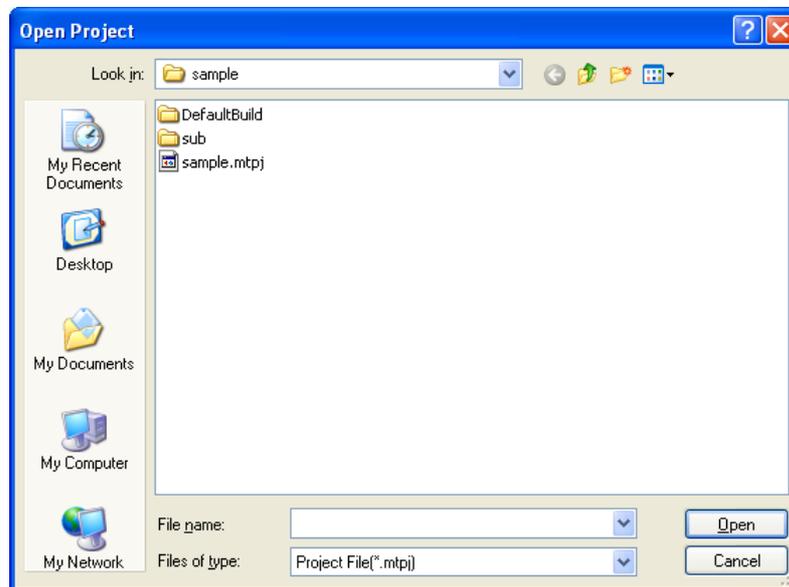


On the dialog box, specify the project file and click the [Open] button.

> **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 2-55.   [Recent Projects] Item**
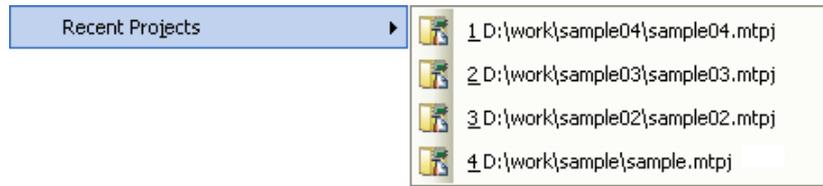


**(3)  Open a project from the favorites menu**

Open a project registered on the favorites menu.

From the [Project] menu, select [Favorite 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 2-56.   [Favorite Projects] Item**
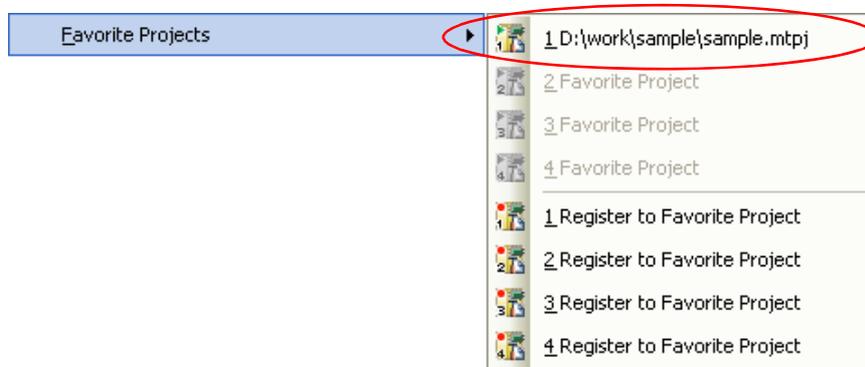


### 2.7.2    Add an existing subproject

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 2-57.   Add Existing Subproject Dialog Box**
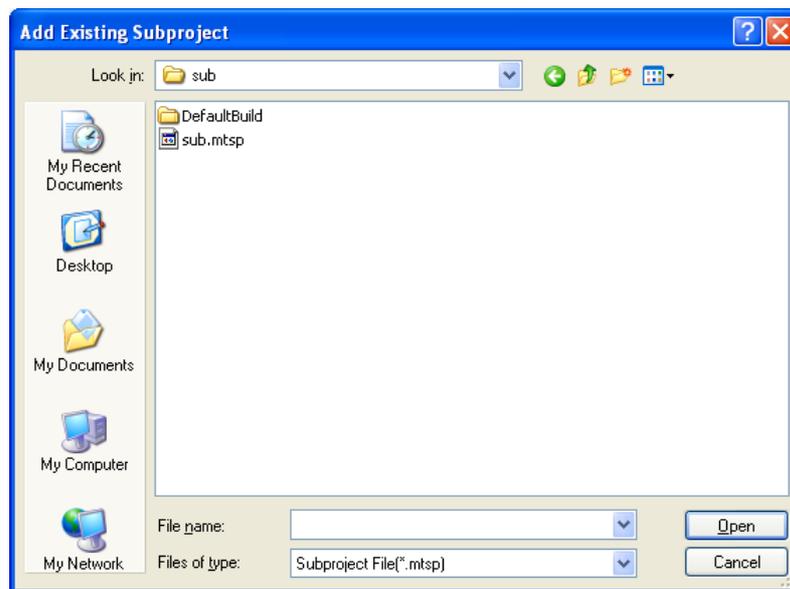


On the dialog box, specify the subproject file of the subproject to add and click the [Open] button.

### 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 [Favorite Projects] >> [*1 - 4* Register to Favorite Project], the path of the currently open project is registered under the [Project] menu >> [Favorite Projects].

**Figure 2-58.   [Register to Favorite Project] Item**
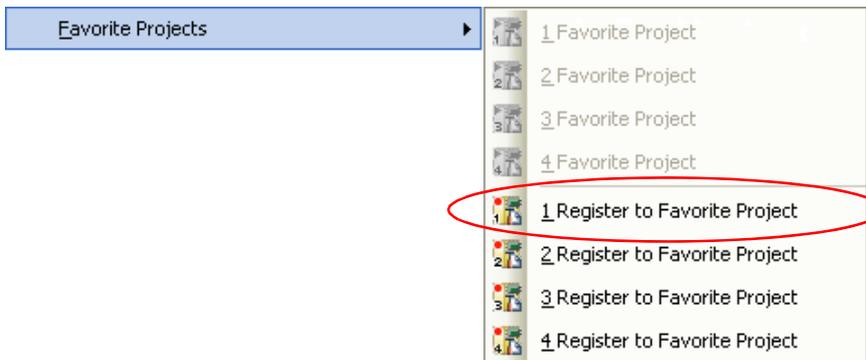


**Figure 2-59.   [Favorite Projects] Item (After Registering a Project)**
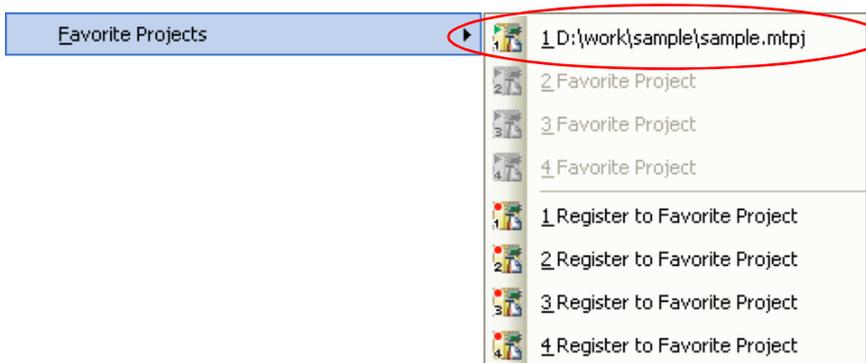
### 2.7.4   Remove a subproject from the project

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 2-60.   [Remove from Project] Item**
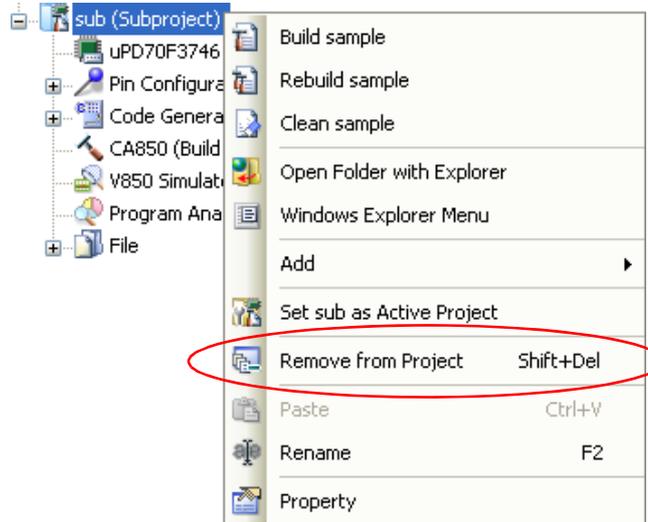


### 2.7.5   Change the project name

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 2-61.   [Rename] Item (For a Project)**
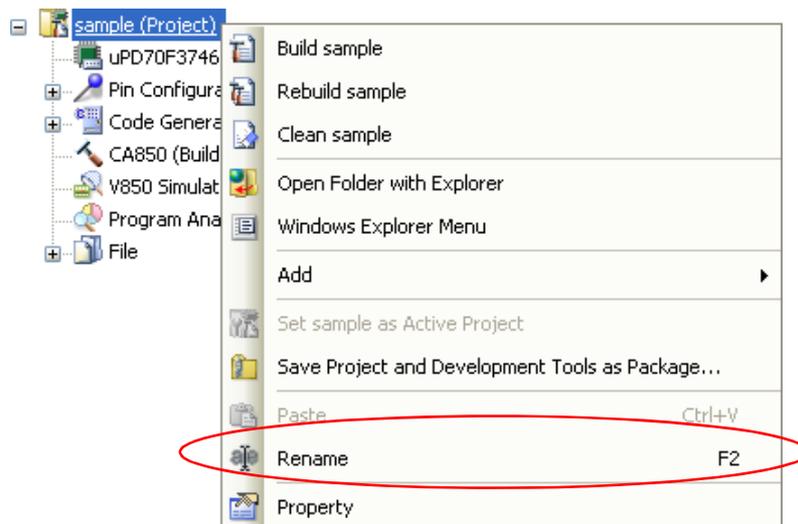


**Remark**   After changing the project name, when you save the project, the actual name of the project file is also changed.

### 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 2-62.   [Open Folder with Explorer] Item (For a Project)**
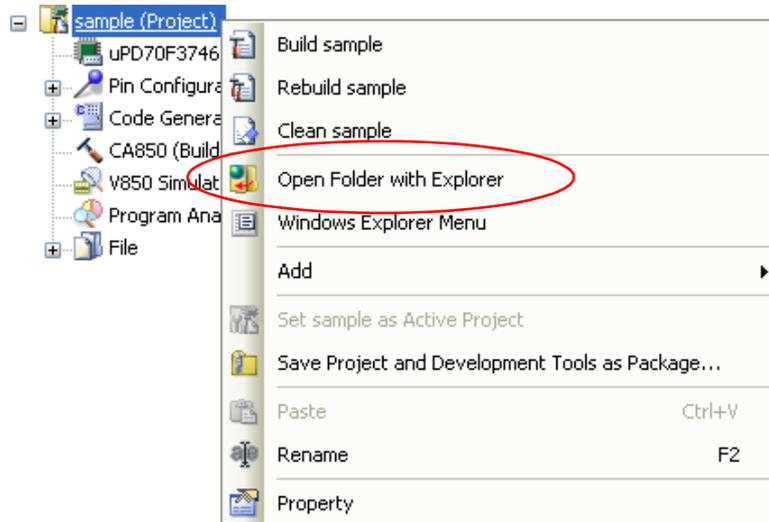


Remark    When you select [Open Folder with Explorer] from a file's context menu, the folder the file is saved in is opened in Explorer.

### 2.7.7    Set the build order of projects

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

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

However, when dependent projects have been set, builds of those projects are run first.

Dependent projects are set in the Dependent Projects Settings dialog box which is opened by selecting the [Project] menu >> [Dependent Projects Settings...].

**Figure 2-63.   Dependent Projects Settings Dialog Box**
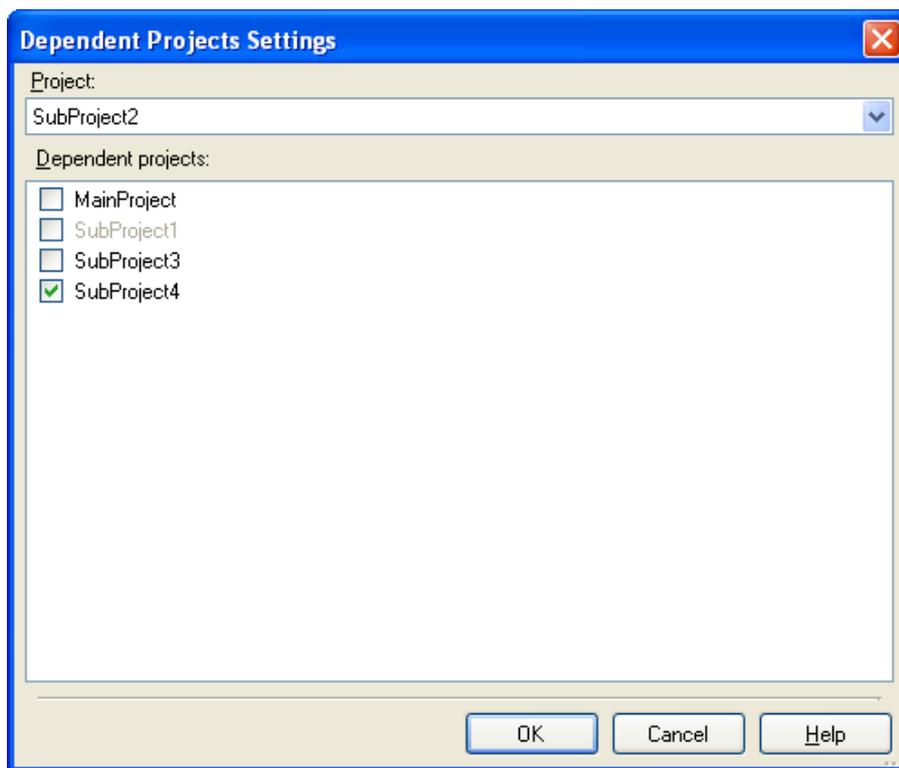


Select the dependent-source project in [Project], select a project to be referenced as the dependent project from a check box in [Dependent projects], and click the [OK] button.

Remarks 1.   By default, an active project is selected in [Project].
       2.   A cyclic-reference project is displayed in gray in [Dependent projects].

Examples 1.   Builds are run in the following order when no dependent project is set.
             SubProject1 -> SubProject2 -> SubProject3 -> SubProject4 -> MainProject
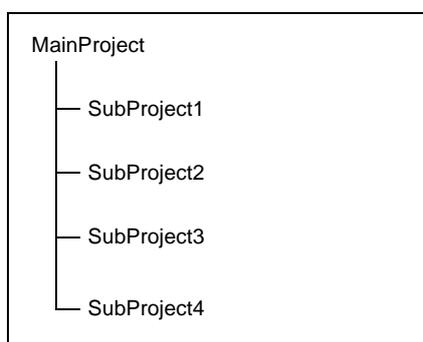


       2.   Builds are run in the following order when MainProject and SubProject2 are set as dependent projects for SubProject1 and SubProject4 is set as the dependent project for SubProject2.
             SubProject4 -> SubProject2 -> MainProject -> SubProject1 -> SubProject3
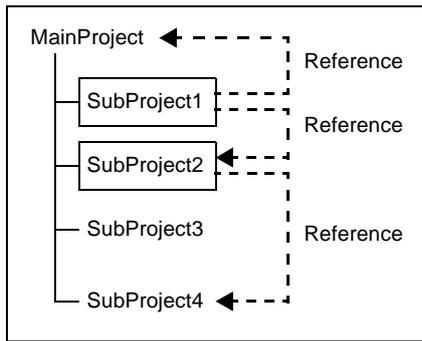
### 2.7.8      Convert a CA850 project into a CX project

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 "2.7.12    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 2-64.   Create Project Dialog Box**
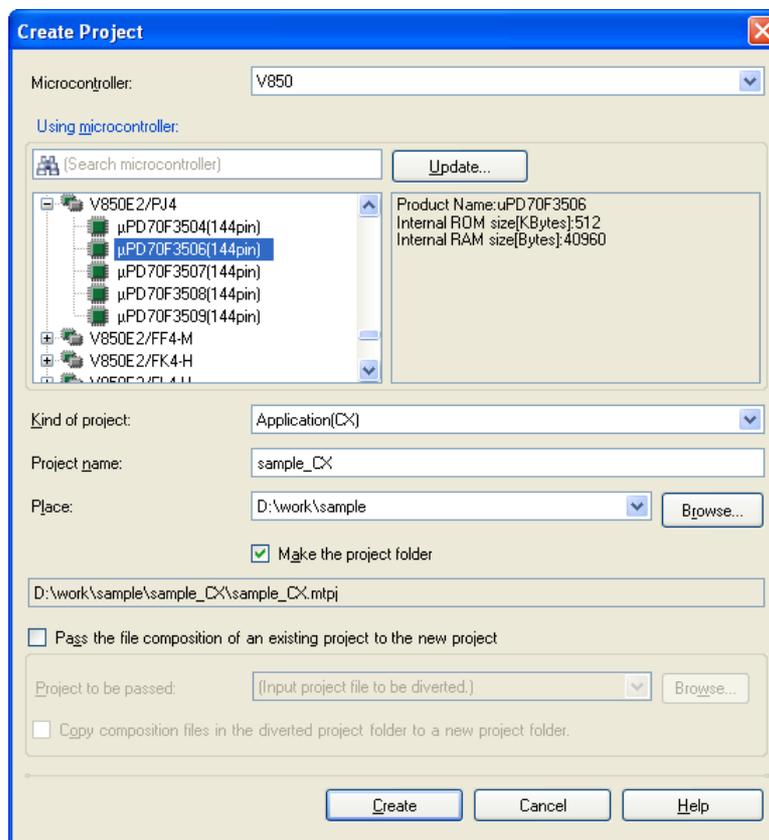


Set the items in the order below and click the [Create] button.


**(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 2-65.   Message Dialog Box**
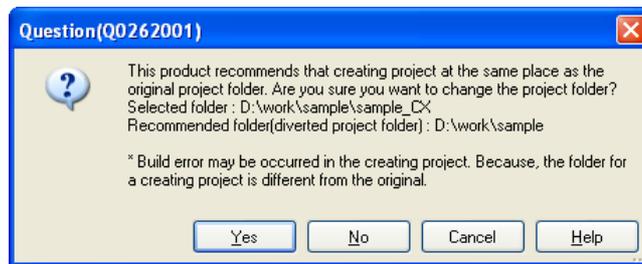


Click the [Create] button.  The Source Convert Setting dialog box [CX] appears.

**Figure 2-66.   Source Convert Setting Dialog Box**
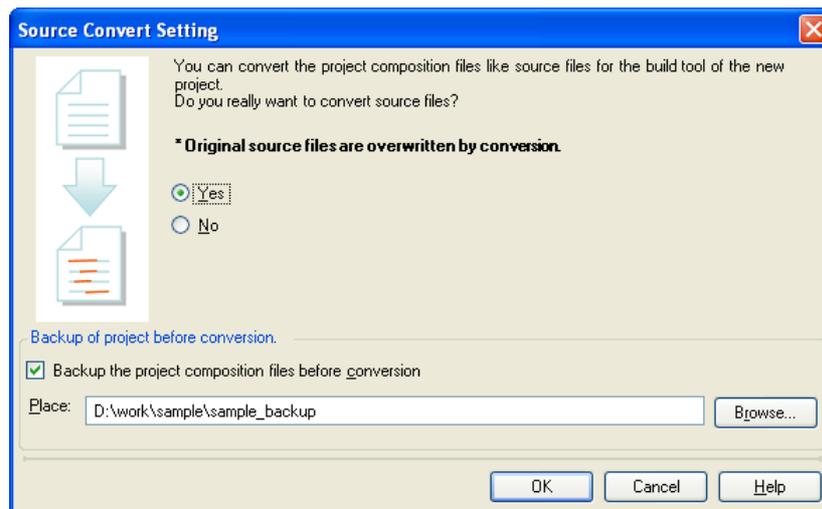


Select [Yes] to perform conversion on the source files.

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 "*dddd*, *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) | Options not to use(*Build mode*)<br><br>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.<br><br><br>  *Command name(Tab name of build tool property)*<br>   *Option*<br><br>        *:*<br><br><br>- This item is output only when the corresponding option exists.<br>- CA850 options are converted into CX options which have the same function.<br>  The option that has the same function and different name is not output.<br>- Build modes are output in the following order: "DefaultBuild", user-created build mode ("DefaultBuild" is the build mode that CubeSuite+ provides by default).<br>  See "CubeSuite+ Integrated Development Environment User's Manual: Build" for detail about a build mode. |
| (5) | Options to change(*Build mode*)<br><br>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.<br><br><br>  *Command name(Tab name of build tool property)*<br>   *Option of CA850 project -> Option of CX project*<br><br>        *:*<br><br><br>- This item is output only when the corresponding option exists.<br>- CA850 options are converted into CX options which have the same function.<br>  The option that has the same function and different name is not output.<br>- Build modes are output in the following order: "DefaultBuild", user-created build mode ("DefaultBuild" is the build mode that CubeSuite+ provides by default).<br>  See "CubeSuite+ Integrated Development Environment User's Manual: Build" for detail about a build mode. |

### 2.7.9     Convert an e$^2$ studio project into a CubeSuite+ project [RX]

You can convert an e$^2$ studio project for RX into a CubeSuite+ project.  Select [Open Project...] from the [Project] menu with CubeSuite+, and then select a project file (*.rcpc).

**Remark**     The project file (*.rcpc) for e$^2$ studio is output by using the export function of e$^2$ studio.

**(1) Select the project for conversion settings**

The Project Convert Setting dialog box will open.

**Figure 2-67.  Project Convert Setting Dialog Box**
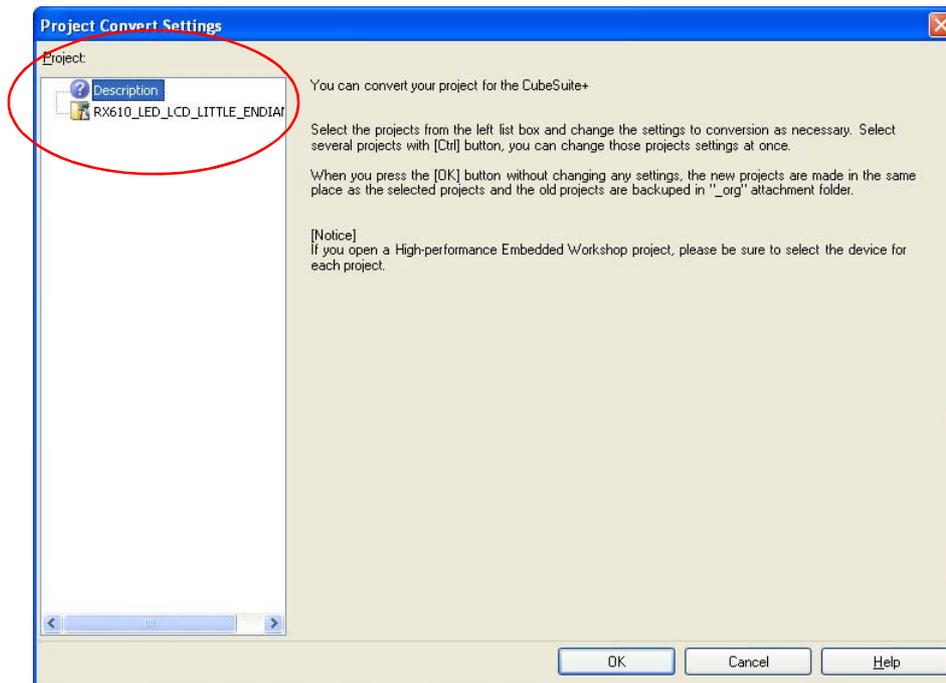


The [Project] area shows the configuration of conversion target projects as a tree; select the project for carrying out the conversion settings.

**(2) Set the conversion target project**

When you select the project, the area on the right shows the conversion target project setting items.

**Figure 2-68.   Project Convert Setting Dialog Box (When Project Is Selected)**
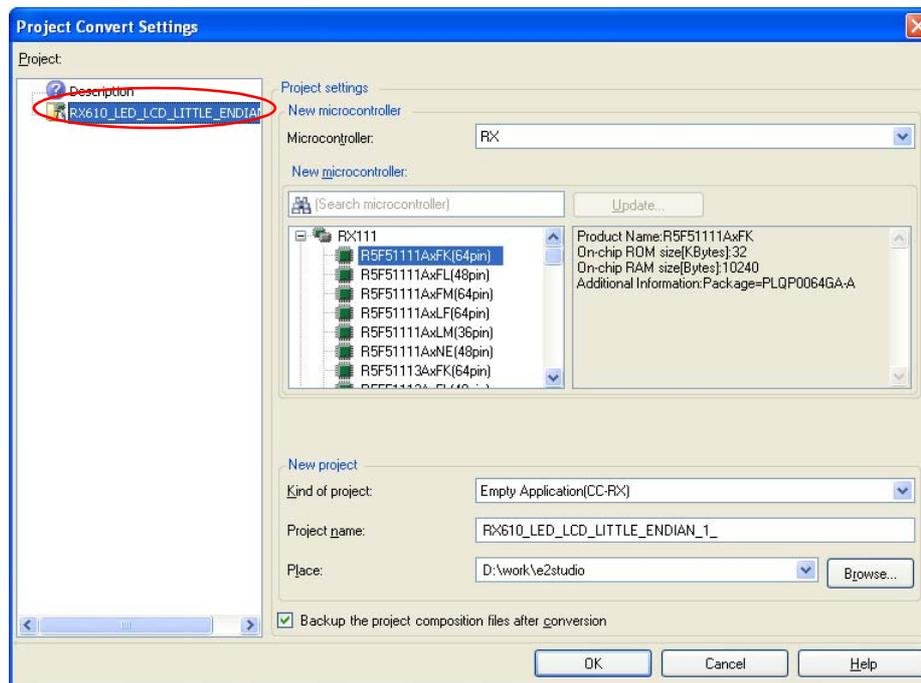


After setting the microcontroller to be used for the conversion target project, and project type, name, and place of creation settings, click the [OK] button.

> **Remarks 1.**   If you select [Backup the project composition files after conversion], immediately after the conversion the project source files and complete project are packed up and saved.
>
> **2.**   See "Project Convert Setting dialog box" for details about each setting item.

**(3)  Convert the CubeSuite+ project**
The e$^2$ studio project is converted to the CubeSuite+ project.

An e$^2$ studio project is converted to a CubeSuite+ project according to the rule below.
   - An e$^2$ studio project is converted to a project with the same name as the original project.  After conversion, the project file name will be "*project name*.mtpj".

> **Remarks 1.**   The target for conversion is the file structure information of the e$^2$ studio project file, configurations, build options, file exclusion, and link order information.
>    However, if a different build tool is used, build options (other than include paths and defined macros) are not converted.  In addition, if extensions of object files are different, the link order is not converted.
>
> **2.**   When build options have been added from the version of the target build tools in the e$^2$ studio 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 specified in the Project Convert Setting dialog box.
>
> **4.**   Configurations of the e$^2$ studio project are replaced with build modes of CubeSuite+ after conversion. Any characters that are not allowed in a CubeSuite+ folder name, however (\, /, :, *, ?, ", <, >, |) will be replaced by underscores ("_").
>    When the build mode names have 110 or more characters, they are rounded to 110 characters.

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.** If nesting is to a depth of 20 or more categories, the 20th stage and deeper stages are ignored. Files in such categories are added to the 20th category.

When the category names have 200 or more characters, they are rounded to 200 characters.

**6.** The following $e^2$ studio and build-tool versions are supported for conversion: $e^2$ studio V.2.0.0.16 or higher and CC-RX V1.00.00 or higher. The latest version of the compiler package installed on your computer is set as the version of your build tool.

**7.** The conversion result 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 to the project folder.
- The project convert information file is displayed under the File node 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 [Date]


(3)<Compiler package version>

    Compiler package used in e2 studio project: Version -> Compiler package used
in CubeSuite+ project: Version


(4)<Options not to use(Build mode)>

    Tool name of e2 studio (Tab name of build tool property in CubeSuite+)

     Option

        :


(5)<Options to change(Build mode)>

    Tool name of e2 studio (Tab name of build tool property in CubeSuite+)

     Option of e2 studio project -> Option of CubeSuite+ project
        :
```

| Number | Description |
|--------|-------------|
| (1) | Time and date on which a project was converted<br><br>The time and date on which an $e^2$ studio project was converted into a CubeSuite+ project is output using format "*dddd*, *mmmm dd*, *yyyy hh*:*mm*:*ss AM/PM*". |
| (2) | IDE version and date<br><br>The version of CubeSuite+ and the date are output. |
| (3) | Compiler package version<br><br>The compiler package used in an $e^2$ studio project and the version, and the compiler package used in a CubeSuite+ project and the version are output.<br><br>The version of the compiler package used in a CubeSuite+ projectis is the latest version in the compiler packages which are installed in the CubeSuite+ environment. |

| Number | Description |
|--------|-------------|
| (4) | Options not to use(*Build mode*)<br><br>If an option that has been set in an e$^2$ studio project and is not used in a CubeSuite+ project (option that has been deleted by upgrading the compiler package, and e$^2$ studio option) exists, the information is output  for each build mode in the format shown below.<br><br>   `Tool name(Tab name of build tool property in CubeSuite+)`<br>    `Option`<br>       `:`<br><br>- 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 e$^2$ studio provides by default. They differ depending on whether the debug information output option is set or not.).<br>- "Other Options" is output if *Tab name of build tool property in CubeSuite+* for *Tool name* does not exist. |
| (5) | Options to change(*Build mode*)<br><br>If an option that has been set in an e$^2$ studio 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.<br><br>   `Tool name(Tab name of build tool property in CubeSuite+)`<br>    `Option of e`$^2$ `studio project -> Option of CubeSuite+ project`<br>       `:`<br><br>- 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 e$^2$ studio provides by default. They differ depending on whether the debug information output option is set or not.).<br>- "Other Options" is output if *Tab name of build tool property in CubeSuite+* for *Tool name* does not exist. |

**2.7.10    Convert a CubeSuite project into a CubeSuite+ project**

You can convert a CubeSuite project into a CubeSuite+ project.  Select [Open Project...] from the [Project] menu with CubeSuite+, and then select a project file (*.cspj).

**(1) Select the settings for conversion**

When a CubeSuite project file is selected, the following message dialog box will open.

**Figure 2-69.   Message Dialog Box**
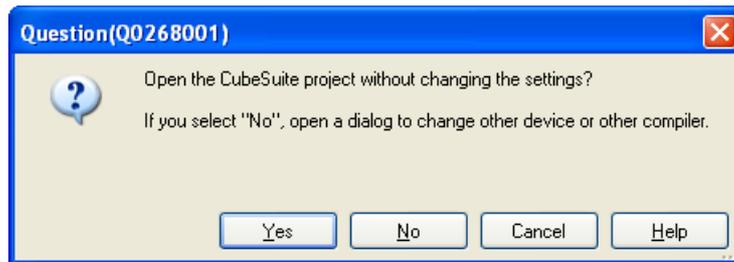


If you select the [Yes] button, the CubeSuite project settings are left unchanged, and the conversion to a CubeSuite+ project is carried out.

To change the microcontroller or project name, click the [No] button (continue to (2)).

**(2) Select the project for conversion settings**

The Project Convert Setting dialog box will open.

**Figure 2-70.   Project Convert Setting Dialog Box**
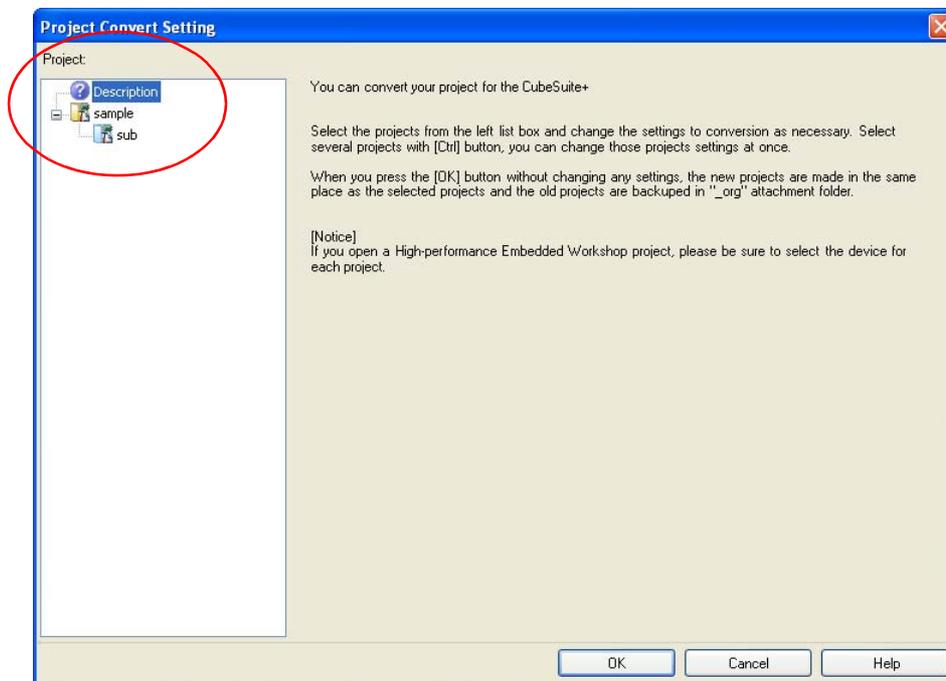


The [Project] area shows the configuration of conversion target projects as a tree; select the project for carrying out the conversion settings.

**(3)  Set the conversion target project**

When you select the project, the area on the right shows the conversion target project setting items.

**Figure 2-71.   Project Convert Setting Dialog Box (When Main Project Is Selected)**
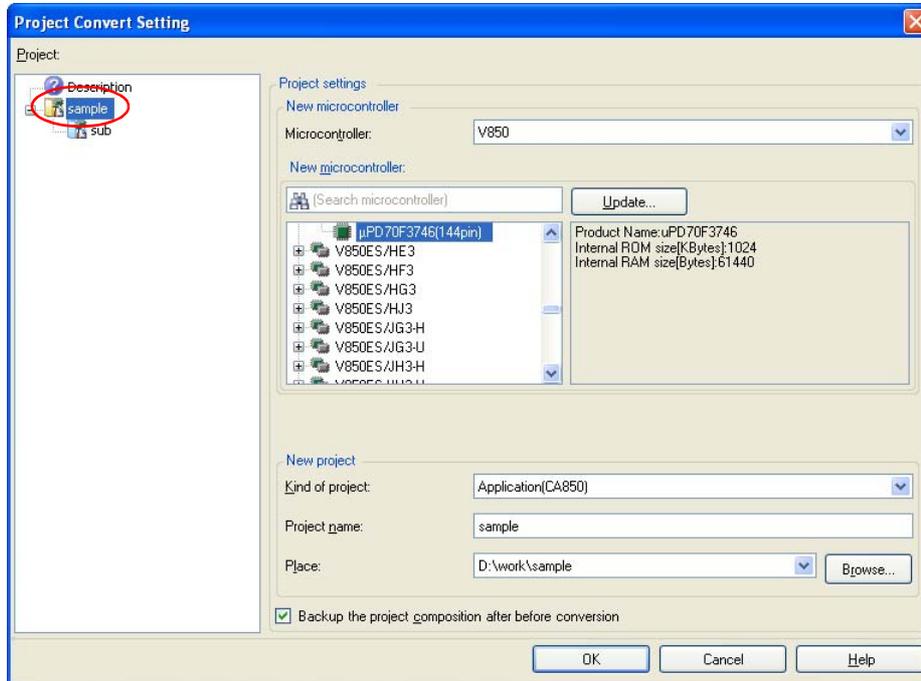


**Figure 2-72.   Project Convert Setting Dialog Box (When Subproject Is Selected)**
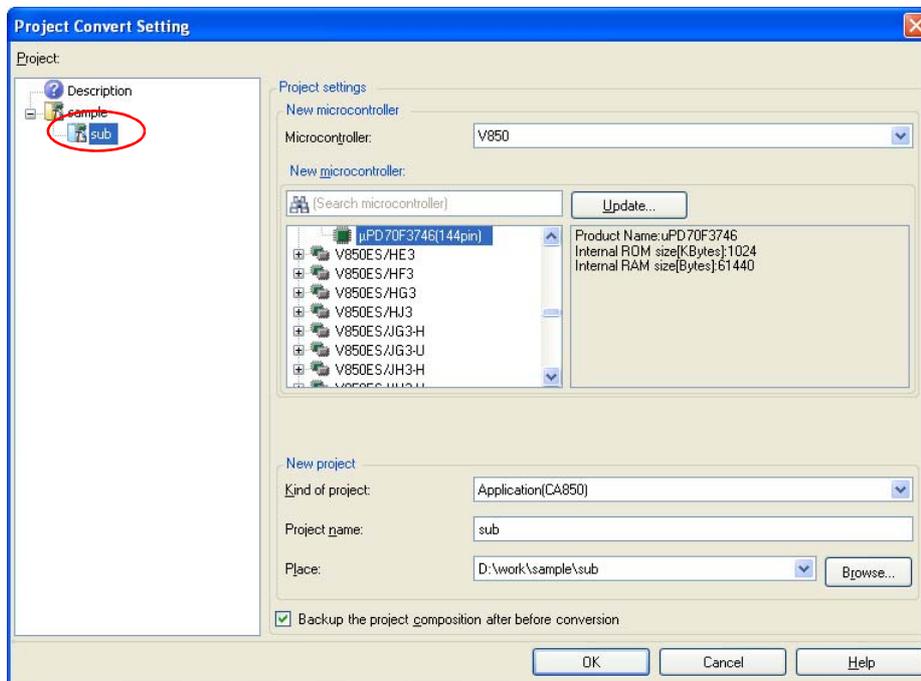


After setting the microcontroller to be used for the conversion target project, and project type, name, and place of creation settings, click the [OK] button.

Remarks 1.     When any subproject does not exist in a CubeSuite project, a subproject is not displayed.

2.     If you select [Backup the project composition files after conversion], immediately after the conversion the project source files and complete project are packed up and saved.

3.     See "Project Convert Setting dialog box" for details about each setting item.

**(4) Convert the CubeSuite+ project**
The CubeSuite project is converted to the CubeSuite+ project.

A CubeSuite project is converted to a CubeSuite+ project according to the rule below.
- A CubeSuite project is converted to a project with the same name as the original project. After conversion, the project file name will be "*project name*.mtpj".

Remarks 1.     The project file after conversion is created in the folder the CubeSuite 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.

2.     The build mode and build options information is not the target for conversion, if the build tool is different from the CubeSuite project file.

### 2.7.11 Convert a HEW project into a CubeSuite+ project

You can convert a High-performance Embedded Workshop (hereafter abbreviated "HEW") project into a CubeSuite+ project. Select [Open Project...] from the [Project] menu with CubeSuite+, and then select a workspace file (*.hws) or project file (*.hwp).

- When opening from a HEW workspace file (*.hws)
- When opening from a HEW project file (*.hwp)

**(1) When opening from a HEW workspace file (*.hws)**

**(a) Select the project for conversion settings**

When a HEW workspace file is selected, the Project Convert Setting dialog box will open.

**Figure 2-73. Project Convert Setting Dialog Box**
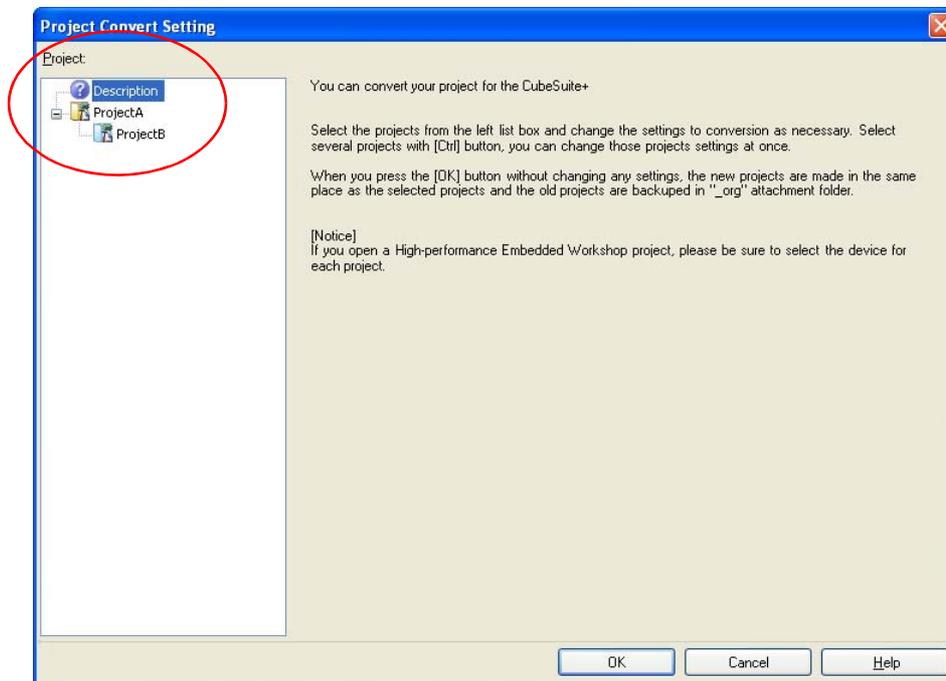


The [Project] area shows the configuration of conversion target projects as a tree; select the project for carrying out the conversion settings.

**(b) Set the conversion target project**

When you select the project, the area on the right shows the conversion target project setting items.

**Figure 2-74.   Project Convert Setting Dialog Box (When Main Project Is Selected)**
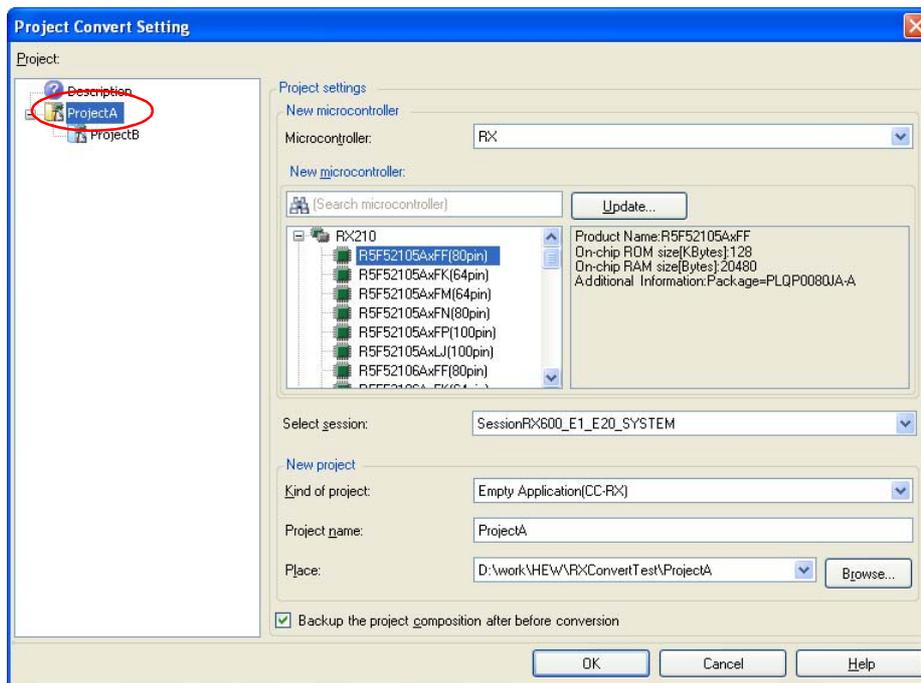


**Figure 2-75.   Project Convert Setting Dialog Box (When Subproject Is Selected)**
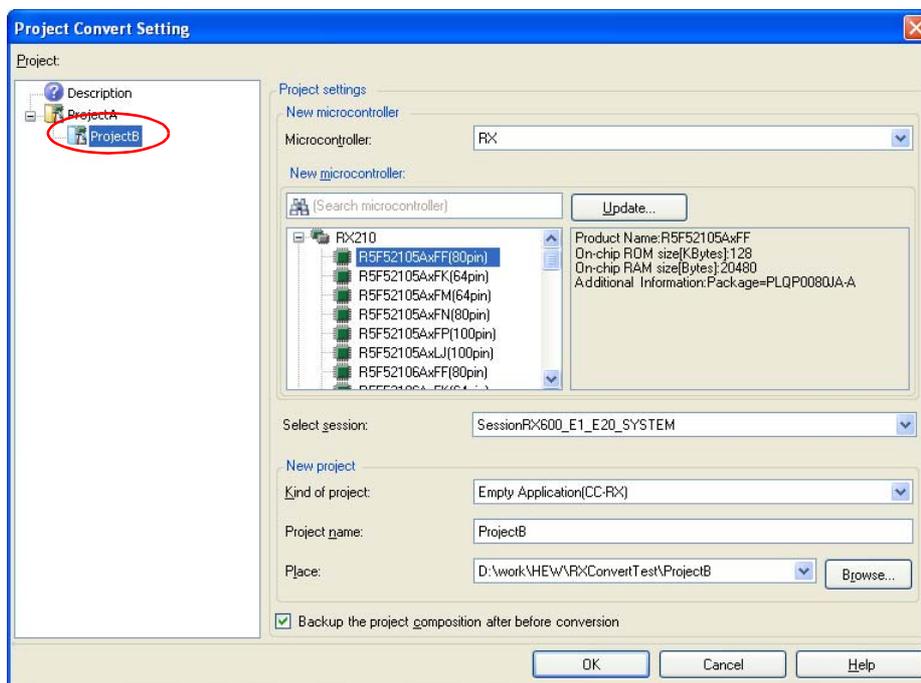


After setting the microcontroller to be used for the conversion target project, and project type, name, and place of creation settings, click the [OK] button.

**Cautions 1.    Confirm that the microcontroller of each project is selected before clicking the [OK] button.**

**2.**     **The CPU option and section (start) option are not changed in accordance with the selected device.  If the device that differs from the HEW project is selected, confirm (change) the CPU option and section (start) option after conversion.**

Remarks 1.    When only one project exists in a HEW project, a subproject is not displayed.
          2.    [Select session] is displayed only when multiple sessions exist in a project.
          3.    If you select [Backup the project composition files after conversion], immediately after the conversion the project source files and complete project are packed up and saved.
          4.    See "Project Convert Setting dialog box" for details about each setting item.

**(c)  Convert the CubeSuite+ project**
      The HEW project is converted to the CubeSuite+ project.

When opening from a HEW workspace file, the project is converted in accordance with the following rules.
   - The HEW workspace is not converted.
   - A HEW project will be converted into a main project or subproject, depending on the inter-project dependencies described in the workspace file.
       - If there are no inter-project dependencies, then the projects will be added to the project tree in the order they appear in the workspace file.
       - If there are inter-project dependencies, then the first project to appear that is not dependent on any other projects will be the main project.
       Subprojects are built in the order that they appear in the project tree.  Thus, projects are added to the project tree in the reverse of the dependency order.

Examples 1.    If projects A, B, C, D have no inter-dependencies:

```
HEW workspace before conversion                CubeSuite+ project after conversion

Workspace                                       Main project A
    ├── Project A                                   ├── Subproject B
    ├── Project B                                   ├── Subproject C
    ├── Project C                                   └── Subproject D
    └── Project D
```
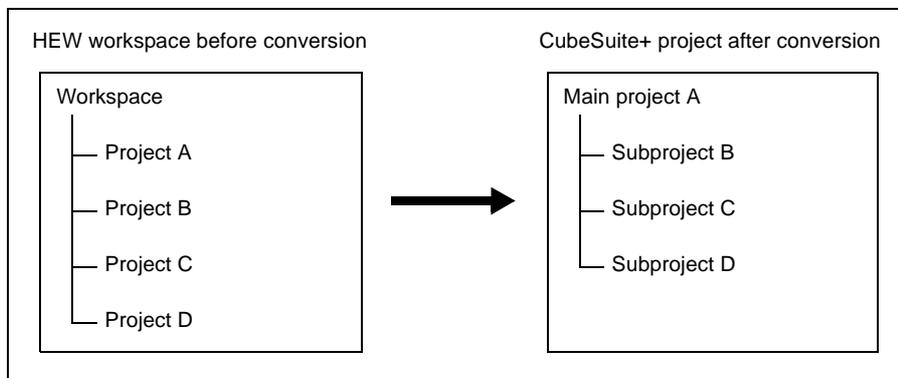
2.    If project C depends on project A, and projects B and D have no dependencies:

```
HEW workspace before conversion                CubeSuite+ project after conversion

Workspace                                       Main project B
    ├── Project A                                   ├── Subproject A
    ├── Project B                                   ├── Subproject C
    ├── Project C                                   └── Subproject D
    └── Project D
```
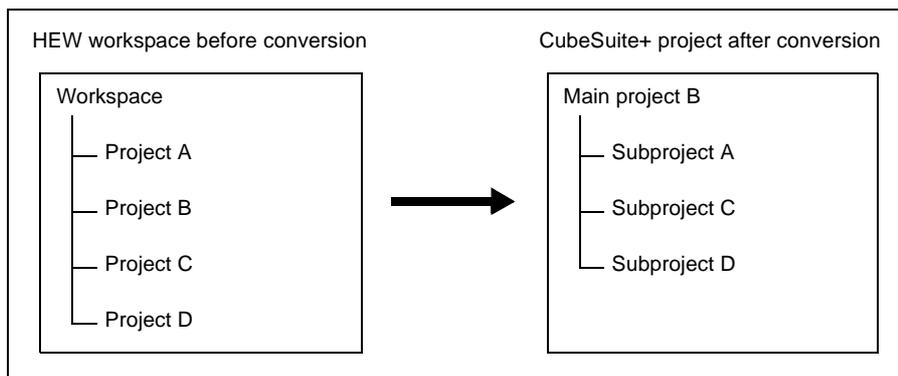
- HEW's placeholders which are used in build options are replaced with CubeSuite+'s placeholders.  However, the converted placeholder may not indicate the correct information because the concepts differs between HEW and CubeSuite+.  Change the build options, if necessary.
- After conversion, the main project file name will be "*project name*.mtpj" and the subproject file name will be "*project name*.mtsp".
- After conversion, the main project will be the active project.

**(2)   When opening from a HEW project file (*.hwp)**

**(a)   Select the project for conversion settings**
When a HEW project file is selected, the Project Convert Setting dialog box will open.

**Figure 2-76.   Project Convert Setting Dialog Box**
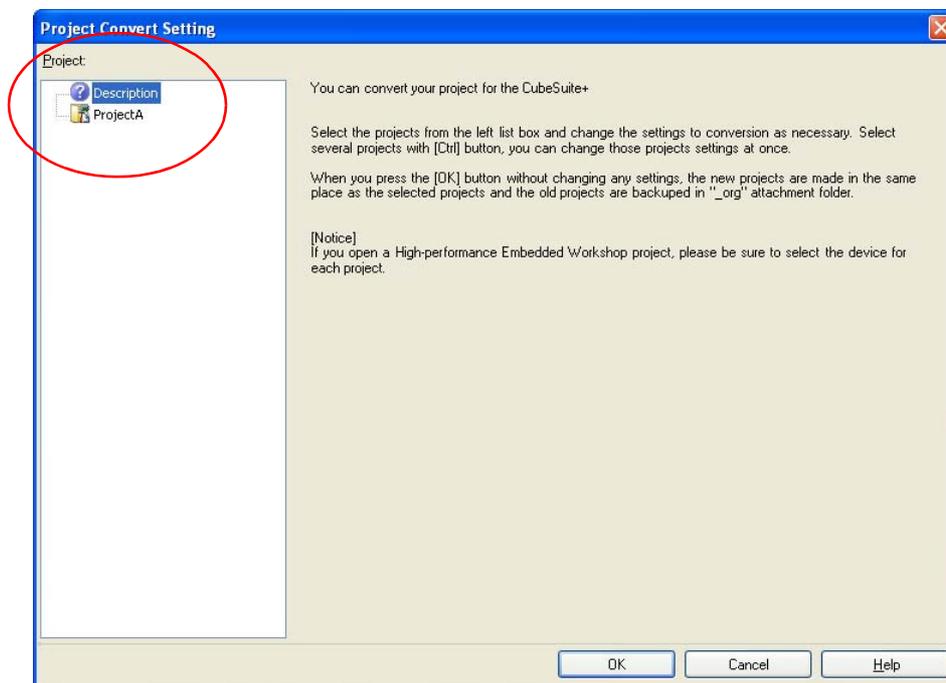


The [Project] area shows the configuration of conversion target projects as a tree; select the project for carrying out the conversion settings.

**(b)   Set the conversion target project**
When you select the project, the area on the right shows the conversion target project setting items.

**Figure 2-77.   Project Convert Setting Dialog Box (When Project Is Selected)**
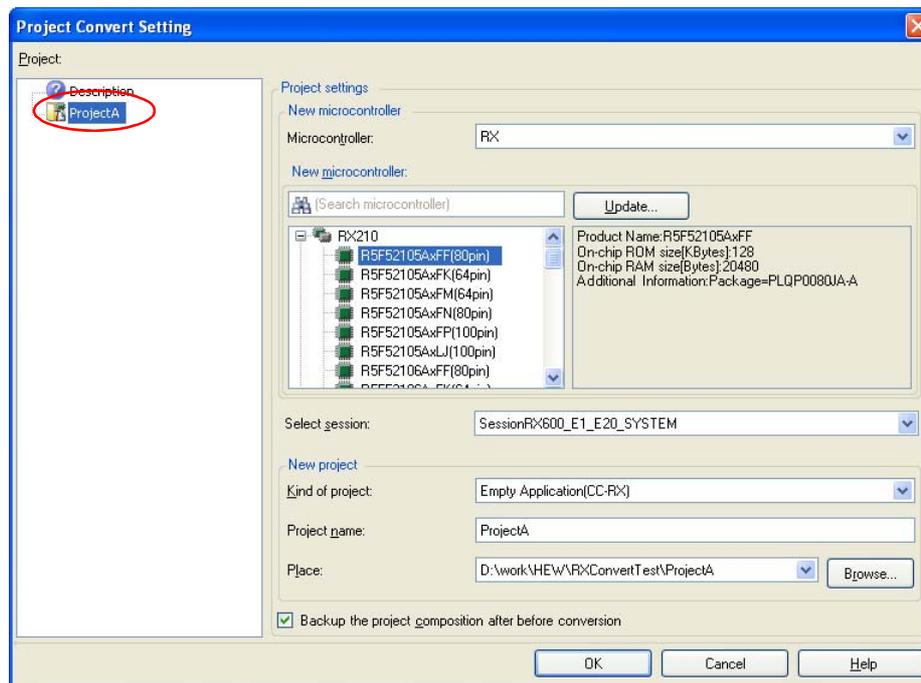


After setting the microcontroller to be used for the conversion target project, and project type, name, and place of creation settings, click the [OK] button.

**Caution     Confirm that the microcontroller is selected before clicking the [OK] button.**

**Remarks 1.**     [Select session] is displayed only when multiple sessions exist in a project.

**2.**     If you select [Backup the project composition files after conversion], immediately after the conversion the project source files and complete project are packed up and saved.

**3.**     See "Project Convert Setting dialog box" for details about each setting item.

**(c) Convert the CubeSuite+ project**
The HEW project is converted to the CubeSuite+ project.

When opening from a HEW project file, the project is converted in accordance with the following rules.
  - A HEW project is converted to a project with the same name as the original project.  After conversion, the project file name will be "*project name*.mtpj".

**Remarks 1.**     When conversion is performed from the HEW project for RX into the CubeSuite+ project for RX and from the HEW project for SuperH into the CubeSuite+ project for RH850, the target for conversion is the file structure information of the HEW project file, configurations, build options (only a part of options if the HEW project for SuperH is converted into the CubeSuite+ project for RH850), file exclusion, and link order information.
  - When build options have been added from the version of the target build tools in the HEW project to the version of the build tools after conversion, the added options are set to the build tools' default values after conversion.
  - Configurations of the HEW project are replaced with build modes of CubeSuite+ after conversion. Any characters that are not allowed in a CubeSuite+ folder name, however (\, /, :, *, ?, ", <, >, |) will be replaced by underscores ("_").

The build mode selected by default is determined by the project temporary file (*project-name*.tps) corresponding to the project.  However, if there is no project temporary file, the build mode will be "DefaultBuild".

If the build target file changed by Configurations(,etc.) in the HEW project files, these projects cannot be converted.

- When the link order has been specified in the HEW project, that setting is maintained. [RX]
  Since the available settings for the link order differ with the build mode, the order is converted according to the build mode.

  When the link order has not been specified in the HEW project, linkage proceeds in alphabetical order.

- When files are classified into folders in the tree view of a HEW project, the folders are replaced with the CubeSuite+ category after the project has been converted.

  If nesting is to a depth of 20 or more folders (categories), the 20th stage and deeper stages are ignored. Files in such folders are added to the 20th category.

2. Except when conversion is performed from the HEW project for RX into the CubeSuite+ project for RX and from the HEW project for SuperH into the CubeSuite+ project for RH850, the target for conversion is only the file structure information of the HEW project file.

3. The project files after conversion are created in the folder specified in the Project Convert Setting dialog box.

4. Conversion of projects that were created in HEW V.4.07 or a later version is supported.
   The supported versions of the build tools of HEW are as shown below.
   - SHC/C++ Ver.7.0 or higher
   - H8C/C++ Ver.6.0.00 or higher
   - NC30 V.5.20 Release 1 or higher
   - NC308 V.5.20 Release 1 or higher
   - NC100 V.1.01 Release 00 or higher
   - CCRX V.1.00 Release 00 or higher

5. The project which used the Custom Build Phase cannot be converted.

6. The target for conversion is only the HEW project file of the Renesas Electronics compiler.

7. The conversion result 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)<Options not to use(Build mode)>
    Tool name of HEW(Tab name of build tool property in CubeSuite+)
     Option
        :
    Tool name of HEW(Tab name of build tool property in CubeSuite+:file name)
     Option
        :


(2)<Options to change(Build mode)>
    Tool name of HEW (Tab name of build tool property in CubeSuite+)
```

```
    Option of HEW project -> Option of CubeSuite+ project
      :
   Tool name of HEW (Tab name of build tool property in CubeSuite+:file name)
    Option of HEW project -> Option of CubeSuite+ project
      :
```

| Number | Description |
|--------|-------------|
| (1) | Options not to use(*Build mode*) <br><br> If an option that has been set in a HEW project and is not used in a CubeSuite+ project (option that has been deleted by upgrading the compiler package, and HEW option) exists, the information is output  for each build mode in the format shown below. <br><br> ```Tool name(Tab name of build tool property in CubeSuite+)```<br>```  Option```<br>```        :``` <br><br> The information for individual compile options is output in the format shown below. <br><br> ```Tool name(Tab name of build tool property in CubeSuite+:file name)```<br>```  Option```<br>```          :``` <br><br> - 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 HEW provides by default. They differ depending on whether the debug information output option is set or not.). <br> - "Other Options" is output if *Tab name of build tool property in CubeSuite+* for *Tool name* does not exist. |
| (2) | Options to change(*Build mode*) <br><br> If an option that has been set in a HEW 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. <br><br> ```Tool name(Tab name of build tool property in CubeSuite+)```<br>```  Option of HEW project -> Option of CubeSuite+ project```<br>```        :``` <br><br> The information for individual compile options is output in the format shown below. <br><br> ```Tool name(Tab name of build tool property in CubeSuite+:file name)```<br>```  Option of HEW project -> Option of CubeSuite+ project```<br>```          :``` <br><br> - 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 HEW provides by default. They differ depending on whether the debug information output option is set or not.). <br> - "Other Options" is output if *Tab name of build tool property in CubeSuite+* for *Tool name* does not exist. |

**2.7.12      Convert a PM+ project into a CubeSuite+ project**

You can convert a PM+ project into a CubeSuite+ project. Select [Open Project...] from the [Project] menu with CubeSuite+, and then select a workspace file (*.prw) or project file (*.prj).

- When opening from a PM+ workspace file (*.prw)
- When opening from a PM+ project file (*.prj)

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

    **(a)    Select the project for conversion settings**

         When a PM+ workspace file is selected, the Project Convert Setting dialog box will open.

**Figure 2-78.    Project Convert Setting Dialog Box**
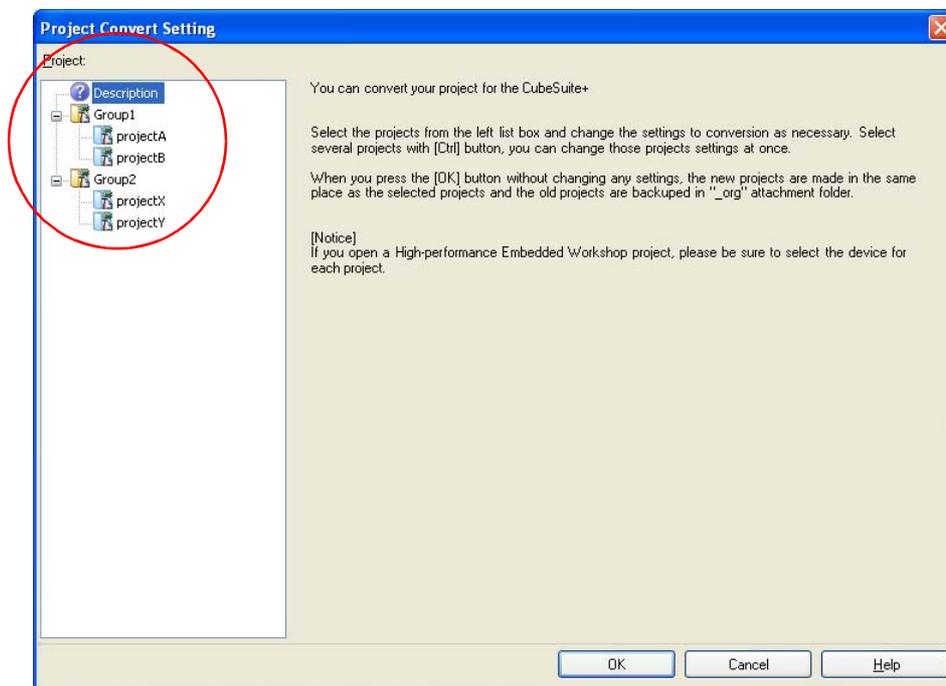


     The [Project] area shows the configuration of conversion target projects as a tree; select the project for carrying out the conversion settings.

    **(b)    Set the conversion target project**

         When you select the project, the area on the right shows the conversion target project setting items.

**Figure 2-79.   Project Convert Setting Dialog Box (When Main Project Is Selected)**
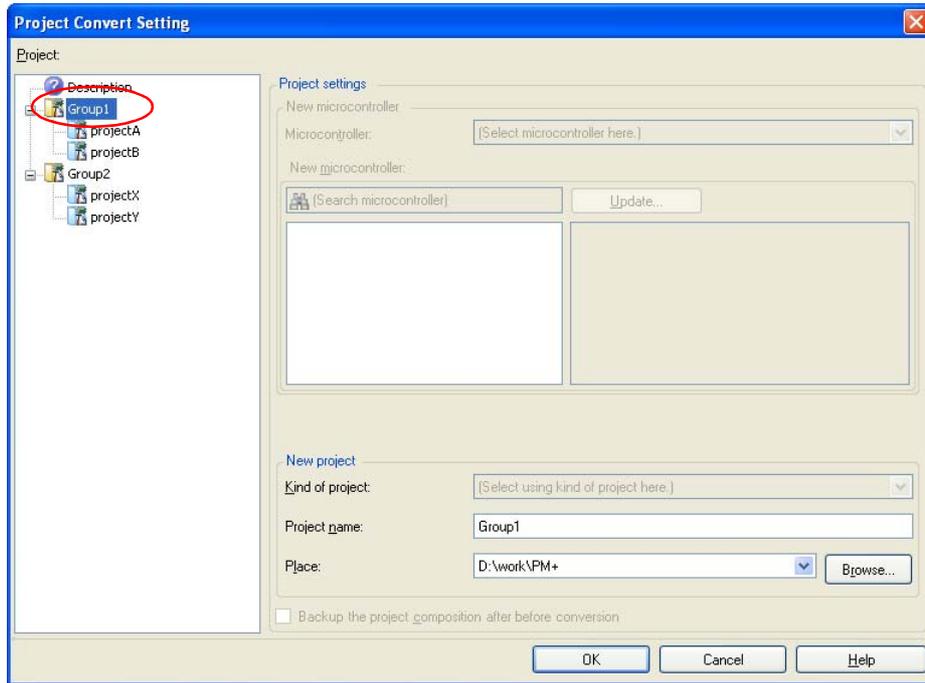


**Figure 2-80.   Project Convert Setting Dialog Box (When Subproject Is Selected)**
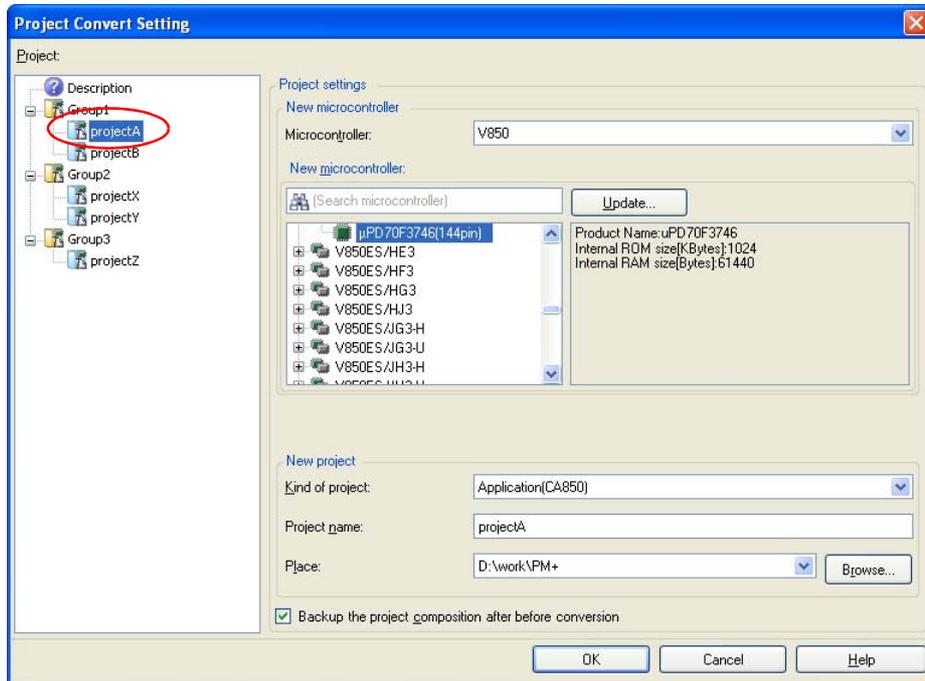


After setting the microcontroller to be used for the conversion target project, and project type, name, and place of creation settings, click the [OK] button.

**Remarks 1.**    When the main project is selected, the [New microcontroller] area and [Kind of project] in the [New project] area will be invalid.

**2.** If you select [Backup the project composition files after conversion], immediately after the conversion the project source files and complete project are packed up and saved.
However, the main project is selected, this item will be invalid.

**3.** See "Project Convert Setting dialog box" for details about each setting item.

**(c) Convert the CubeSuite+ project**

The PM+ project is converted to the CubeSuite+ project.

PM+ projects 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*.mtpj".
- 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.
- After conversion, the link order in PM+ is not held.

**Figure 2-81.   PM+ Project Conversion Image**
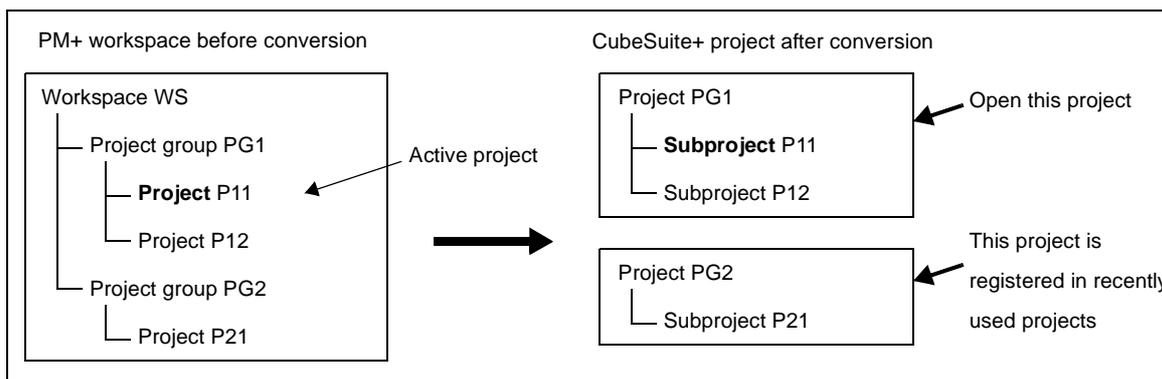


**Remarks 1.** After conversion, CubeSuite+ opens the project that was converted from the PM+ project group that includes the active project.  Projects other than this project are registered in recently used projects.  However, the number of projects that exceeds four is not registered.

**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.**

**3.** **When only one project exists in a PM+ workspace, the project will be converted to the main project.**

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

    **(a) Select the project for conversion settings**

        When a PM+ project file is selected, the Project Convert Setting dialog box will open.

**Figure 2-82. Project Convert Setting Dialog Box**
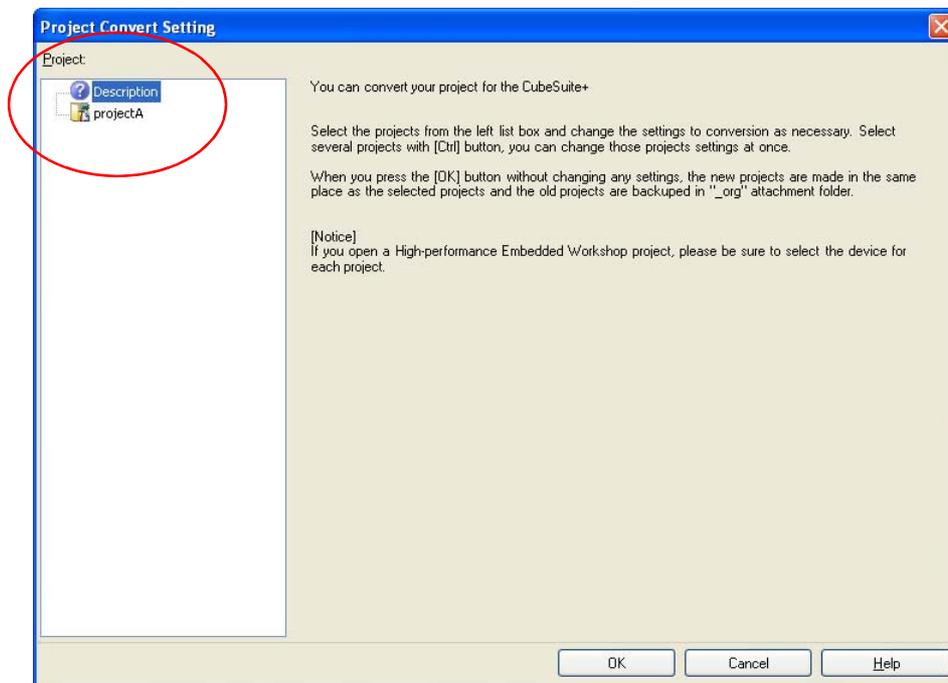


The [Project] area shows the configuration of conversion target projects as a tree; select the project for carrying out the conversion settings.

    **(b) Set the conversion target project**

        When you select the project, the area on the right shows the conversion target project setting items.

**Figure 2-83.   Project Convert Setting Dialog Box (When Project Is Selected)**
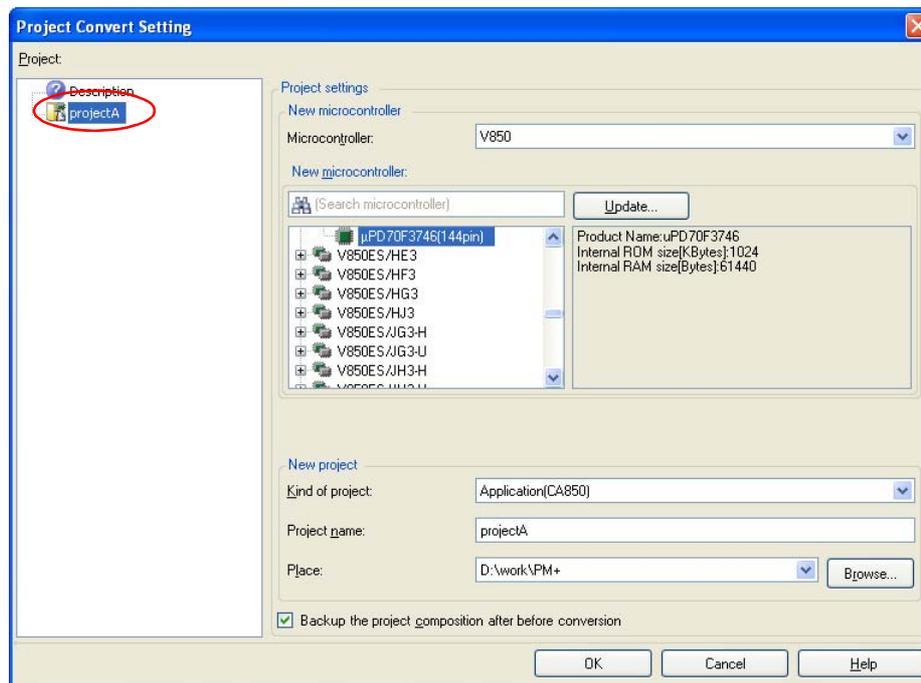


After setting the microcontroller to be used for the conversion target project, and project type, name, and place of creation settings, click the [OK] button.

**Remarks 1.**   If you select [Backup the project composition files after conversion], immediately after the conversion the project source files and complete project are packed up and saved.

**2.**   See "Project Convert Setting dialog box" for details about each setting item.

**(c)  Convert the CubeSuite+ project**

The PM+ project is converted to the CubeSuite+ project.

PM+ projects are converted to CubeSuite+ projects according to the rules below.

- A PM+ project is converted to a project with the same name as the original project.  After conversion, the project file name will be "*project name*.mtpj".

**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.

The build mode and build options information is not the target for conversion, if the build tool is different from the PM+ project file.

**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 conversion result 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 [Date]


(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) | Time and date on which a project was converted |
|  | The time and date on which a PM+ project was converted into a CubeSuite+ project is output using format "*dddd*, *mmmm dd*, *yyyy hh*:*mm*:*ss AM/PM*". |
| (2) | IDE version and date |
|  | The version of CubeSuite+ and the date are output. |
| (3) | Compiler package version |
|  | 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. |
|  | The version of the compiler package used in a CubeSuite+ projectis is the latest version in the compiler packages which are installed in the CubeSuite+ environment. |

| Number | Description |
|---|---|
| (4) | Options not to use(*Build mode*)<br><br>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.<br><br>`Tool name(Tab name of build tool property in CubeSuite+)`<br>` Option`<br>`      :`<br><br>- 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.).<br>- "Other Options" is output if *Tab name of build tool property in CubeSuite*+ for *Tool name* does not exist. |
| (5) | Options to change(*Build mode*)<br><br>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.<br><br>`Tool name(Tab name of build tool property in CubeSuite+)`<br>` Option of PM+ project -> Option of CubeSuite+ project`<br>`        :`<br><br>- 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.).<br>- "Other Options" is output if *Tab name of build tool property in CubeSuite*+ for *Tool name* does not exist. |

**2.7.13 Change the microcontroller**

You can change the microcontroller to be used in the created project.

**(1) Select the change for the microcontroller**

Select the Microcontroller node and select [Change microcontroller...] on the context menu.

The following message dialog box will open.

**Figure 2-84. Message Dialog Box**
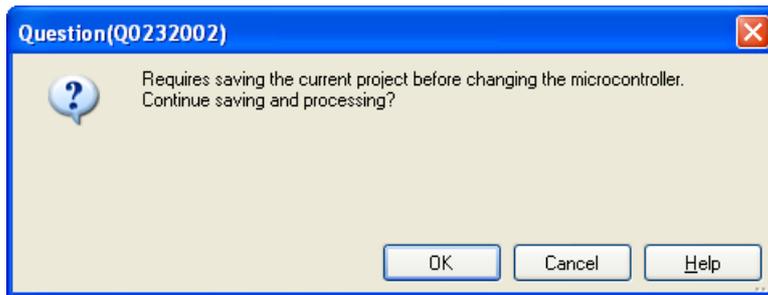


The project needs to be saved before making the following steps (The current project is overwritten by clicking the [OK] button).

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

**Remark** When multiple Microcontroller nodes for the projects using the same microcontroller are selected, the microcontroller settings can be changed at one time.

**(2) Select the new microcontroller**

The Change Microcontroller dialog box will open.

At this time, the current microcontroller is selected in the [Change microcontroller to] area.

**Figure 2-85. Change Microcontroller Dialog Box**
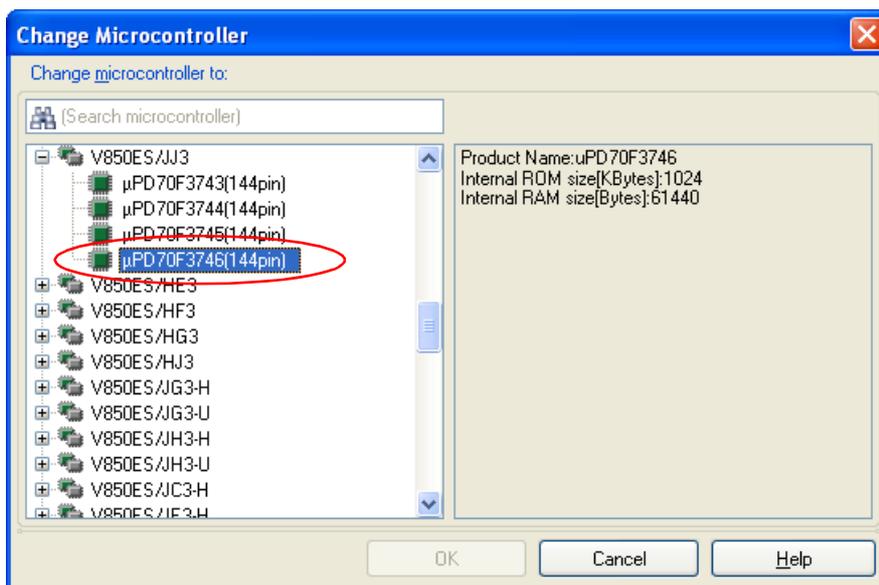


After selecting the new microcontroller, click the [OK] button.

**Remark**   The [OK] button becomes valid when the selected microcontroller differs from the current microcontroller.

**Figure 2-86.   Change Microcontroller Dialog Box (After Selecting New Microcontroller)**
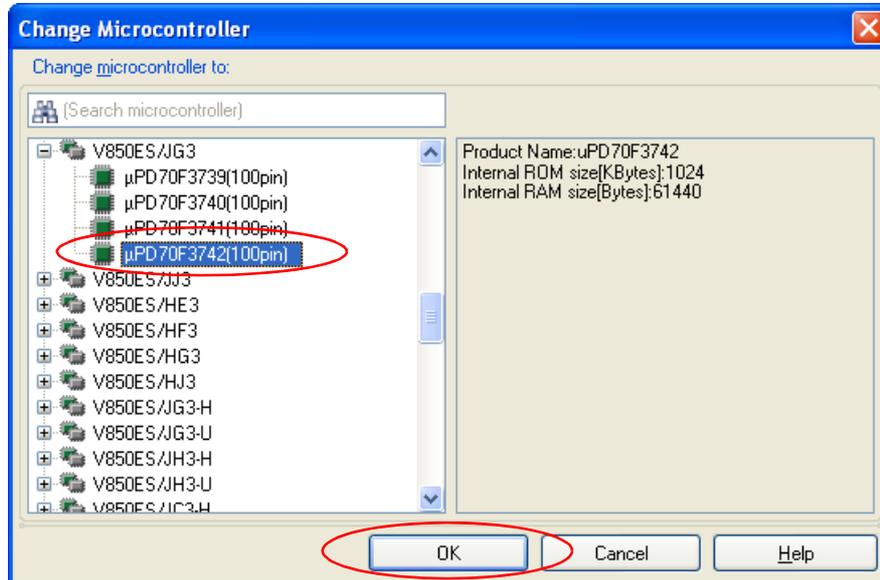


**(3)  Change the microcontroller**
    The current microcontroller is changed to the selected microcontroller.

**Remarks 1.**   The microcontroller can only be changed to another from the same family (RH850, RX, V850, R8C (Localised support), RL78, 78K0R, and 78K0) that is compatible with the same build tools.
    **2.**   When the microcontroller is changed, it must be disconnected from the debug tool.
    **3.**   When the microcontroller is changed, the project must be saved.
    **4.**   After the microcontroller is changed, information for the pin configurator (design tool), code generator (design tool), and debug tool (except for registration of the Watch panel) is not maintained.

## 2.8   Save the Project File

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

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

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

**Figure 2-87.   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 2-88.   Save Project As Dialog Box**
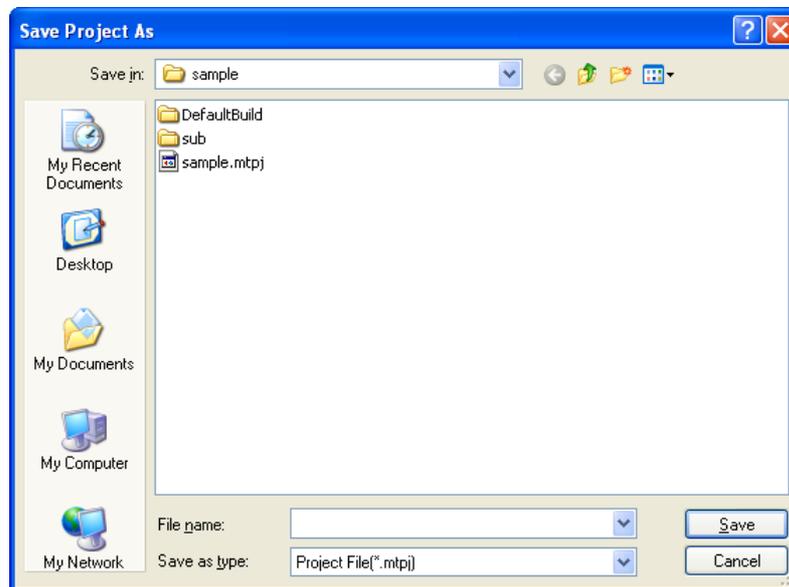


Specify the save folder and separate project filename (*.mtpj) on the dialog box.  The project file is saved in the speci-
fied folder with the filename.

**Cautions 1.    The files registered to the project are the same as those in the original project (the files regis-
tered 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.**

**Remarks 1.** 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.** To save the subproject files, new folders are created with the names "*new project name_n*" (*n*: 1, 2, ...) in the same folder as where the project file is to be saved. One subproject file is saved in one folder.

### 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 2-89.   [Save All] Item**
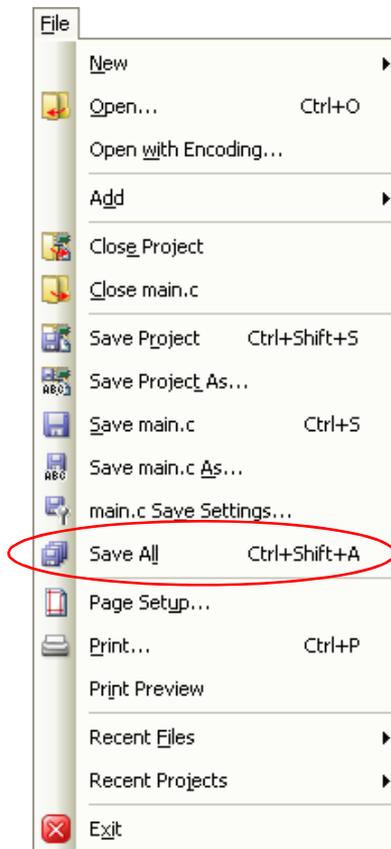


### 2.8.3    Pack and save the project and this product

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 Development Tools as Package] on the context menu.

If the project composition and properties have been changed, the message dialog box below will open.

**Figure 2-90.   Message Dialog Box**
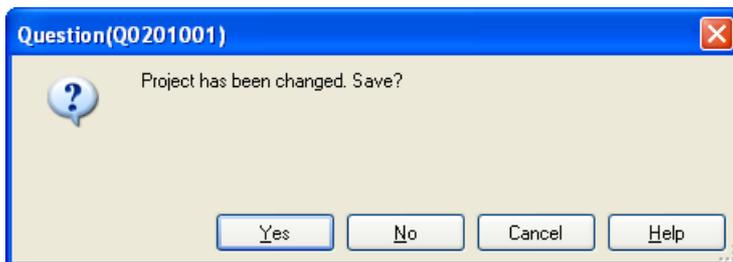


When you save the project, click the [Yes] button.  When you don't save the project, click the [No] button.
When files are being edited, the message dialog box below will open.

**Figure 2-91.   Message Dialog Box**
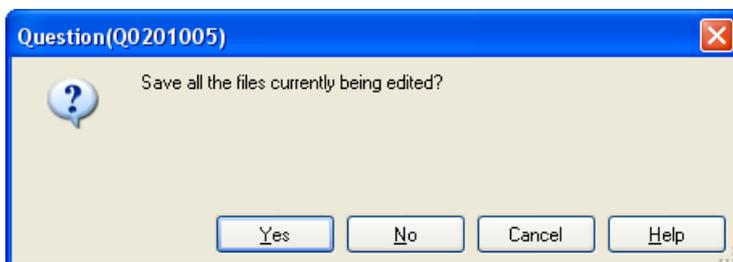


When you save the files, click the [Yes] button.  When you don't save the files, click the [No] button.
When using external tools such as an external text editor and files are being edited, the message dialog box below will open.

**Figure 2-92.   Message Dialog Box**
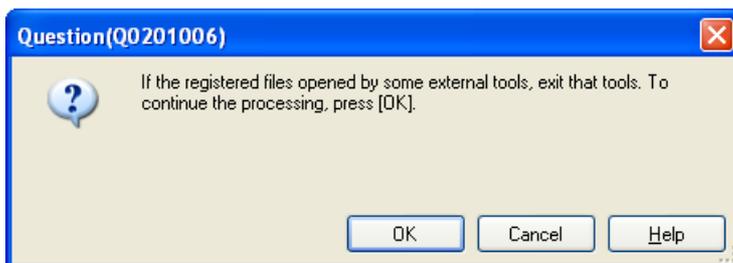


**Remark**    This dialog box only appears if in the Option dialog box, in the [General - External Text Editor] category, you selected the [Use external text editor] property.

To continue with the operation, click the [OK] button.
The Pack Settings dialog box will open.

**Figure 2-93. Pack Settings Dialog Box**
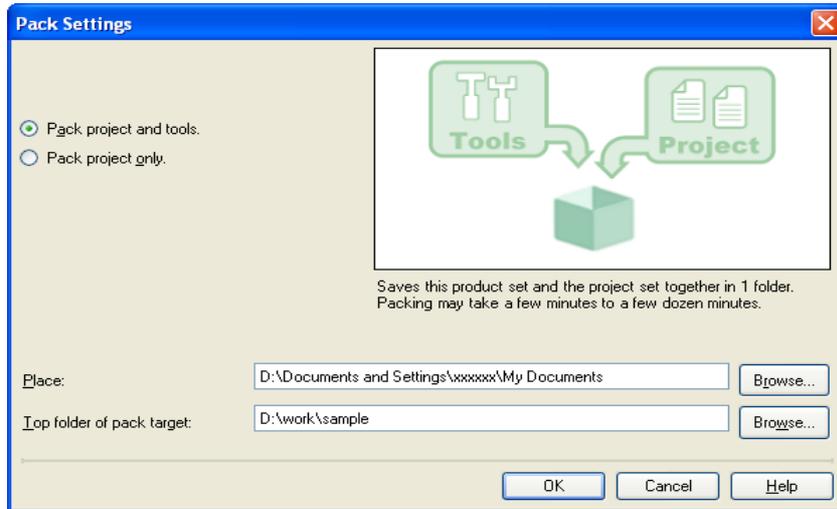


Specify the pack type (project and tools, or project only), save location folder,and the top folder of the packing target on the dialog box.

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

**Remark** Specify the top folder of the packing target to save with the project folder structure preserved.

**Figure 2-94. Sample of Project Pack**



Note src: The folder that source files are stored

inc: The folder that is specified as "Additional include paths"

lib: The folder that is specified as "Additional library paths"
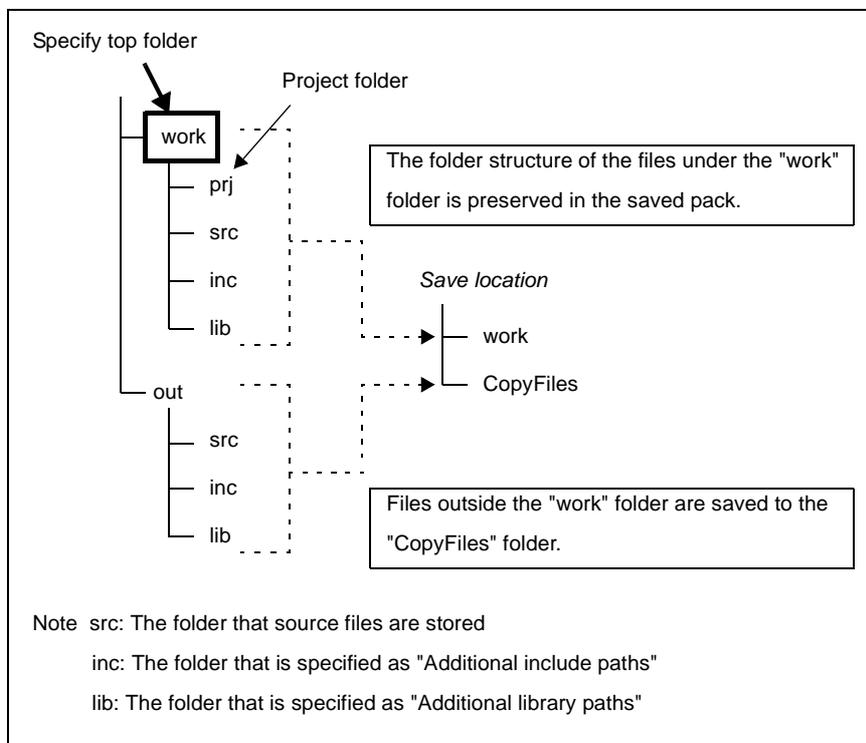
If you click the [OK] button, the copy process will begin.
The dialog box below is displayed during the copying process.

**Figure 2-95.   Copying**
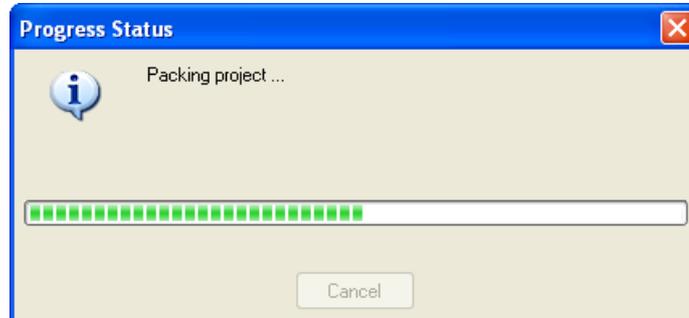


The save location folder structure is shown below.

**Figure 2-96.   Save Location Folder Structure**

```
Save location
    ── readme.txt        :  The explanation of the structure of the save location
    ── errorlog.txt      :  A record of the files that failed to copy
    ── Top folder name   :  The folder that is specified as the top folder of the packing targetNote 1
    ── Tools             :  The folder that this product is installedNote 2
    ── Copy_Files        :  The copy location folder for files not placed in the top folder of the packing targetNote 3
          ── copylog.txt :  A record of the copied files
          ── 1           :  The storage folder 1 for copied filesNote 4
          ── 2           :  The storage folder 2 for copied filesNote 4
             :
```

**Notes 1.** The following files in the top folder of the packing target specified in the Pack Settings dialog box are stored.
- All files in the project folder
- All files in the folders specified in the [Additional include paths] and [Additional library paths] properties of the build tool, and the C source file or assembler source file that is the build target
- Files added to the project
**2.** If you save the project only in the Pack Settings dialog box, this folder is not created.
**3.** Files that are not in the top folder of the packing target are stored in folders for each identical path.
**4.** 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.  **Files not placed inside the top folder of the packing target that is specified in the Pack Settings dialog box 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 2-97.   [Close Project] Item**
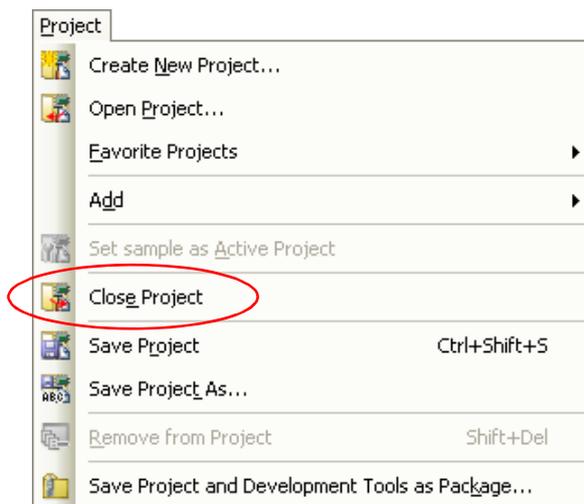


When the open project or changed files are not saved, a Message dialog box is displayed.

**Figure 2-98.   Message Dialog Box**
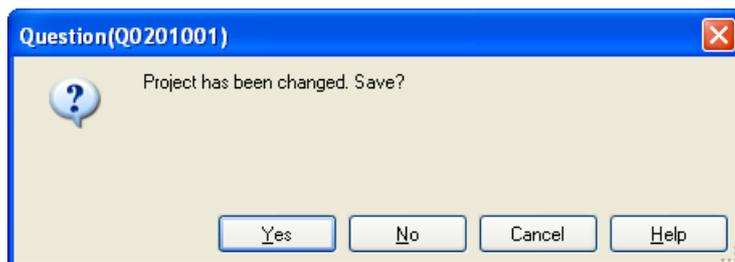


Click the [Yes] button to save, click the [No] button to not save.

### 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 2-99.   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 2-100.   [Reset Layout] Item**
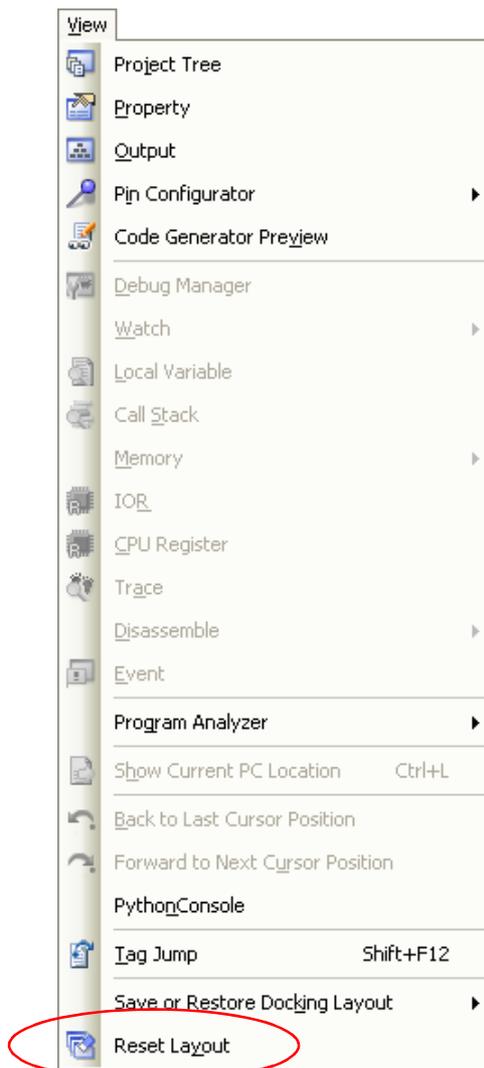
**2.10    Accelerate the Startup Time of CubeSuite+**

Use the following method to accelerate the startup time of CubeSuite+.
- Use rapid start
- Use acceleration utility

**2.10.1    Use rapid start**

The rapid start function makes CubeSuite+ permanently resident, to accelerate the startup time.

You can enable or disable the rapid start in the [General - Startup and Exit] category in the Option dialog box, with option [Reduce startup time by enabling rapid startup] (Default: enabled).

When the rapid start is enabled, after login to Windows, CubeSuite+ is in the standby state, and the 🔵 icon appears in the task tray.

**Figure 2-101.   Context Menu of Rapid Start Icon**



**(1)  Displaying the main window**

To release CubeSuite+ from the standby state, and display the Main window, select [Start] in the context menu of the 🔵 icon, or double-click the icon.

Click the ❌ button on the Main window to return CubeSuite+ to the standby state (an 🔵 icon is displayed in the task tray during standby).

**(2)  Exiting CubeSuite+**

To exit CubeSuite+, select [Exit] in the context menu.

### 2.10.2    Use acceleration utility

This product provides a utility to accelerate the startup time of CubeSuite+ when the rapid start is not used.

Run "AccelerationUtility.exe" (located in the same folder as the CubeSuite+ executable**Note**).  The following window will be opened.  Click the [Accelerate] button.

**Figure 2-102.   Acceleration Utility for CubeSuite+**
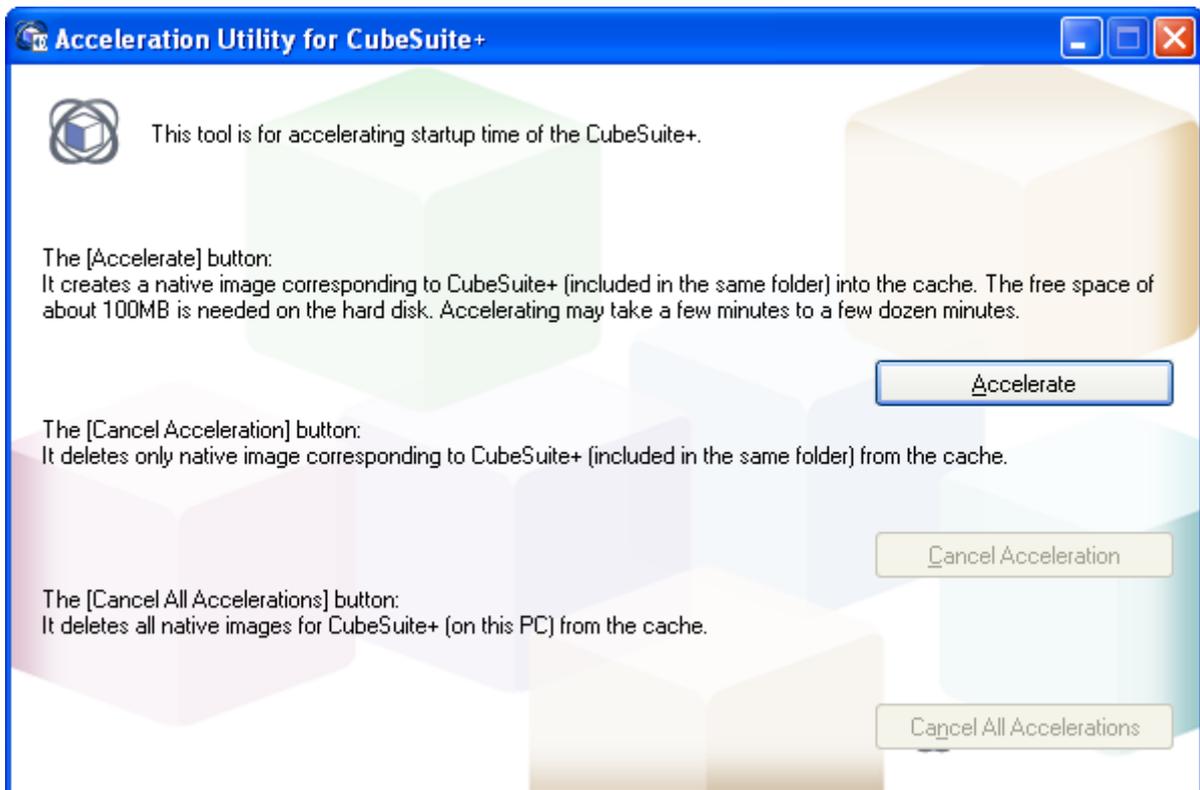


**Note**   The default installation folder of this product is as follows.
   C:\Program Files\Renesas Electronics\CubeSuite+


**Caution    The effectiveness of this utility will vary depending on your computer.**

### 2.11   Execute Python Fuctions

   CubeSuite+ enables the execution of IronPython functions and control statements, and CubeSuite+ Python functions
(see "G.3   CubeSuite+ Python Function/Class/Property/Event") 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 2-103.   Python Console Panel**
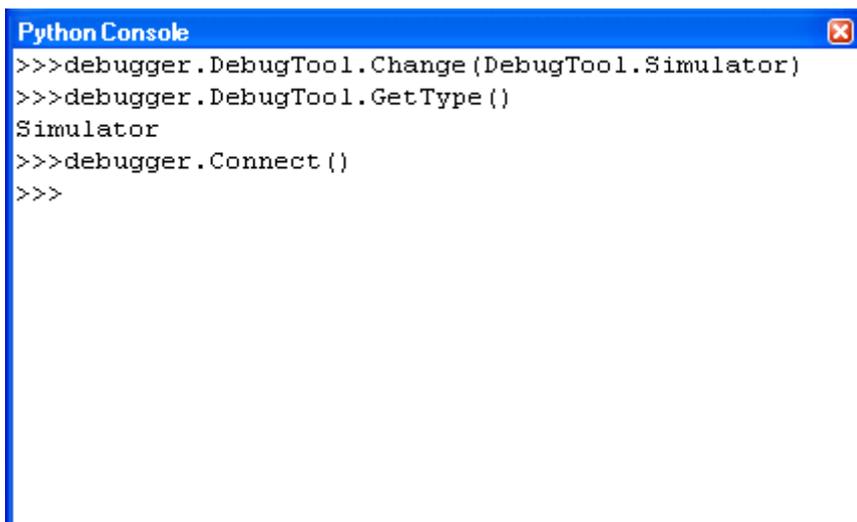


```
Python Console                                          ×
>>>debugger.DebugTool.Change(DebugTool.Simulator)
>>>debugger.DebugTool.GetType()
Simulator
>>>debugger.Connect()
>>>
```

**Remark**   See "APPENDIX G   Python CONSOLE/Python FUNCTIONS" for details about the Python console and
Python functions.

## 2.12 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 CubeSuiteW+.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[[/lpΔplug-in-name[,plug-in-name,...]]]Δ[/lpall]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.<br><br>Launch without displaying the Main window, build all the projects included in specified *project-file-name*, with the build mode specified by *build-mode-name*, and then exit.<br><br>If a project does not have the build mode specified by *build-mode-name*, then the build mode is copied based on the DefaultBuild, and the build is performed.<br><br>If the build mode specified by *build-mode-name* is not defined in the project specified by *project-file-name*, then an error will be displayed, and processing will end.<br><br>If *build-mode-name* is omitted, then the build will use the DefaultBuild.<br><br>If *project-file-name* is omitted, then an error will be displayed, and processing will end. |
| /brΔ[*build-mode-name*] | Execute a rebuild.<br><br>Launch without displaying the Main window, build all the projects included in specified *project-file-name*, with the build mode specified by *build-mode-name*, and then exit.<br><br>If a project does not have the build mode specified by *build-mode-name*, then the build mode is copied based on the DefaultBuild, and the build is performed.<br><br>If the build mode specified by *build-mode-name* is not defined in the project specified by *project-file-name*, then an error will be displayed, and processing will end.<br><br>If *build-mode-name* is omitted, then the build will use the DefaultBuild.<br><br>If *project-file-name* is omitted, then an error will be displayed, and processing will end. |

| Option | Description |
|---|---|
| /bcbΔ[*build-mode-name*] | Perform a clean and then execute a build. |
| | Launch without displaying the Main window, build all the projects included in specified *project-file-name*, with the build mode specified by *build-mode-name*, and then exit. |
| | If a project does not have the build mode specified by *build-mode-name*, then the build mode is copied based on the DefaultBuild, and the build is performed. |
| | If the build mode specified by *build-mode-name* is not defined in the project specified by *project-file-name*, then an error will be displayed, and processing will end. |
| | If *build-mode-name* is omitted, then the build will use the DefaultBuild. |
| | If *project-file-name* is omitted, then an error will be displayed, and processing will end. |
| /bcΔ[*build-mode-name*] | Perform a clean. |
| | Launch without displaying the Main window, build all the projects included in specified *project-file-name*, with the build mode specified by *build-mode-name*, and then exit. |
| | If a project does not have the build mode specified by *build-mode-name*, then the build mode is copied based on the DefaultBuild, and the build is performed. |
| | If the build mode specified by *build-mode-name* is not defined in the project specified by *project-file-name*, then an error will be displayed, and processing will end. |
| | If *build-mode-name* is omitted, then the build will use the DefaultBuild. |
| | If *project-file-name* is omitted, then an error will be displayed, and processing will end. |
| /lpΔ*plug-in-name*[,*plug-in-name*,...] | Start CubeSuite+ without displaying the Main window, and with loading the specified plugins in the Plugins folder. |
| | In *plug-in-name*, specify the name of the folder in which each DLL file is saved. Note that folder names are case-insensitive. |
| | You can specify multiple plugin names by separating them with commas. |
| | You can also specify the "/lp" option itself multiple times.  Each DLL will be loaded. |
| | When a plugin with the same name is specified for the /np option, the option specified backward of the command line will take precedence. |
| | If the plugin specified by this option does not exist, it will be ignored. |
| | If *project-file-name* is omitted, then an error will be displayed, and processing will end. |
| /lpall | Start CubeSuite+ without displaying the Main window, and with loading all plugins in the Plugins folder. |
| | If this is specified together with the "/np" or "/lp" option, then this option will take precedence.  When a plugin is specified together with the /npall option, the option specified backward of the command line will take precedence. |
| | If *project-file-name* is omitted, then an error will be displayed, and processing will end. |

| Option | Description |
|---|---|
| /np∆*plug-in-name*[,*plug-in-name*,...] | Start CubeSuite+ without displaying the Main window, and without loading the specified plugins in the Plugins folder.<br><br>In *plug-in-name*, specify the name of the folder in which each DLL file is saved. Note that folder names are case-insensitive.<br><br>You can specify multiple plugin names by separating them with commas.<br><br>You can also specify the "/np" option itself multiple times.  Each DLL will not be loaded.<br><br>When a plugin with the same name is specified for the /lp option, the option specified backward of the command line will take precedence.<br><br>If the plugin specified by this option does not exist, it will be ignored.<br><br>If *project-file-name* is omitted, then an error will be displayed, and processing will end. |
| /npall | Start CubeSuite+ without displaying the Main window, and without loading any of the specified plugins in the Plugins folder.<br><br>If this is specified together with the "/np" or "/lp" option, then this option will take precedence.  When a plugin is specified together with the /lpall option, the option specified backward of the command line will take precedence.<br><br>If *project-file-name* is omitted, then an error will be displayed, and processing will end. |
| *plug-in-option*∆[*plug-in-parameter*,...] | Specify an option for the plug-in (see "(3)  Plug-in Options"). |
| *project-file-name* | Start CubeSuite+ without displaying the Main window, with the specified project file loaded. |

**Remarks 1.** When the "/lp", "/lpall", "/np", or "/npall" option is specified, the settings of the Plug-in Manager dialog box are not changed.

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

The results of execution are output to the command prompt.

**Figure 2-104.   Command Prompt (If Build Is Executed with CubeSuite+.exe)**
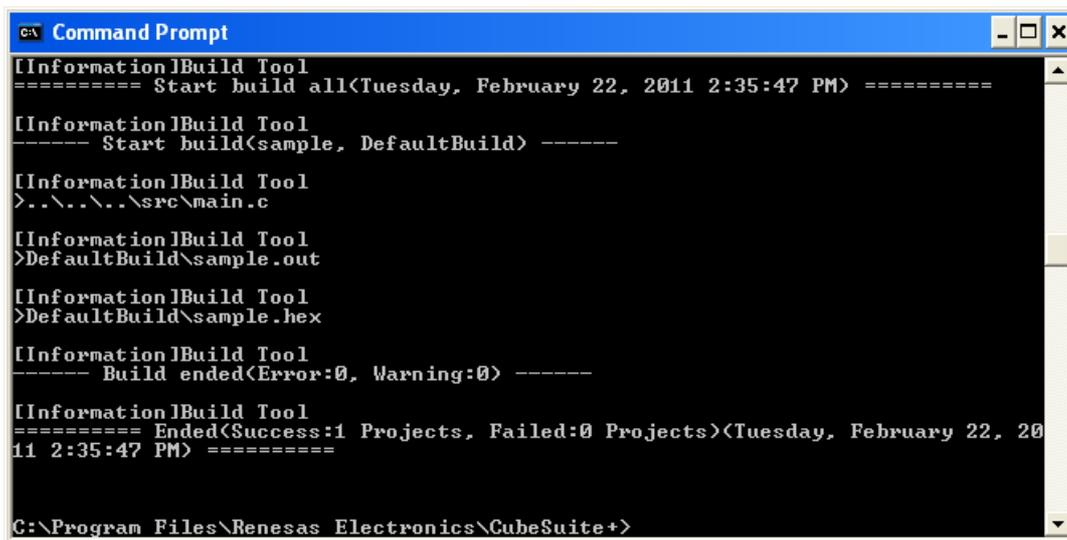
**(2)  When executing CubeSuiteW+.exe**

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[[/lpΔplug-in-
name[,plug-in-name,...]]]Δ[/lpall]Δ[/noproj]Δ[/nrs]Δ[plug-in-optionΔ[plug-in-parame-
ter,...]]][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. |
| /lpΔ*plug-in-name*[,*plug-in-name*,...] | Start CubeSuite+ displaying the Main window, with loading the specified plugins in the Plugins folder.<br>In *plug-in-name*, specify the name of the folder in which the DLL file is saved. Note that folder names are case-insensitive.<br>You can specify multiple plug-in names by separating them with commas.<br>You can also specify the "/lp" option itself multiple times. Each DLL will be loaded.<br>When a plugin with the same name is specified for the /np option, the option specified backward of the command line will take precedence.<br>If the plugin specified by this option does not exist, it will be ignored. |
| /lpall | Start CubeSuite+ displaying the Main window, with loading all plug-ins in the Plugins folder.<br>If an invalid option is specified, it will be ignored. |
| /npΔ*plug-in-name*[,*plug-in-name*,...] | Start CubeSuite+ displaying the Main window, without loading the specified plugins in the Plugins folder.<br>In *plug-in-name*, specify the name of the folder in which the DLL file is saved. Note that folder names are case-insensitive.<br>You can specify multiple plug-in names by separating them with commas.<br>You can also specify the "/np" option itself multiple times. Each DLL will not be loaded.<br>When a plugin with the same name is specified for the /lp option, the option specified backward of the command line will take precedence.<br>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.<br>If an invalid option is specified, it will be ignored. |
| /noproj | Start CubeSuite+ displaying the Main window without reading in a project (either a project specified on the command line, or the last used project). |
| /nrs | If CubeSuite+ is already started and in the standby state using rapid start (see "2.5 Start CubeSuite+"), ignore this and start another process. |
| *plug-in-option*Δ[*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. |

**Remark**   When the "/lp", "/lpall", "/np", or "/npall" option is specified, the settings of the Plug-in Manager dialog box are not changed.

**(3) Plug-in Options**

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

| Option | Description |
|---|---|
| / psΔscript-file-name | This is the Python Console plugin option. <br><br>After loading the project file in CubeSuite+, run the code in *script-file-name*. <br><br>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. |
| 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 Convert Setting dialog box | This dialog box includes the settings for conversion from e$^2$ studio and the old development environment (CubeSuite, HEW, PM+) project (or workspace) to a CubeSuite+ project. |
| 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 | This panel is used to display the message that is output from the build tool/debug tool/each plug-in or the result of the Find In Files with the Find and Replace dialog box. |
| Error List panel | This panel is used to list the information of messages output from the build tool (error information). |
| Change Microcontroller dialog box | This dialog box is used to change the microcontroller to be used in the project. |
| 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. |
| Text Edit dialog box | This dialog box is used to input and edit texts in multiple lines. |

| Window/Panel/Dialog Box Name | Function Description |
|---|---|
| Encoding dialog box | This dialog box is used to select a file-encoding. |
| Bookmarks dialog box | This dialog box is used to display the position where a bookmark is to be set or to delete a bookmark. |
| Go to Line dialog box | This dialog box is used to move the caret to a specified source line. |
| Jump to Function dialog box [CC-RH][CC-RX][CX][NC30] | This dialog box is used to select a function to be jumped if there are some functions with the same names when a program jumps to the function specified on the Editor panel. |
| Find and Replace dialog box | This dialog box is used to find and replace the designated characters. |
| 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. |
| Print Preview window | This window is used to preview the file currently being displayed in the Editor panel before printing. |
| Dependent Projects Settings dialog box | This dialog box is used to reference/set the dependent projects. |
| Build Mode Settings dialog box | This dialog box is used to add and delete build modes and configure the current build mode in batch. |
| Character String Input dialog box | This dialog box is used to input and edit characters in one line. |
| 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. |
| 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. |
| Pack Settings dialog box | This dialog box is used to pack and save the project and this product. |
| Option dialog box | This dialog box is used to configure the CubeSuite+ environment. |
| Plug-in Manager dialog box | This dialog box is used to set the plug-ins to be read in when this product is started. |
| 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. |
| Detail Version Information dialog box | This dialog box is used to display detail version information of this product and the present project information. |
| 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. |
| Open File dialog box | This dialog box is used to open a file. |

| Window/Panel/Dialog Box Name | Function Description |
|---|---|
| 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. |
| Open Option Setting File dialog box | This dialog box is used to select an option setting file to import to the [General - Font and Color] category of the Option dialog box. |
| Save Option Setting File dialog box | This dialog box is used to save the setting of the [General - Font and Color] category of the Option dialog box to an option setting file. |
| 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. |
| Select Script File dialog box | This dialog box is used to select the script file for the Python console. |
| 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**
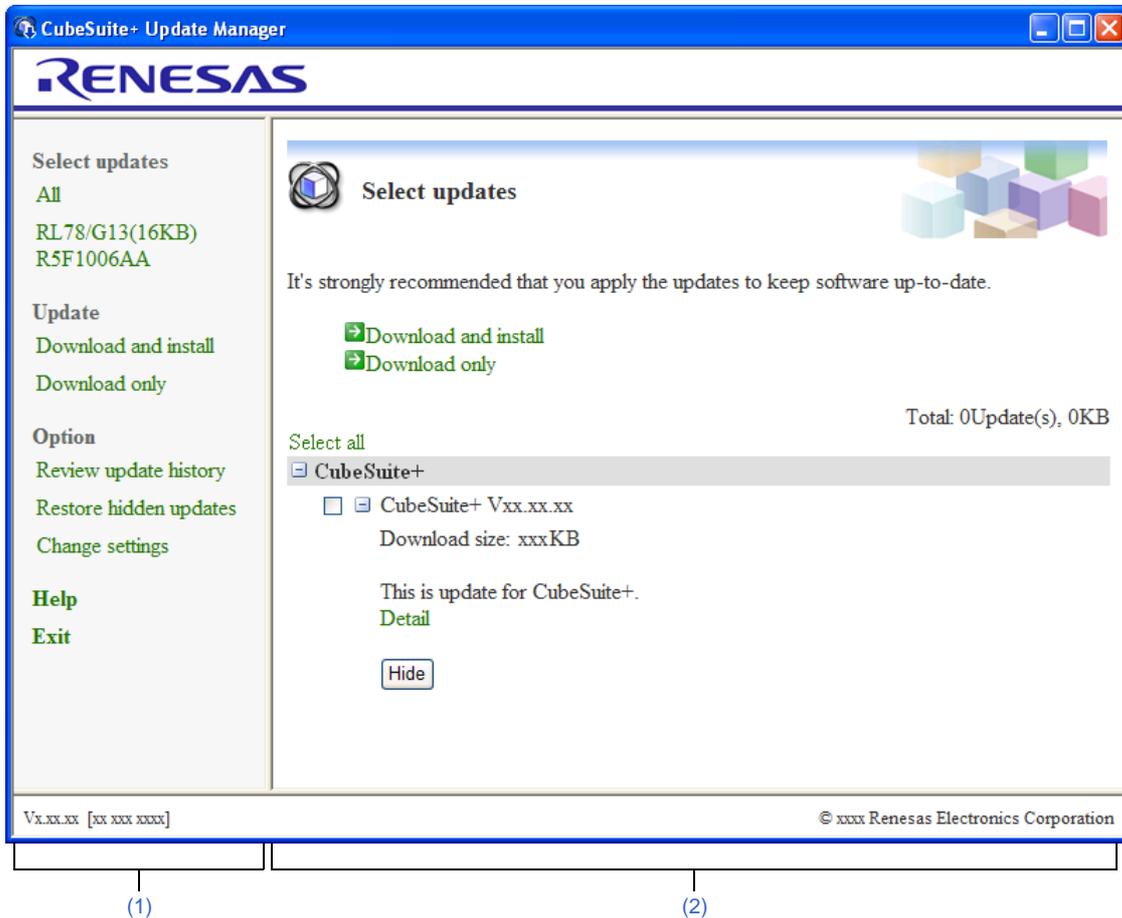


(1)    (2)

The following items are explained here.

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

**[How to open]**

- From the Windows [Start] menu, select [Programs] >> [Renesas Electronics CubeSuite+] >> [Update Manager].

    **Remark**    In Windows 8 or Windows 8.1, double-click on [Update Manager] on the start screen.

- From the CubeSuite+ [Help] menu, select [Check for Updates...].
- In the Windows task tray, click the 🔄 or ℹ️ 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. |
|---|---|
| *Microcontroller Name* | 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**
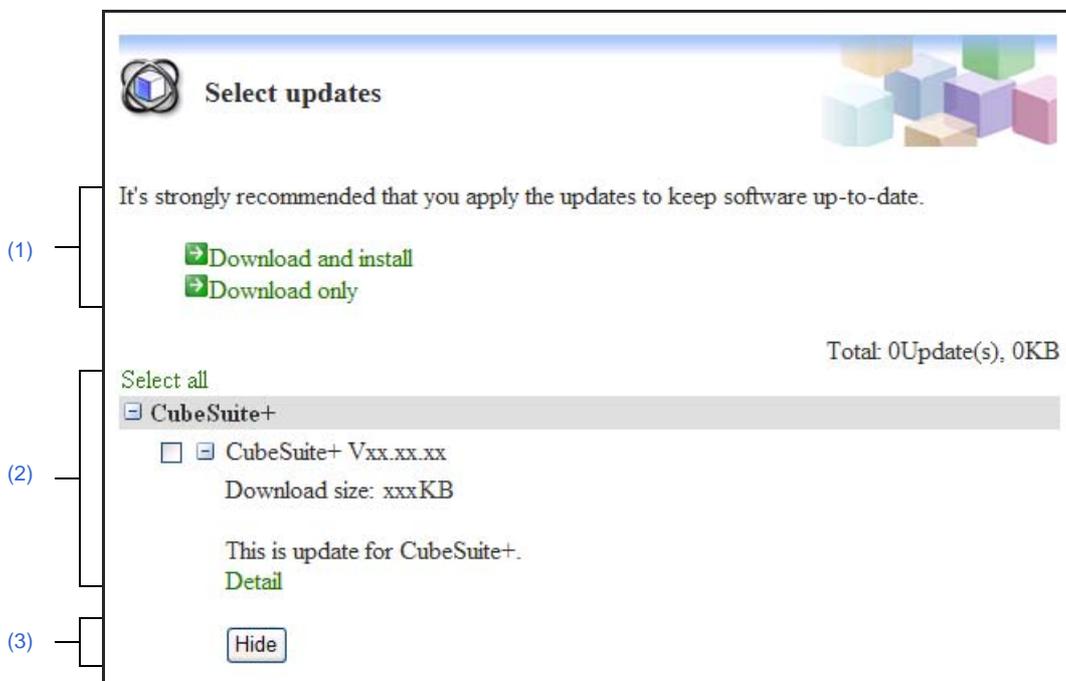


The following items are explained here.

  - [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.
   Click [Select all] to select the check boxes of all updates.
   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**
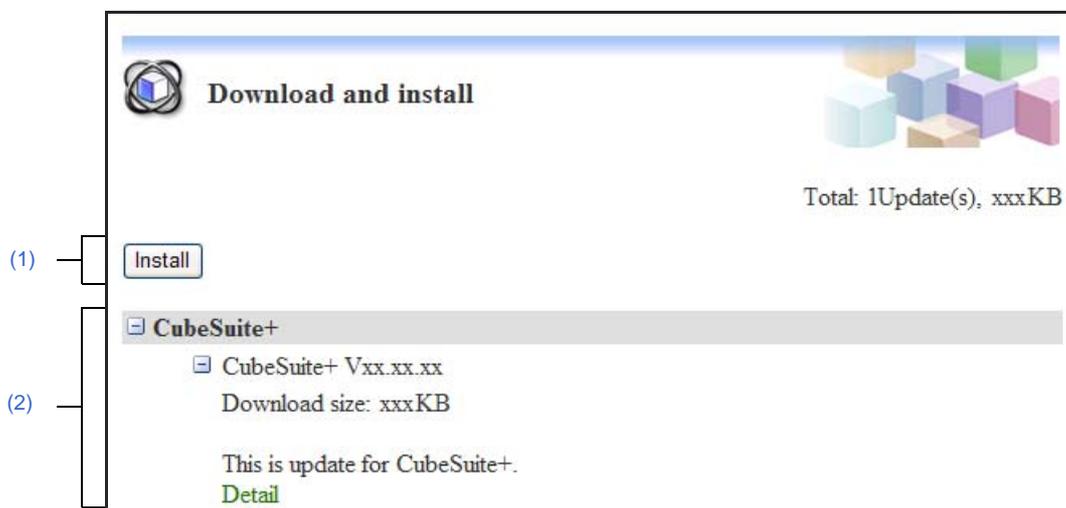


The following items are explained here.
- [How to open]
- [Description of each area]


**[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 an emulator USB driver is installed on Windows Vista, Windows 7, Windows 8, or Windows 8.1, the warning dialog box may appear.**


**(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**
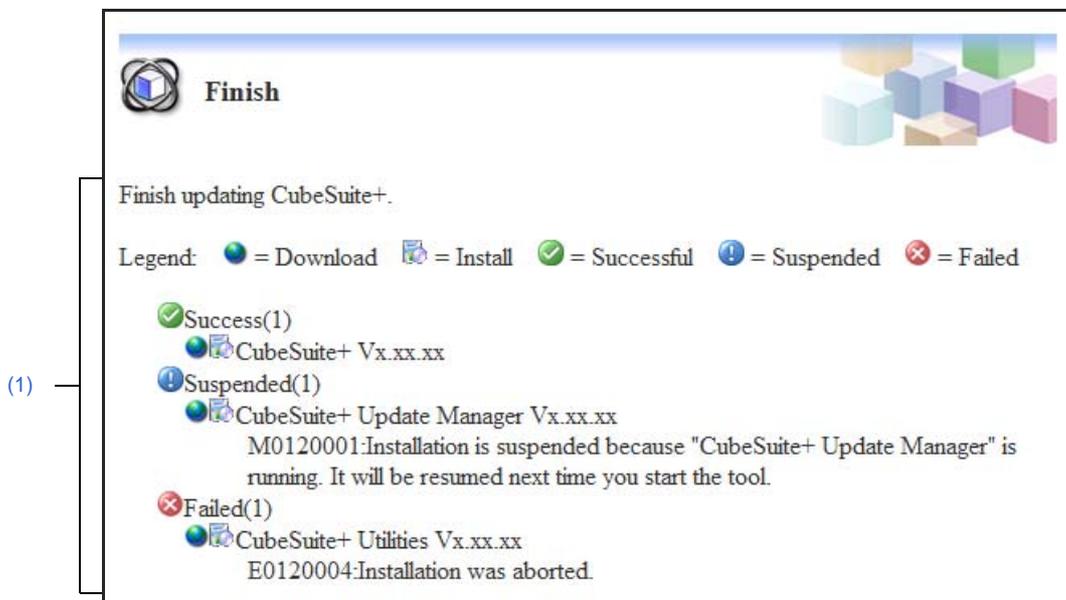


The following items are explained here.
- [How to open]
- [Description of each area]

**[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**
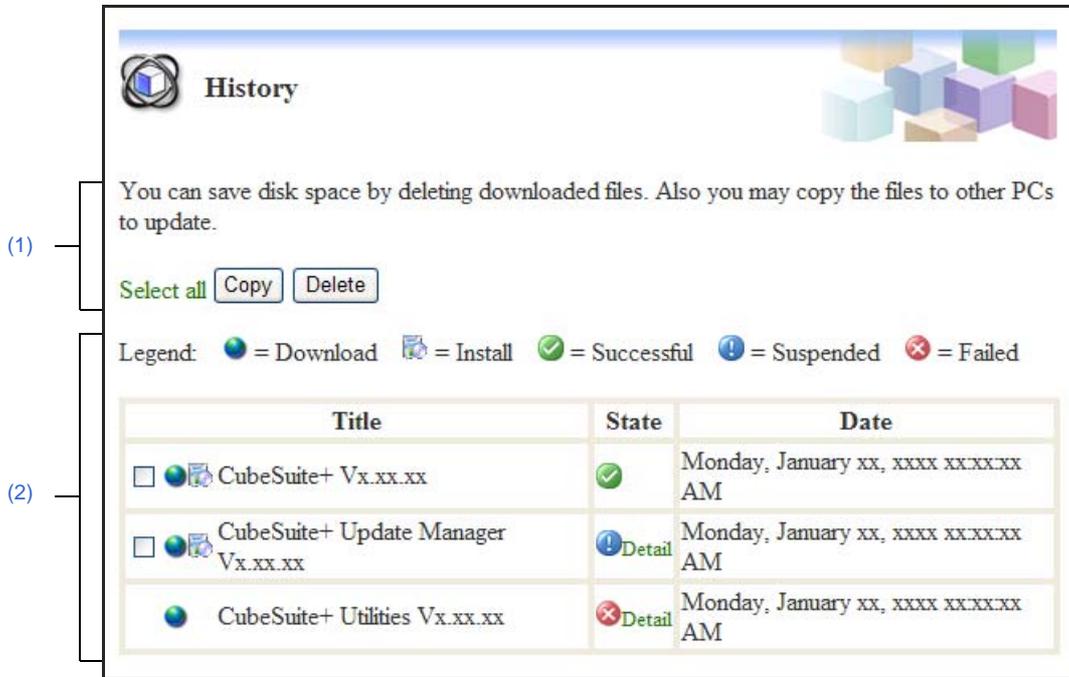


The following items are explained here.

- [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)  Behavior Selection area**

| Select all | Selects all the check boxes in the Update History Display area. |
|---|---|
| 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**
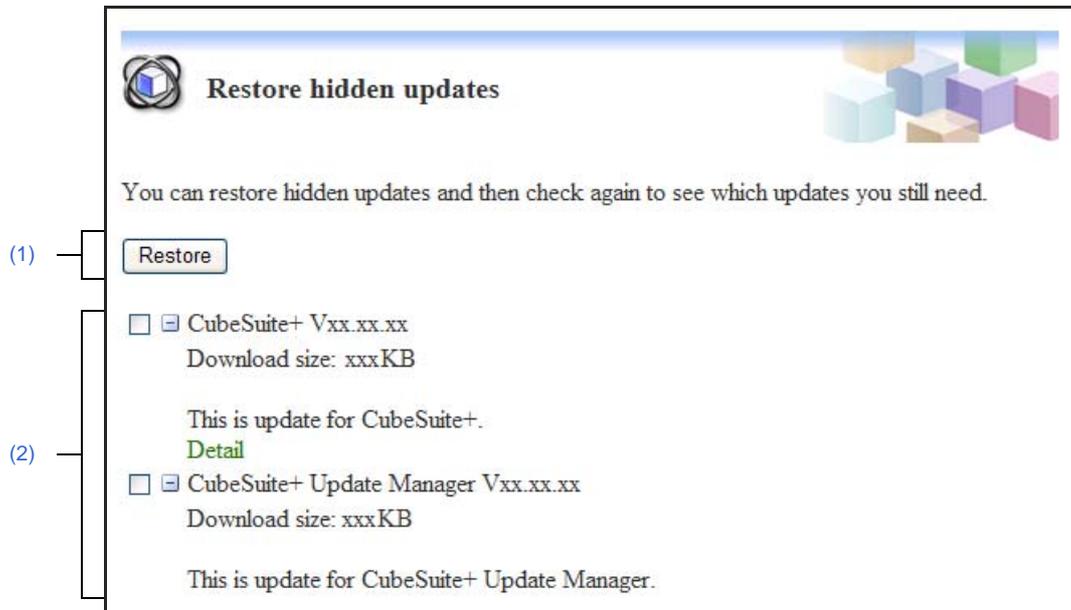


The following items are explained here.

- [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**
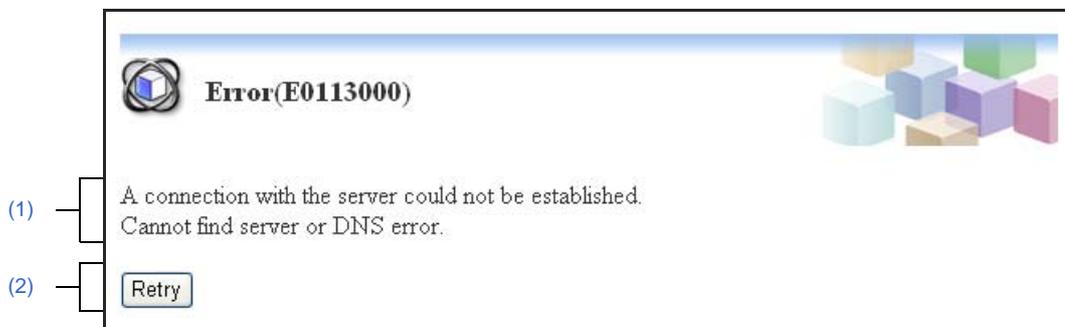


The following items are explained here.
- [How to open]
- [Description of each area]

**[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**
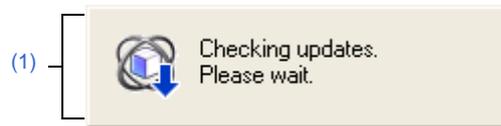
(1) ──

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**
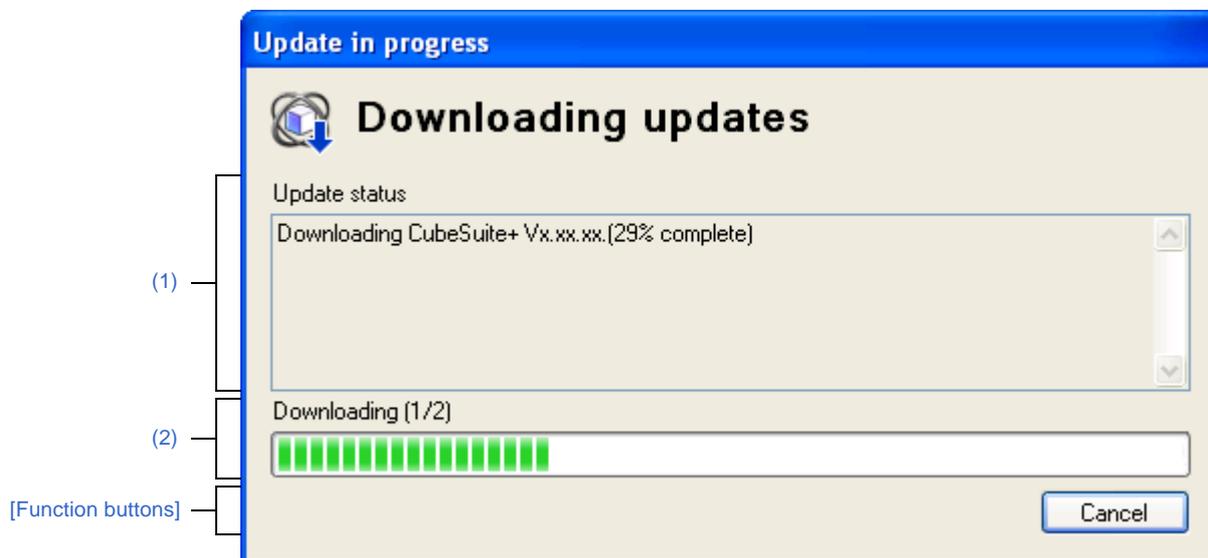


(1)

(2)

[Function buttons]

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**
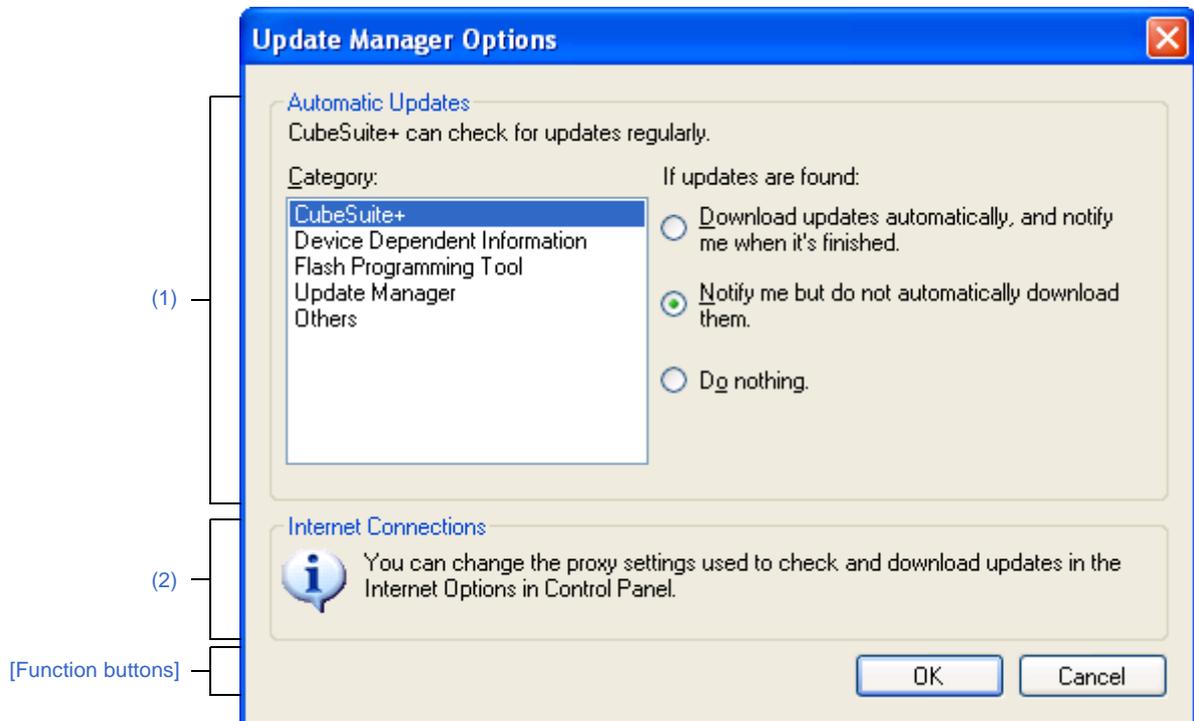


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[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**
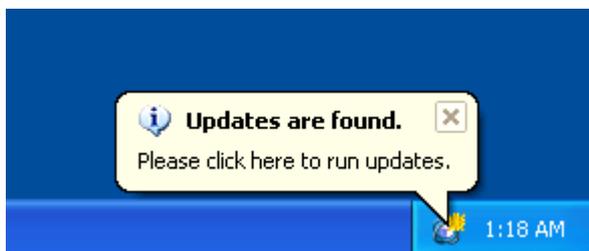


The following items are explained here.

- [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. |

---

License Manager window

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

**Figure A-13.   License Manager Window**
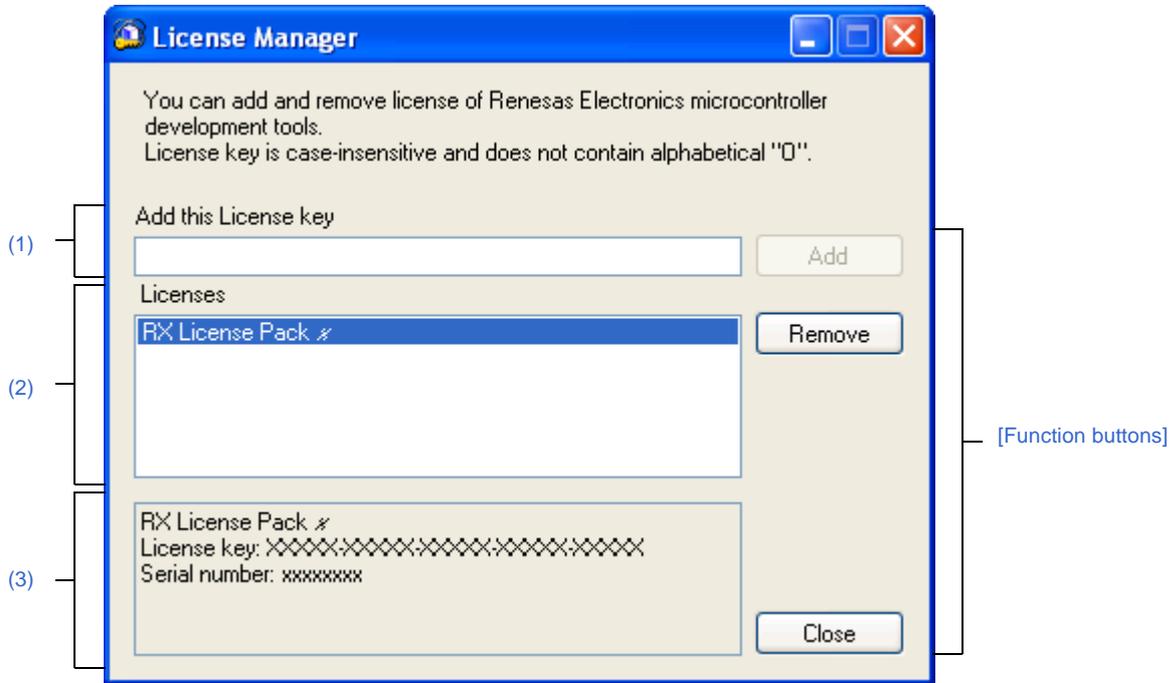


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[How to open]**

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

   **Remark**    In Windows 8 or Windows 8.1, double-click on [License Manager] on the start screen.

**[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**
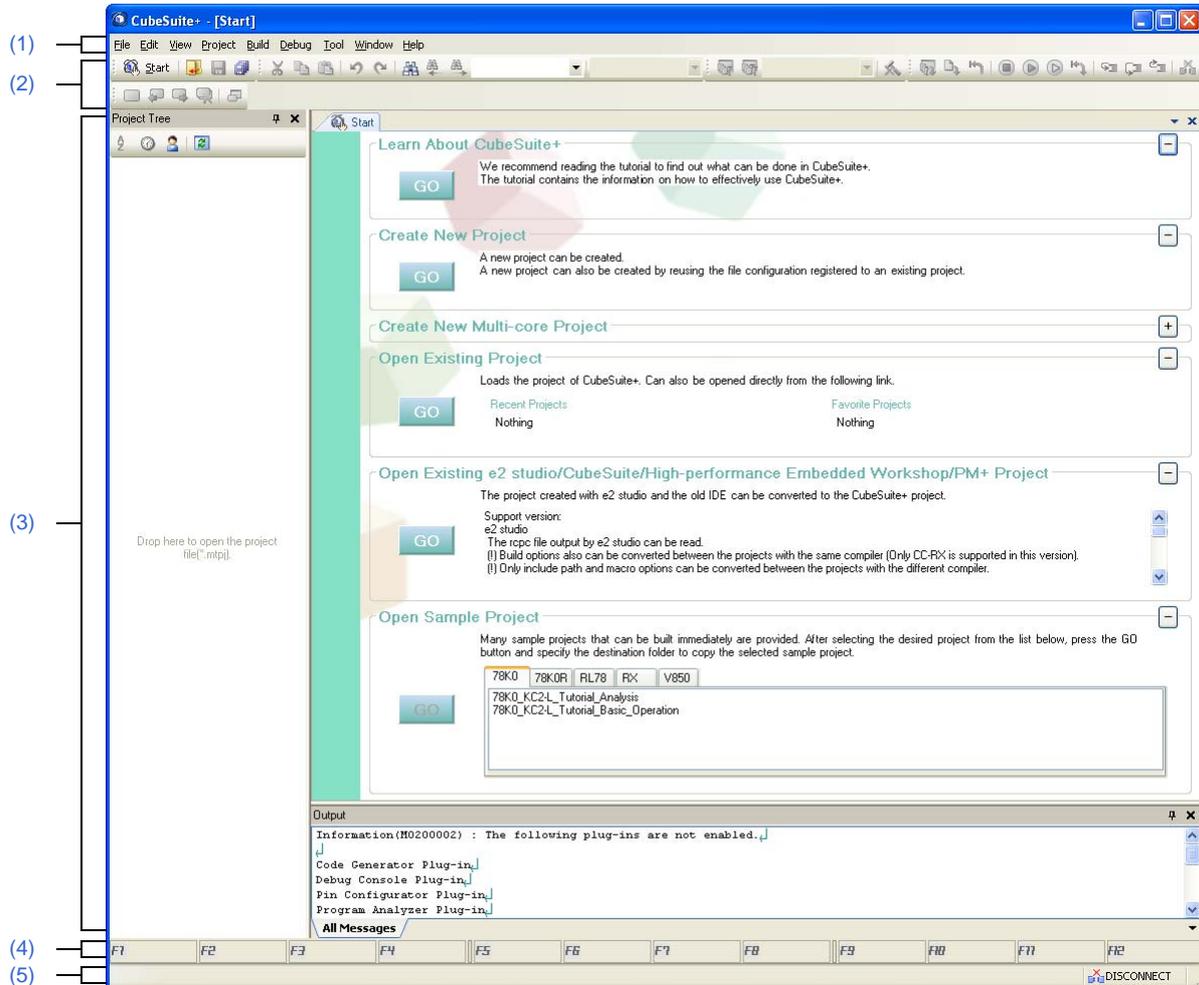


The following items are explained here.
- [How to open]
- [Description of each area]

**[How to open]**

- Select Windows [Start] >> [Programs] >> [Renesas Electronics CubeSuite+] >> [CubeSuite+].

    **Remark**    In Windows 8 or Windows 8.1, double-click on [CubeSuite+] on the start screen.

---

**[Description of each area]**

Remark    Parts of the menus and buttons are displayed only when the corresponding plug-ins are enabled.

**(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.<br>If the currently open project or its files have been modified, then the program will ask if you would like to save your changes. |
| | Create New File | Creates a new blank file ready for editing.<br>The default title for a newly created and unsaved Editor panel is "Source*X*"<br>(where *X* is the unique incremented number for creating the new source file). |
| Open... | | Opens the Open File dialog box for opening files and projects. |
| Open with Encoding... | | Opens the Open File dialog box, in which you can select a file to be opened with a specific encoding, and then the Encoding dialog box, in which you can select the encoding. |
| 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.<br>A file extension is assigned to the new file, and it is opened by the application. |
| Close Project | | Closes the currently open project.<br>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.<br>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 *Target* | | Saves the file that is currently in focus. |
| Save *Target*  As... | | Opens the Save As dialog box in order to save the contents of the file currently in focus with a different name. |
| *Target* 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 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. |
| Print Preview | Opens the Print Preview window to preview the source file before printing. |
| 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 | When the rapid start is enabled, this product enters the wait state. When the rapid start is disabled, exit from this product proceeds. 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 Find 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 Find and Replace dialog box and replace the specified string with another string. |
| Go To... | Opens the Go to Line dialog box if the Editor panel has the focus, and jumps to the specified line. Opens the Go to the Location dialog box if the Disassemble panel, SFR panel [R8C (Localised support)][78K0R][78K0], or IOR panel [RH850][RX][V850] has the focus, and jumps to the specified location. |

| Bookmark | | Displays a cascading menu for the bookmark. |
|---|---|---|
| | Toggle Bookmark | Insets/deletes the bookmark on the current line. |
| | Next Bookmark | Moves a caret to the position of the next bookmark. This menu is disabled in the following cases.  - No bookmark is registered.  - A bookmark is registered only in one line with a caret. |
| | Previous Bookmark | Moves a caret to the position of the previous bookmark. This menu is disabled in the following cases.  - No bookmark is registered.  - A bookmark is registered only in one line with a caret. |
| | Clear All Bookmarks | Clears all the registered bookmarks. This menu is disabled when no bookmark is registered. |
| | List Bookmarks... | Opens the Bookmarks dialog box for displaying the list of bookmarks. This menu is disabled when no bookmark is registered. |
| Outlining | | Displays a cascading menu for controlling expand and collapse states of source file outlining. |
| | Collapse to Definitions | Collapses all nodes that are marked as implementation blocks (e.g. function definitions). |
| | Toggle Outlining Expansion | Toggles the current state of the innermost outlining section in which the cursor lies when you are in a nested collapsed section. |
| | Toggle All Outlining | Toggles the collapsed state of all outlining nodes, setting them all to the same expanded or collapsed state. If there is a mixture of collapsed and expanded nodes, all nodes will be expanded. |
| | Stop Outlining | Stops code outlining and remove all outlining information from source files. |
| | Start Automatic Outlining | Starts automatic code outlining and automatically displayed in supported source files. |
| Advanced | | Displays a cascading menu for performing an advanced operation for the Editor panel. |

| Increase Line Indent | Increases the indentation of the current cursor line by one tab. |
|---|---|
| Decrease Line Indent | Decreases the indentation of the current cursor line by one tab. |
| Uncomment Lines | Removes the first set of line-comment delimiters from the start of the current cursor line, appropriate to the current language (e.g. C++). This operation will only be available when the language of the current source file has line-comment delimiters specified (e.g. C++). |
| Comment Lines | Places line-comment delimiters at the start of the current cursor line, appropriate to the current language (e.g. C++). This operation will only be available when the language of the current source file has line-comment delimiters specified (e.g. C++). |
| Convert Tabs to Spaces | Converts all tabs on the current cursor line into spaces. |
| Convert Spaces to Tabs | Converts each set of consecutive space characters on the current line to tab characters, but only for those sets of spaces that are at least equal to one tab size. |
| Tabify Selected Lines | Tabifies the current line, causing all spaces at the start of the line (prior to any text) to be converted to tabs where possible. |
| Untabify Selected Lines | Untabifies the current line, causing all tabs at the start of the line (prior to any text) to be converted to spaces. |
| Make Uppercase | Converts all letters within the selection to uppercase. |
| Make Lowercase | Converts all letters within the selection to lowercase. |
| Toggle Character Casing | Toggles the character cases (uppercase or lowercase) of all letters within the selection. |
| Capitalize | Capitalizes the first character of every word within the selection. |
| Delete Horizontal Whitespace | Deletes any excess white space either side of the cursor position, leaving only one whitespace character remaining. If there the cursor is within a word or not surrounded by whitespace, this operation will have no effect. |
| Trim Trailing Whitespace | Deletes any trailing whitespace that appears after the last non-whitespace character on the cursor line. |
| Delete Line | Completely delete the current cursor line. |
| Duplicate Line | Duplicates the cursor line, inserting a copy of the line immediately after the cursor line. |
| Delete Blank Lines | Deletes the line at the cursor if it is empty or contains only whitespace. |

**(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. |
| Error List | Shows the Error List 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 [R8C (Localised support)][78K0R][78K0]/IOR [RH850][RX][V850] | | Shows the SFR panel [R8C (Localised support)][78K0R][78K0]/IOR panel [RH850][RX][V850]. |
| CPU Register | | Shows the CPU Register panel. |
| Trace | | Shows the Trace panel.[IECUBE][IECUBE2][Full-spec emulator][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. |
| Back to Last Cursor Position | | Returns to the last location before moving to the defined location. |
| Forward to Next Cursor Position | | Goes to the move destination. |
| Python Console | | Shows the Python Console panel. |
| 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. |

**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. |

| Favorite Projects | | Shows the cascading menu to open and add to your favorite project. |
|---|---|---|
| | 1 *path* | Opens the project added by [1 Register to Favorite Project] in [Favorite Projects]. If no project is added, "1 Favorite Project" is shown. |
| | 2 *path* | Opens the project added by [2 Register to Favorite Project] in [Favorite Projects]. If no project is added, "2 Favorite Project" is shown. |
| | 3 *path* | Opens the project added by [3 Register to Favorite Project] in [Favorite Projects]. If no project is added, "3 Favorite Project" is shown. |
| | 4 *path* | Opens the project added by [4 Register to Favorite Project] in [Favorite Projects]. If no project is added, "4 Favorite Project" is shown. |
| | 1 Register to Favorite Project | The current project path is added to [1 *path*] in [Favorite Projects]. |
| | 2 Register to Favorite Project | The current project path is added to [2 *path*] in [Favorite Projects]. |
| | 3 Register to Favorite Project | The current project path is added to [3 *path*] in [Favorite Projects]. |
| | 4 Register to Favorite Project | The current project path is added to [4 *path*] in [Favorite 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 *Project* Active Project | | Sets the selected project or subproject to an active project. |
| Dependent Projects Settings... | | Opens the Dependent Projects Settings dialog box to set the dependent projects. Note that this menu is disabled when the build tool is in operation and any subproject does not exist. |
| 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 Development Tools as Package | | Saves a set of the project and this product 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 F USING AN EXTERNAL BUILD TOOL") are shown here.

See the "CubeSuite+ Integrated Development Environment User's Manual: Build" for the information about building for a project other than a debug-dedicated project.

| | |
|---|---|
| Build Project | Runs a build of the project (see "F.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 "F.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 "F.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 "F.5.4 Run a rapid build") between enabled (default) and disabled. |
| Build *active project* | Runs a build of the active project (see "F.5.1 Run a build"). If the active project is the main project, a build of its subproject is not run. When a dependent project is set for the active project, a build of the dependent project is also run. Note that this menu is disabled when the build tool is in operation. |
| Rebuild *active project* | Runs a rebuild of the active project (see "F.5.2 Run a rebuild"). If the active project is the main project, a rebuild of its subproject is not run. When a dependent project is set for the active project, a rebuild of the dependent project is also run. Note that this menu is disabled when the build tool is in operation. |
| Clean *active project* | Runs a clean of the active project (see "F.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 "F.5.5 Run a batch build"). |
| Build Option List | Lists the currently set build options in the Output panel. |

**(f) [Debug]**

See the "CubeSuite+ Integrated Development Environment User's Manual: Debug" for information about debugging.

**(g) [Tool]**

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

| | |
|---|---|
| Options... | Opens the Option dialog box. |
| Plug-in Setting... | Opens the Plug-in Manager dialog box. |

| Customize... | Opens the User Setting dialog box. |
| --- | --- |

**(h) [Window]**

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

| Split | Splits the active Editor panel horizontally. |
| --- | --- |
| | Only the active Editor panel can be splited. Other panels will not be splited. A panel can be splited a maximum of two times. |
| Remove Split | Removes the split view of the Editor panel. |
| Close All Panels | Closes all splited panels. |
| 1 *Tab name of panel* | Moves the focus to 1 *tab name of panel*. |
| 2 *Tab name of panel* | Moves the focus to 2 *tab name of panel*. |
| 3 *Tab name of panel* | Moves the focus to 3 *tab name of panel*. |
| 4 *Tab name of panel* | Moves the focus to 4 *tab name of panel*. |
| 5 *Tab name of panel* | Moves the focus to 5 *tab name of panel*. |
| 6 *Tab name of panel* | Moves the focus to 6 *tab name of panel*. |
| 7 *Tab name of panel* | Moves the focus to 7 *tab name of panel*. |
| 8 *Tab name of panel* | Moves the focus to 8 *tab name of panel*. |
| 9 *Tab name of panel* | Moves the focus to 9 *tab name of panel*. |
| Other Windows... | If 10 or more divide panels are open, open the Other Windows... dialog box. |

**(i) [Help]**

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

| Help | Opens the help for this product. |
| --- | --- |
| Open Help for *Target* | 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 tutorial for this product. |
| Browse Renesas Electronics Microcontrollers Web | Opens our microcontroller website in the web browser. |
| Detail Version Information... | Shows the Detail Version Information dialog box. |
| Check for Updates... | Opens the CubeSuite+ Update Manager window, and begin checking for updates. |
| About *Product name*... | Opens the Version Information dialog box. |

**(2) Toolbar**

The toolbar shows common command buttons.  See the "CubeSuite+ Integrated Development Environment User's Manual: Build", "CubeSuite+ Integrated Development Environment User's Manual: Debug", and "CubeSuite+ Integrated Development Environment User's Manual: Analysis" for buttons relating to building, debugging, and analysis.

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.

| | |
|---|---|
| Start | Opens the Start panel. |
| | Opens the Open File dialog box for opening files and projects. |
| | 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 Find 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. |
| ▼ | Pressing the [Enter] key leads to the execution of a quick search for the specified word.**Note** |
| 100% ▼ | Zooms in and out on the display of the panel that currently has focus.<br><br>The zoom percentage (25 to 300%) is specified by selecting it from the drop-down list or directly entering the value into the text box.<br><br>It is also possible to zoom in and out on the display by moving the mouse wheel forward or backward while holding down the [Ctrl] key. |

**Note** The find (including Quick Find) and replace conditions are individually set for the Output panel and Editor panel.
Accordingly, for example, when finding of a word with the Editor panel active proceeds after finding in the Output panel, if the find and replace conditions in use are different, the string found in the Output panel may not be found in the Editor panel.

**(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.<br>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.<br>Note that this button is disabled when the build tool is in operation. |
| DefaultBuild ▾ | Changes the current build modes for projects (main project and subprojects) simulta-neously.<br>All build modes that exist in the current project (main project and subprojects) are displayed in this drop-down list.<br>The current build modes of all projects are same, the build mode is selected by default. If they are not same, it will be blank.<br>The build mode that exists only in part of the project is shown with the mark "*". When the build mode with "*" mark is selected and it does not exist in the project, "DefaultBuild" will be duplicated and set with the selected build mode name.<br>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+ Integrated Development Environment User's Manual: Build"
- When the target project type is a debug-dedicated project: "F.5 Run a Build"

**(c) Bookmark toolbar**

Bookmark toolbar shows buttons operating bookmarks on the active Editor panel.

Note that this toolbar is disabled when the Editor panel does not have focus and the debug tool is connected (in mixed display mode).

| | |
|---|---|
| | Insets/deletes the bookmark on the current line. |
| | Moves a caret to the position of the previous bookmark.<br>This button is disabled in the following cases.<br> - No bookmark is registered.<br> - A bookmark is registered only in one line with a caret. |
| | Moves a caret to the position of the next bookmark.<br>This button is disabled in the following cases.<br> - No bookmark is registered.<br> - A bookmark is registered only in one line with a caret. |
| | Clears all the registered bookmarks.<br>This button is disabled when no bookmark is registered. |
| | Opens the Bookmarks dialog box for displaying the list of bookmarks. |

**(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+ Integrated Development Environment User's Manual: 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**
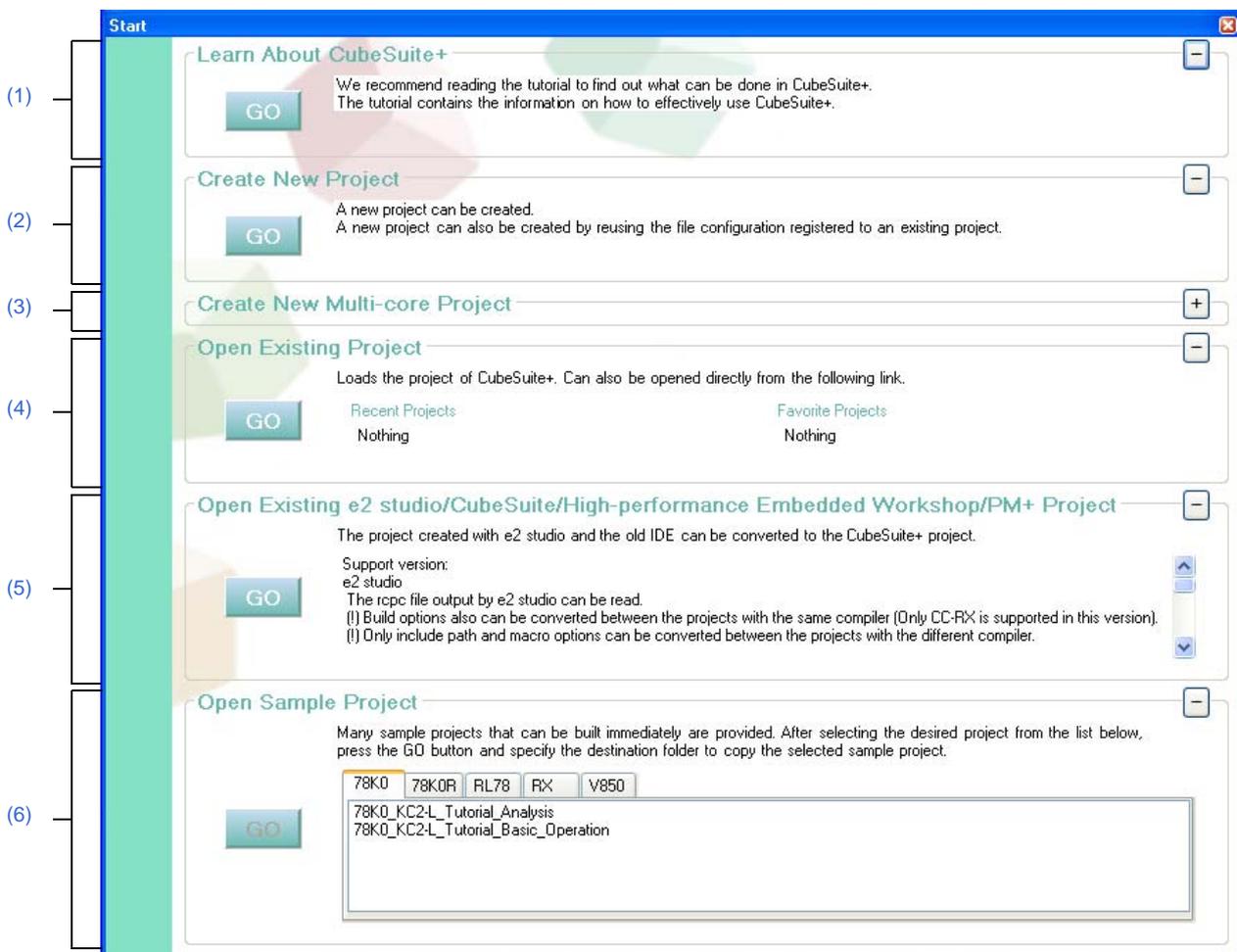


The following items are explained here.

- [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 GO button to show a tutorial aimed at improving your understanding of CubeSuite+.

---

**(2) [Create New Project] area**

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

**(3) [Create New Multi-core Project] area**

**(a) [Create New Multi-core Project]**
Click the  GO  button to open the Create Project dialog box, in order to create a new boot loader project and a new application project for multi-core.

**Remark**   This area is minimized by default; click  +  to resize the area.

**(4) [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  GO  button to open the Open Project dialog box, in order to open an existing CubeSuite+ project file (with ".mtpj" extension).

**(5) [Open Existing e2 studio/CubeSuite/High-performance Embedded Workshop/PM+ Project] area**

**(a) [Open Existing e2 studio/CubeSuite/High-performance Embedded Workshop/PM+ Project]**
Click the  GO  button to open the Open Project dialog box, in order to open an existing $e^2$ studio project file (with ".rcpc" extension)/CubeSuite project file (with ".cspj" extension)/HEW workspace file (with ".hws" extension)/HEW project file (with ".hwp" extension)/PM+ workspace file (with ".prw" extension)/PM+ project file (with ".prj" extension).

**(6) [Open Sample Project] area**

**(a) [Open Sample Project]**
Load a sample project file (with ".mtpj" extension) provided by CubeSuite+.
Select the [*Microcontroller]* tab, then select a sample project from the list.  Next, click the  GO  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.

**(7) 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 (Opened From Other Area than [Create New Multi-core Project] On Start Panel)**
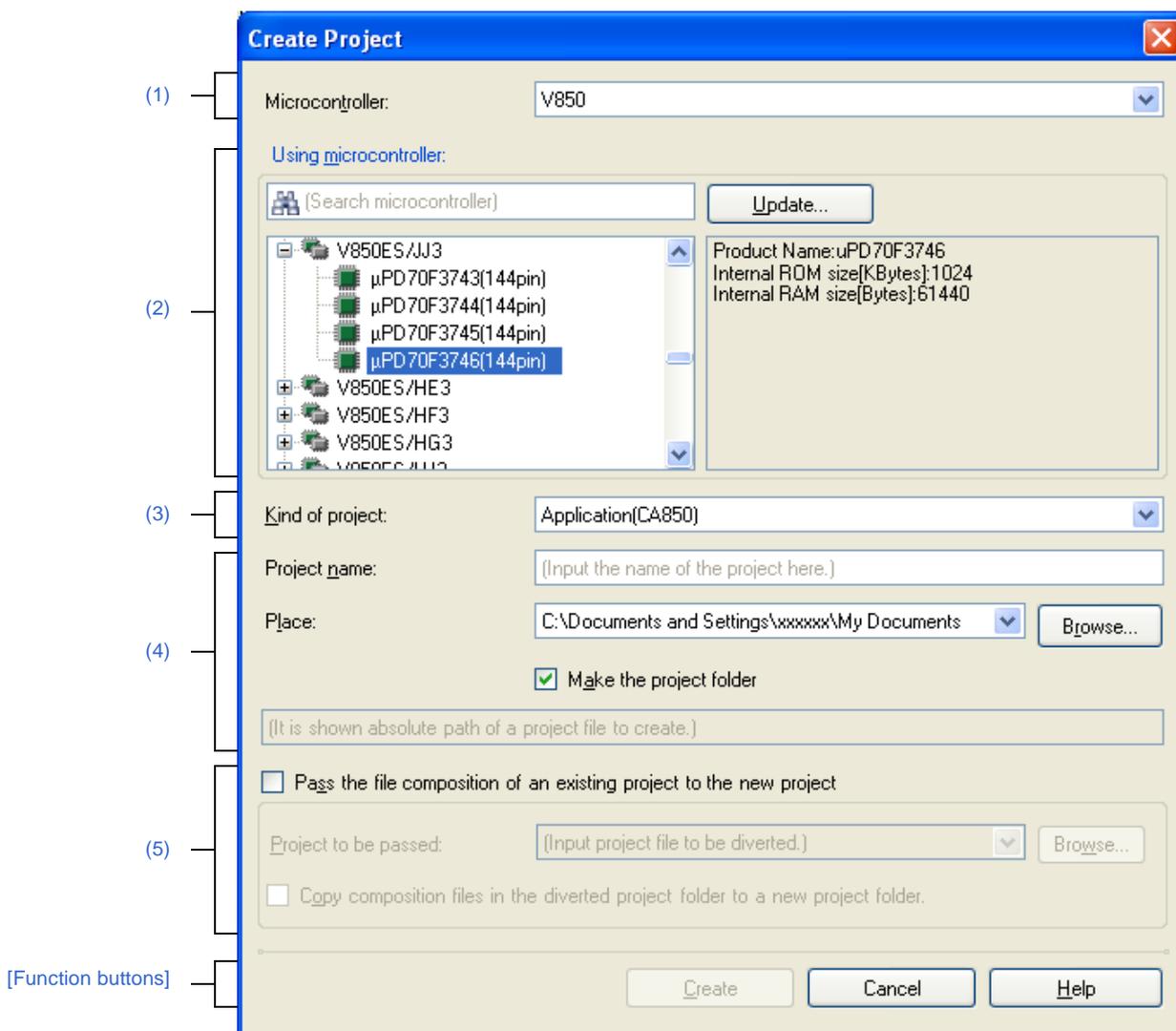


**Remark**      When a subproject is created, the title bar says "Create Subproject".

---

**Figure A-17.   Create Project Dialog Box (Opened From [Create New Multi-core Project] Area On Start Panel)**
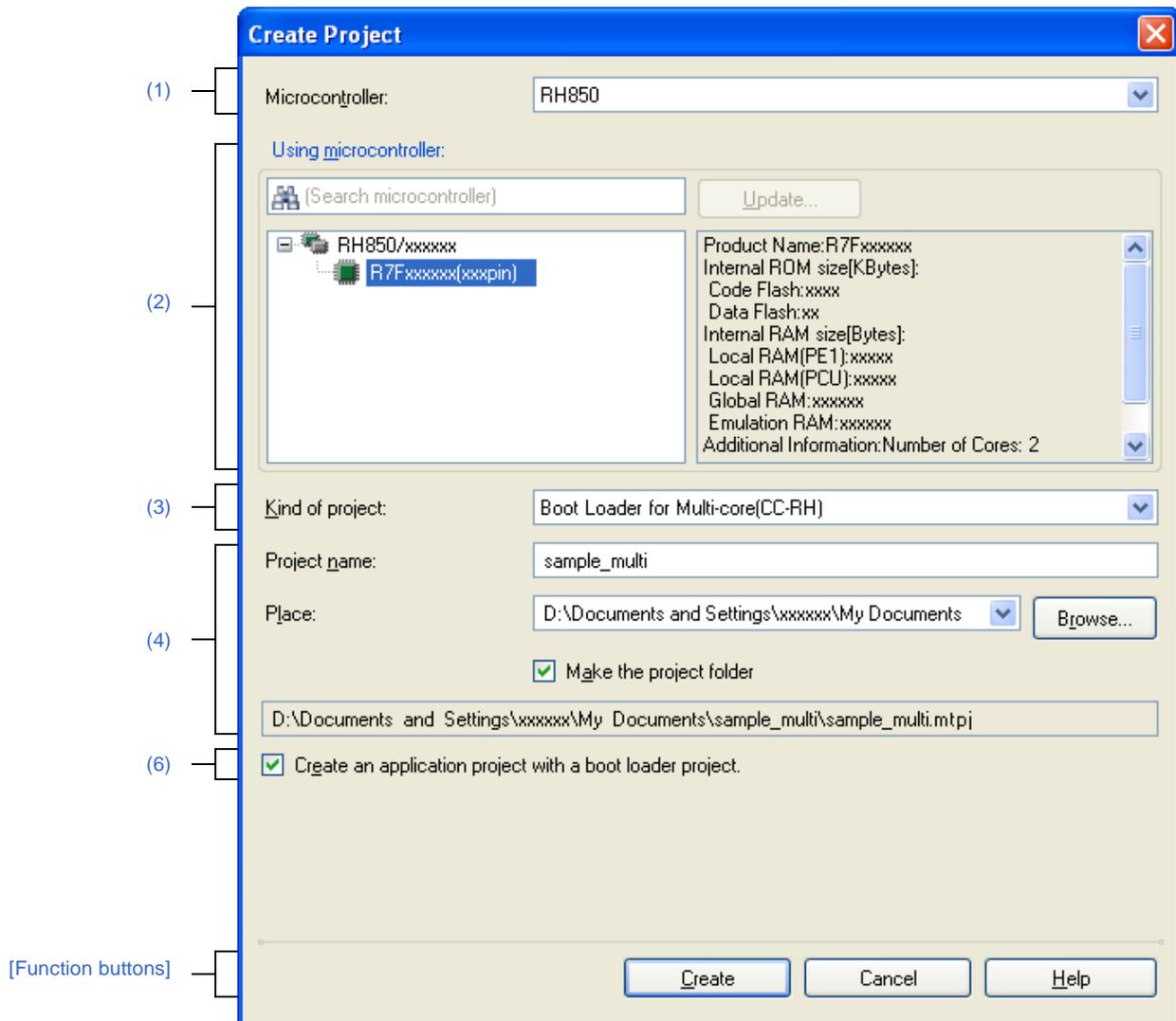


The following items are explained here.

- [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 or [Create New Multi-core 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.

- RH850
- RX
- V850
- R8C (Localised support)
- RL78
- 78K0R
- 78K0

**(2)  [Using microcontroller] area**

Select a microcontroller to use in the project.

**(a)  Microcontroller search box**

You can search a microcontroller selected in the list of microcontrollers simply.

**(b)  List of microcontrollers**

Select a microcontroller.

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

**Caution**    **If this dialog box is opened from the [Create New Multi-core Project] area on the Start panel, only the microcontrollers for RH850 multi-core are displayed.**

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

- Product Name
- On-chip ROM size [Kbytes] [RX]
- Internal ROM size [Kbytes] [RH850][V850][R8C (Localised support)][RL78][78K0R][78K0]
- On-chip RAM size [Bytes] [RX]
- Internal RAM size [Bytes] [RH850][V850][R8C (Localised support)][RL78][78K0R][78K0]
- Additional Information[Note]

**Note**   Other information appears only when present.

**(c)  Button**

| Update... | Displays the CubeSuite+ Update Manager window. |
|---|---|
| | When the microcontroller to be used does not appear in the list, you can search for the microcontroller via the net work. |

**Caution**    **This button is only enabled when this product is installed using the installer.  It is disabled when a packaged item is being used.**

**(3)  [Kind of project]**

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

You can select the item below.

**Caution      Only [Boot Loader for Multi-core(CC-RH)] can be selected when this dialog box is opened from the [Create New Multi-core Project] area on the Start panel.**

- Application(CC-RH/CC-RX/CA850/CX/NC30 (Localised support)/CA78K0R/CA78K0)
  Select this to generate the ROMization module file [CA850][CA78K0R], load module file, and hex file from C source files, by using the build tool provided by CubeSuite+.
  The generated file will be the debug target.

**Caution      This item is not displayed when microcontrollers for RH850 multi-core are selected in the [Using microcontroller] area.**

**Remarks 1.**    If the build tool is CC-RH, the following start-up source files are generated in the project folder. These files are also registered in the project tree.

| File Name | Description |
|-----------|-------------|
| cstart.asm | For defining the start-up routine from the occurence of a reset to a branch to the main function |
| iodefine.h | For defining I/O registers |
| main.c | For defining the empty main function |
| vecttbl.asm | For defining the interrupt vector table |

**2.**    If the build tool is CC-RX, the following start-up source files are generated in the project folder. The files marked with "OK" in the "Registration in Project Tree" column are also registered in the project tree.  When necessary, also register the files marked with "--" in the project tree.

| File Name | Description | Registration in Project Tree |
|-----------|-------------|------------------------------|
| *ProjectName*.c | For defining the main function | OK |
| dbsct.c | For setting up standard sections | OK |
| hwsetup.c | For initializing hardware | -- |
| intprg.c | For defining interrupt functions | OK |
| iodefine.h | For defining I/O registers | OK |
| lowlvl.src | For defining low-level I/O functions (source file for assembler) | -- |
| lowsrc.c | For defining low-level I/O functions | -- |
| lowsrc.h | Prototypes for low-level I/O functions | -- |
| resetprg.c | For defining initialization for C language | OK |
| sbrk.c | For defining the function for allocating the heap memory | OK |
| sbrk.h | For defining the heap size | OK |
| stacksct.h | For defining pragma for the stack | OK |

| File Name | Description | Registration in Project Tree |
|---|---|---|
| typedefine.h | For defining typedefine for the types used in sbrk.c, etc. | OK |
| vect.h | Prototypes for interrupt vector functions | OK |
| vecttbl.c | For defining the interrupt vector table | OK |

**3.**   If the build tool is NC30 (Localised support), the following start-up source files are generated in the project folder.

These files are also registered in the project tree.

| File Name | Description |
|---|---|
| device.c | For defining a constant or initialization of the standard input/output |
| firm.c | For defining a firmware area for OCD |
| fvector.c | For defining the fixed vector table |
| heap.c | For defining the static variable for the heap area |
| init.c | For defining initialization of individual standard inputs/outputs (stdin, stdout, etc.) |
| initsct.c | For defining initialization of each section |
| initsct.h | For defining a macro for initializing sections |
| intprg.c | For defining the address of the interrupt vector |
| resetprg.c | For defining initialization of the C language |
| resetprg.h | For defining a constant for resetprg.c, a macro, and include |
| sfr_r8xx.h | For defining SFR (for C language) |
| sfr_r8xx.inc | For defining SFR (for assembly language) |
| typedefine.h | For defining the scalar (integer) type |

- C++ Application(CC-RX)

Select this to generate the load module file and hex file from a C++ source file (only a file that has a main function) and C source files, by using build tool CC-RX provided by CubeSuite+.

The generated file will be the debug target.

**Remark**   The following start-up source files are generated in the project folder.

The files marked with "OK" in the "Registration in Project Tree" column are also registered in the project tree.  When necessary, also register the files marked with "--" in the project tree.

| File Name | Description | Registration in Project Tree |
|---|---|---|
| *ProjectName*.c | For defining the main function | OK |
| dbsct.c | For setting up standard sections | OK |
| hwsetup.c | For initializing hardware | -- |
| intprg.c | For defining interrupt functions | OK |
| iodefine.h | For defining I/O registers | OK |
| lowlvl.src | For defining low-level I/O functions (source file for assembler) | -- |

| File Name | Description | Registration in Project Tree |
|-----------|-------------|------------------------------|
| lowsrc.c | For defining low-level I/O functions | -- |
| lowsrc.h | Prototypes for low-level I/O functions | -- |
| resetprg.c | For defining initialization for C language | OK |
| sbrk.c | For defining the function for allocating the heap memory | OK |
| sbrk.h | For defining the heap size | OK |
| stacksct.h | For defining pragma for the stack | OK |
| typedefine.h | For defining typedef for the types used in sbrk.c, etc. | OK |
| vect.h | Prototypes for interrupt vector functions | OK |
| vecttbl.c | For defining the interrupt vector table | OK |

- Empty Application(CC-RH/CC-RX/NC30 (Localised support))
  Select this to generate the load module file, by using build tool CC-RH/CC-RX/NC30 (Localised support) pro-
  vided by CubeSuite+.
  The generated file will be the debug target.
  Sample startup programs are not generated when the project is created.

- Boot Loader for Multi-core(CC-RH)
  Select this to create a boot loader project for multi-core, by using build tool CC-RH provided by CubeSuite+.
  A boot loader project is used to set application projects configuring a project for multi-core.

**Caution**   **This item is displayed only when microcontrollers for RH850 multi-core are selected in the**
              **[Using microcontroller] area.**

**Remark**    The following start-up source files are generated in the project folder.
              These files are also registered in the project tree.

| File Name | Description |
|-----------|-------------|
| boot.asm | For defining the processing from the occurence of a reset to a branch to each application project |
| iodefine.h | For defining I/O registers |
| vecttbl.asm | For defining the interrupt vector table |

- Application for Multi-core(CC-RH)
  Select this to create an application project for multi-core, by using build tool CC-RH provided by CubeSuite+.

**Caution**   **This item is displayed only when microcontrollers for RH850 multi-core are selected in the**
              **[Using microcontroller] area.**

**Remark**    The following start-up source files are generated in the project folder.
              These files are also registered in the project tree.

| File Name | Description |
|-----------|-------------|
| cstartm.asm | For defining the start-up routine for each application |

| File Name | Description |
|---|---|
| iodefine.h | For defining I/O registers |
| main.c | For defining the empty main function |

- Library(CC-RH/CC-RX/CA850/CX/NC30 (Localised support)/CA78K0R/CA78K0)
  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 F   USING AN EXTERNAL BUILD TOOL" for details on how to create and use the debug-dedicated project.

**(4) Project Creation area**
Set the project to create.

**(a) [Project name]**
Directly enter the name of the project to create.
The entered project name is followed by the extension, ".mtpj" 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... | Opens the Browse For Folder dialog box. |
|---|---|
| | When a folder is selected, an absolute path of the folder is shown in [Place]. |

**(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.
> **2.**  When the input for [Project name] violates any restriction, the following messages are shown in the tooltip.

| Message | Description |
|---|---|
| The project name is invalid.  Enter characters that can be used as a file name. | The project file name uses characters that are not allowed. |
| A subproject with the same ID has already been registered. | A subproject with the same ID has already been registered. |
| The subproject name is invalid.  A subproject with the same name cannot be made in the location where the main project exists. | In the specified location there is already a main project with the same name (excluding extension). |

**3.** When the input for [Place] violates any restriction, the following messages are shown in the tooltip.

| 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 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 location where the project is to be created does not exist. | The folder creation was canceled or failed, so that a nonexistent folder has been specified. |
| The location for project creation is invalid. Enter characters that can be used as a path name. | The file name with the invalid path is designated.  The characters, \, /, :, *, ?, ", <, >, \|, cannot be used for the folder name. |

**4.** 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**
Set this area when reusing the file composition of the existing project (main or subproject) to create a new project.

**Caution    This area is not displayed when this dialog box is opened from the [Create New Multi-core Project] area on the Start panel.**

**(a) [Pass the file composition of an existing project to the new project]**
Select this check box when reusing the file composition of the existing project 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 e$^2$ studio, CubeSuite, HEW, or PM+ project file.**

**Remarks 1.** Up to 259 characters can be entered.

**2.** 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 divert project does not exist. | The path includes a nonexistent folder, or the specified project file cannot be found. |
| The divert project name is invalid. Enter characters that can be used as a file name. | The file name with the invalid path is designated.  The characters, \, /, :, *, ?, ", <, >, |, cannot be used for the folder name. |
| The project which is being used can't be diverted. | The currently open project (main project or subproject) has been specified as the source project. |
| The specified file is invalid. Enter a divert project file. | An existing file that is not a project file or subproject file has been specified. |

**3.** 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 ▶▶.

**(c) Button**

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

**(d) [Copy composition files in the diverted project folder to a new project folder.]**
Select this check box when copying composition files in the project folder of the project specified in [Project to be passed] to a new project folder.
The check box is unchecked by default.
This field is enabled only when the [Pass the file composition of an existing project to the new project] check box is checked.

**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).

**2.** 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.).

**(6) [Create an application project with a boot loader project]**
When an application project is created at the same time as a subproject for the boot loader project, select this check box.
The check box is selected by default.

**Caution   This area is not displayed when this dialog box is opened from the [Create New Multi-core Project] area on the Start panel.**

**[Function buttons]**

| Button | Function |
|---|---|
| Create | Creates a project according to the designated condition and closes the dialog box.<br><br>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].<br><br>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-18.   Source Convert Setting Dialog Box**
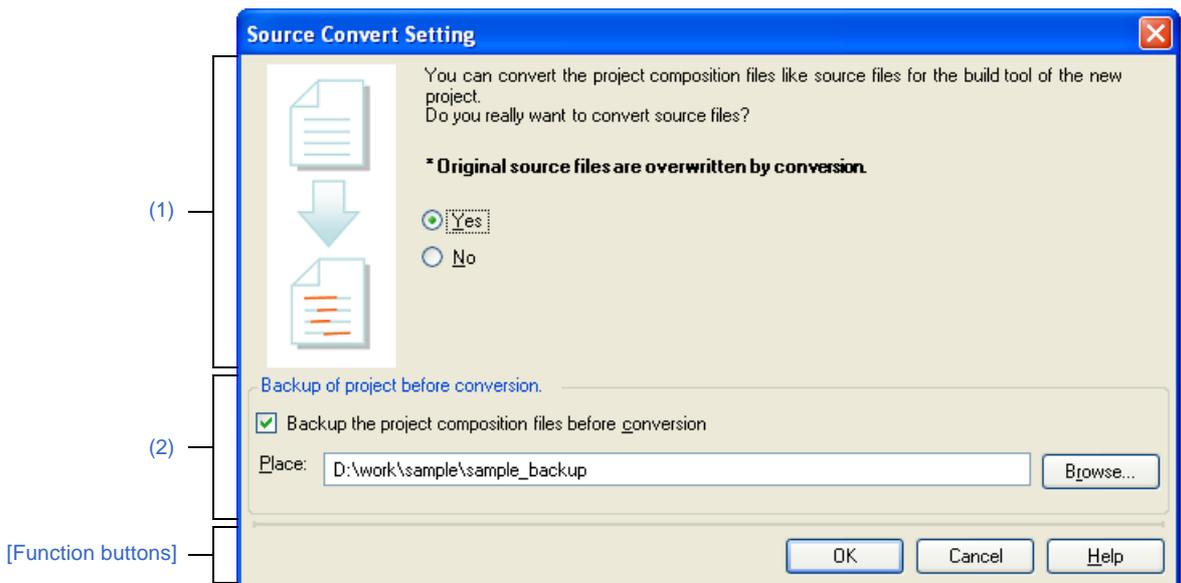


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[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.

**Caution**   **You cannot open this dialog box when the [Copy composition files in the diverted project folder to a new project folder] check box is checked from the Create Project dialog box.**

**[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 Convert Setting dialog box**

This dialog box includes the settings for conversion from e$^2$ studio and the old development environment (CubeSuite, HEW, PM+) project (or workspace) to a CubeSuite+ project.

**Remark** For details of the method of conversion of an old development environment project, see the following.
- "2.7.9 Convert an e$^2$ studio project into a CubeSuite+ project [RX]"
- "2.7.10 Convert a CubeSuite project into a CubeSuite+ project"
- "2.7.11 Convert a HEW project into a CubeSuite+ project"
- "2.7.12 Convert a PM+ project into a CubeSuite+ project"

**Figure A-19. Project Convert Setting Dialog Box**
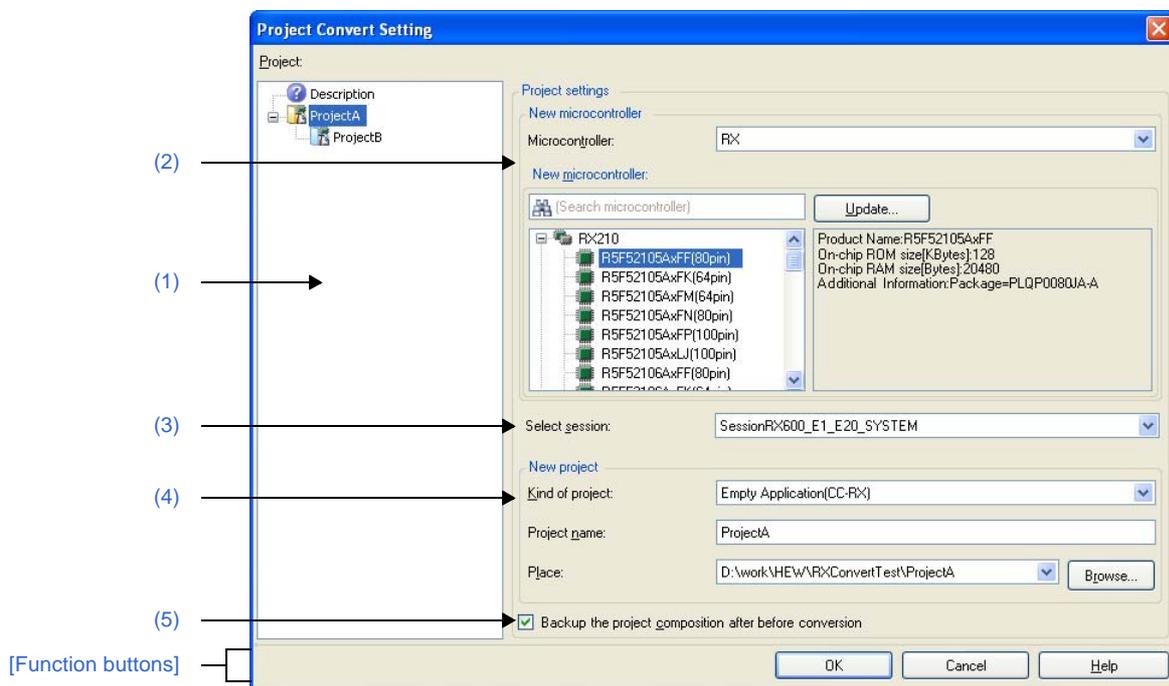


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[How to open]**

- From the [Project] menu, select [Open Project...] to open the Open Project dialog box.
  And then select an e$^2$ studio, CubeSuite, HEW, PM+ project (or workspace) and click the [Open] button.
- [Open Existing e2 studio/CubeSuite/High-performance Embedded Workshop/PM+ Project] area on the Start panel, click the [GO] button to open the Open Project dialog box.
  And then select an e$^2$ studio, CubeSuite, HEW, PM+ project (or workspace) and click the [Open] button.

---

**[Description of each area]**

**(1)  [Project]**

This area displays the conversion target project configuration tree.  Select the project for carrying out the conversion settings.

| | |
|---|---|
| ? | **Description node**<br>When you make the selection, the area on the right shows a summary of operations in this dialog box. |
| | **Main project node**<br>Indicates that this is the main project for the conversion target project.<br>When selected, the area on the right shows the main project setting items. |
| | **Subproject node**<br>Indicates that this is the subproject for the conversion target project.<br>When selected, the area on the right shows the subproject setting items. |

**Remarks 1.**   Holding down the [Ctrl] key and selecting a main project node and subproject node, allows you to make a multiple project setting.

**2.**   Selecting a main project node and then pressing the [*] key, allows you to expand the selected node.

Selecting a main project node and then pressing the [Shift] + [*] key, allows you to collapse the selected node.

**3.**   For the following conversions, subproject nodes do not appear.

-In an e$^2$ studio project

-In a CubeSuite project when there are no subprojects

-In a HEW workspace, when there is only one project

-In a HEW project

-In a PM+ project

**4.**   When the input for [Place] violates any restriction, the following messages are shown in the tooltip.

| Message | Description |
|---|---|
| Some project names are too long. Make it within 259 characters. | For project settings not selected in this area, the absolute path of the conversion target project file exceeds 259 characters. |
| The device is not specified in some projects.  Select the project in tree view, and please set the device. | The device list of the microcontroller selected for a project that is not selected in this area does not include the settings for that project.<br>This message is shown when CubeSuite+ does not support the device for the e$^2$ studio, CubeSuite, or PM+ project or when a device was not set after the HEW project was opened. |

**(2)  [New microcontroller] area**

Carry out the microcontroller settings for the conversion target project.

**Remark**   This area will be invalid when a PM+ workspace is converted and the main project node is selected.

**(a)  [Microcontroller]**

Select a microcontroller type to use in the new project.

You can select the item below.

- RH850
- RX
- V850
- R8C (Localised support)
- RL78
- 78K0R
- 78K0

**(b)  [New microcontroller] area**

Select a microcontroller to use in the new project.

**<1>  Microcontroller search box**

You can search a microcontroller selected in the list of microcontrollers simply.

**<2>  List of microcontrollers**

Select a microcontroller.

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
- On-chip ROM size [Kbytes] [RX]
- Internal ROM size [Kbytes] [RH850][V850][R8C (Localised support)][RL78][78K0R][78K0]
- On-chip RAM size [Bytes] [RX]
- Internal RAM size [Bytes] [RH850][V850][R8C (Localised support)][RL78][78K0R][78K0]
- Additional Information[Note]

**Note**   Other information appears only when present.

**<3>  Button**

| Update... | Displays the CubeSuite+ Update Manager window. |
|---|---|
| | When the microcontroller to be used does not appear in the list, you can search for the microcontroller via the net work. |

**Caution**   **This button is only enabled when this product is installed using the installer. It is disabled when a packaged item is being used.**

**Remark**   When the selection violates any restriction, the following messages are shown.

| Message | Description |
|---|---|
| Select the device. | Because no device is selected (for example, a device category is selected), it is not possible to determine the project type. |

**(3)　[Select session]**

Select the session used in the new project from the drop-down list.

Only the sessions in the HEW project that are linked to the target will appear in the drop-down list.

**Remark**　　This item is displayed only when the HEW project is converted and there are multiple sessions in it.

**(4)　[New project] area**

Set the project to create.

**(a)　[Kind of project]**

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

**Remark**　　This item will be invalid when a PM+ workspace is converted and the main project node is selected.

You can select the item below.

- Application(CA850/CX/CA78K0R/CA78K0)

Select this to generate the ROMization module file [CA850][CA78K0R], load module file, and hex file from C source files, by using the build tool provided by CubeSuite+.

The generated file will be the debug target.

- Empty Application(CC-RH/CC-RX/NC30 (Localised support))

Select this to generate the load module file, by using build tool provided by CubeSuite+.

The generated file will be the debug target.

Sample startup programs are not generated when the project is created.

- Library(CC-RH/CC-RX/CA850/CX/NC30 (Localised support)/CA78K0R/CA78K0)

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 F　USING AN EXTERNAL BUILD TOOL" for details on how to create and use the debug-dedicated project.

**(b)　[Project name]**

Directly enter the name of the project to create.

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

**Remarks 1.**　　When the subproject is created, the entered project name is followed by the extension, ".cssp" and forms the subproject name.

　　　　**2.**　　When there is an existing project file with the same name in the location for creating the project file, the file name has a number (from 1 to 99) appended thus: "_n_".

**(c)　[Place]**

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

By default, the old development environment project folder appears.

**(d)　Button**

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

**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.

**2.** When the input for [Project name] violates any restriction, the following messages are shown in the tooltip.

| Message | Description |
|---|---|
| The project name is invalid.  Enter characters that can be used as a file name. | The project file name uses characters that are not allowed. |
| The subproject name is invalid.  A subproject with the same name cannot be made in the location where the main project exists. | In the specified location there is already a main project with the same name (excluding extension). |
| Set the project name. | The project name is not specified. |

**3.** When the input for [Place] violates any restriction, the following messages are shown 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 location where the project is to be created does not exist. | The folder creation was canceled or failed, so that a nonexistent folder has been specified. |
| The location for project creation is invalid. Enter characters that can be used as a path name. | The file name with the invalid path is designated.  The characters, \, /, :, *, ?, ", <, >, \|, cannot be used for the folder name. |
| Some other projects are same setting. Change the project name or the folder. | Attempt to create a project file with the same absolute path as another project. |
| Set the place. | The creation location is not specified. |

**(5) [Backup the project composition after before conversion]**
Select this check box to pack up and save the project source files and complete project immediately after the conversion.

**Remark** This item will be invalid when a PM+ workspace is converted and the main project node is selected.

The destination folder for saving is on the same level as the conversion target project folder, named "*conversion-target-project-folder*_org".
If there is an existing folder file with the same name, the folder name has a number (from 1 to 99) appended thus: "_n_".

**[Function buttons]**

| Button | Function |
|---|---|
| OK | Closes this dialog box and converts 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-20.   Project Tree Panel**
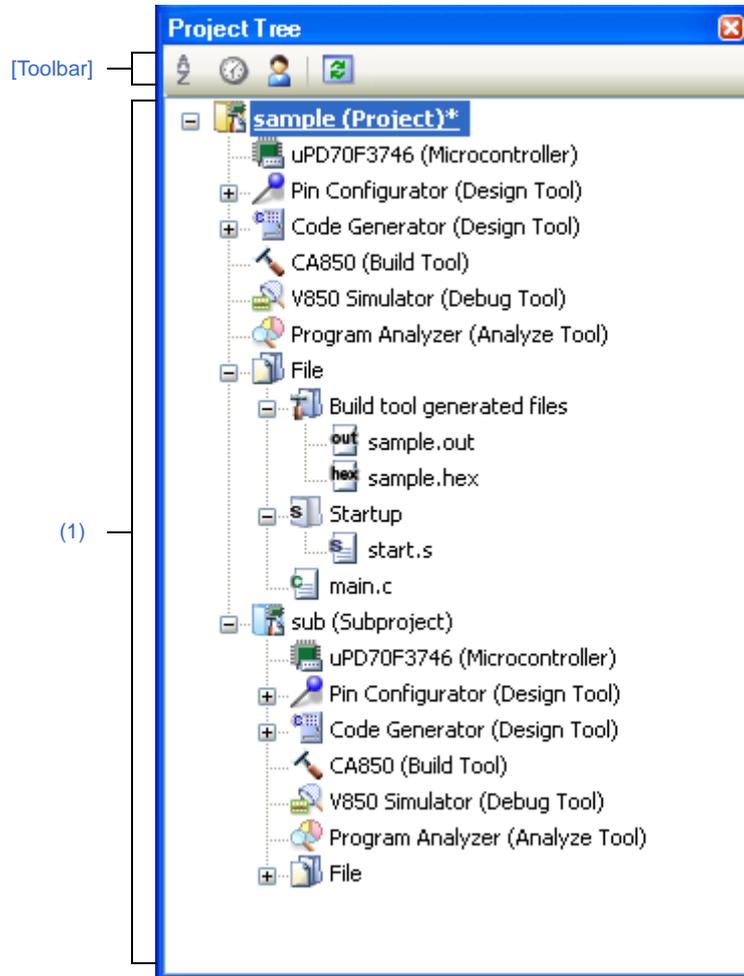
The following items are explained here.

- [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 |
|---|---|
| *Project name* (Project)<br>(Subsequently called "Project node") | The project's name. |
| *Microcontroller name* (Microcontroller)<br>(Hereafter referred to as "Microcontroller node") | The microcontroller used in the project. |
| Boot Loader (Configuration Tool for Multi-core) [RH850]<br>(Hereafter referred to as "Configuration Tool for Multi-core node") | The node for setting application projects which configure a project for multi-core.<br>Note that this node is shown when the project type is a boot loader project. |
| *Design Tool name* (Design Tool)<br>(Hereafter referred to as "Design Tool node") | The design tool (pin configurator, code generator, etc.) used.<br>Note that Code Generator (Design Tool) node is not shown when the project type is a debug-dedicated project. |
| *Build tool name* (Build Tool)<br>(Hereafter referred to as "Build Tool node") | The build tool (compiler, assembler, etc.) used.<br>When the project type is a debug-dedicated project, "None" is shown as *build tool name*. |
| *Debug tool name* (Debug Tool)<br>(Hereafter referred to as "Debug Tool node") | The debug tool (in-circuit emulator, simulator, etc.) used. |
| Program Analyzer (Analyze Tool)<br>(Hereafter referred to as "Analyze Tool node") | The analyze tool used.<br>Note that this node is not shown when the project type is a debug-dedicated project. |
| File<br>(Hereafter referred to as "File node") | Files registered to the project are displayed directly below the File node. |
| Download files<br>(Hereafter referred to as "Download files node") | This is a node for adding download files to the project.<br>Note that this node is shown only when the project type is a debug-dedicated project. |
| Build tool generated files<br>(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).  This node is always shown under the File node.<br>Note that this node is not shown when the project type is a debug-dedicated project. |
| Files for Multi-core [RH850]<br>(Hereafter referred to as "Files for multi-core node") | Files related to the project for multi-core are displayed directly below the Files for Multi-core node.<br>Note that this node is shown when an application project is related to the boot loader project.**Note** |
| Startup [V850][RL78][78K0R][78K0]<br>(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.<br>Note that this node is not shown when the project type is a debug-dedicated project. |

| Node | Explanation |
|---|---|
| *Category name*<br>(Hereafter referred to as "Category node") | These user-defined categories are used to classify files into modules.<br>Note that this node is always shown under the files node. |
| *Subproject name* (Subproject)<br>(Hereafter referred to as "Subproject node") | Subprojects added to the project. |

**Note**   The boot loader project and the application project are related in the Select Constituent Application Projects dialog box that is opened from the [Constituent application projects] property of the Configuration Tool for Multi-core node.

**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 F   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) Expand/collapse a node**

You can expand the selected node by the following procedure.
- Select the node and then press the [*] key.

You can collapse the selected node by the following procedure.
- Select the node and then press the [Shift] + [*] key.

**(g) Running the editor**

The file with the specific extension is opened in the Editor panel.  When an external text editor is set to be used in the Option dialog box, the file is opened with the external text 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.

**[Toolbar]**

| | |
|---|---|
| (icons) | Sorts files and Category nodes in Project tree area in order of their names.<br> : Ascending order<br> : Descending order<br> : Ascending order |
| (icons) | Sorts files in Project tree area in order of the time stamp.<br> : Descending order<br> : Ascending order<br> : Descending order |
| (icon) | Sorts files (other than the dependency files) and Category nodes in Project tree area in order of the user definition (default).<br>Display order is changed by dragging and dropping the file and Category node. |
| (icon) | Refreshes the state of highlighting for files, read-only files, non-existent files, and overlay icons. |

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

| | |
|---|---|
| Copy | Copies the selected file or Category node to the clip board. |
| | While editing the file name or the category name, the characters of the selection are copied to the clip board. |
| | Note that this menu is only enabled when the file (other than the dependency files) or the Category node is selected. |
| Paste | Inserts the contents of the clip board to the root of the node that is selected on the Project Tree. |
| | While editing the file name or the category name, insert the contents of the clip board. |
| | 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. |
| Rename | You can rename the selected project, subproject, file, and Category node. Press [Enter] key to confirm the rename. Press the [ESC] key to cancel. |
| | When the file is selected, the actual file name is also changed. |
| | When the selected file is added to other project, those file names are also changed. |
| | Note that this menu is only enabled when the project, subproject, file (other than the dependency files), and Category node is selected. Note that rename is disabled when the build tool is operating. |

**[[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 *Actively project* | Builds the active project. |
| | If the active project is the main project, its subproject is not built. |
| | When a dependent project is set for the active project, a build of the dependent project is also run. |
| | Note that this menu is disabled when the build tool is in operation. |
| Rebuild *Actively project* | Rebuilds the active project. |
| | If the active project is the main project, its subproject is not rebuilt. |
| | When a dependent project is set for the active project, a rebuild of the dependent project is also run. |
| | Note that this menu is disabled when the build tool is in operation. |
| Clean *Actively project* | Cleans the active project. |
| | If the active project is the main project, its subproject is not cleaned. |
| | Note that this menu is disabled when the build tool is in operation. |
| Open Folder with Explorer | Opens the folder that contains the project file of the selected project with Explorer. |
| Windows Explorer Menu | Displays the Windows Explorer menu corresponding to the project file for the selected project. |

| 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 *Selected project* Active Project | | Sets the selected project to an active project. |
| Save Project and Development Tools as Package | | Saves a set of the project and this product 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 *Actively project* | Builds the active project. When a dependent project is set for the active project, a build of the dependent project is also run. Note that this menu is disabled when the build tool is in operation. |
|---|---|
| Rebuild *Actively project* | Rebuilds the active project. When a dependent project is set for the active project, a rebuild of the dependent project is also run. Note that this menu is disabled when the build tool is in operation. |
| Clean *Actively project* | 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. |
| Windows Explorer Menu | Displays the Windows Explorer menu corresponding to the subproject file for the selected subproject. |

| 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.<br>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.<br>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.<br>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.<br>The default category name is "New category". The new category name can be changed to the same name as the existing Category node.<br>Note that this menu is disabled when the build tool is in operation. |
| Set *Selected subproject* Active Project | | Sets the selected subproject to an active project. |
| Remove from Project | | Removes the selected subproject from the project.<br>The subproject file itself is not deleted from the file system with this operation.<br>When the selected subproject is the active project, it cannot be removed from the project.<br>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**

| Change Microcontroller... | Opens the Change Microcontroller dialog box to change the microcontroller.<br>When multiple Microcontroller nodes for the projects using the same microcontroller are selected, the microcontroller settings can be changed at one time. |
|---|---|
| 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 F USING AN EXTERNAL BUILD TOOL") is selected are shown here. See the "CubeSuite+ Integrated Development Environment User's Manual: 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.<br>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 sub-project 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 sub-project 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. |
| Import Build Options... | Opens the Import Build Options dialog box to import the build options from the selected project file. |
| 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 F   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. |
| Open Folder with Explorer | This menu is always disabled. |
| Windows Explorer Menu | This menu is always disabled. |
| Remove from Project | This menu is always disabled. |
| Copy | This menu is always disabled. |
| Paste | Inserts the contents of the clipboard directly below this node. |
| | However, this menu is disabled when the contents of the clipboard exist in the same project. |
| Rename | This menu is always disabled. |
| Property | Displays this node's property on the Property panel. |

**(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.<br>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.<br>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.<br>Up to 200 characters can be specified.<br>The default category name is "New category".  You can also add a Category node with the same name as the existing Category node.<br>This menu is disabled when the build tool is in operation and when categories are nested 20 levels. |
| Open Folder with Explorer | | This menu is always disabled. |
| Windows Explorer Menu | | This menu is always disabled. |
| Remove from Project | | This menu is always disabled. |
| Copy | | This menu is always disabled. |
| Paste | | Inserts the contents of the clipboard directly below this node.<br>However, this menu is disabled when the contents of the clipboard exist in the same project. |
| Rename | | This menu is always disabled. |
| Property | | Displays this node's property on the Property panel. |

**(7)  When a Python console script file is selected**

| Execute in Python Console | Opens the Python Console panel to execute the selected Python script file. |
|---|---|
| Open | Opens the selected file with the application corresponds to the file extension (see "(g) Running the editor"). |
| Open with Internal Editor... | Opens the selected file with the Editor panel. |
| Open with Selected Application... | Opens the Open with Program dialog box to open the selected file with the designated application. |
| Open Folder with Explorer | Opens the folder that contains the selected file with Explorer. |
| Windows Explorer Menu | Displays the Windows Explorer menu corresponding to the selected file. |
| 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.<br>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.<br>Up to 200 characters can be specified.<br>The default category name is "New category".  You can also add a Category node with the same name as the existing Category node.<br>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.<br>The file itself is not deleted from the file system.<br>Note that this menu is disabled when the build tool is in operation. |
| Copy | Copies the selected file to the clipboard.<br>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.<br>The actual file is also renamed.<br>When the selected file is added to other project, that file name is also changed. |
| Change Extension... | Opens a message dialog box to confirm whether to change the file extension.<br>Clicking on the [Yes] button in the dialog box will open the Character String Input dialog box, in which the extension of the selected file can be changed.<br>When multiple files are selected, they are changed at one time.<br>Note that this menu item is disabled if the name of the selected file cannot be changed or the file cannot be removed from the project, or the build tool is in operation. |
| Property | Displays the selected file's property on the Property panel. |

**(8)  When a file (excluding a Python cosole script file) is selected**

Only menu items that are displayed when a file for a debug-dedicated project (see "APPENDIX F   USING AN EXTERNAL BUILD TOOL") is selected are shown here.  See the "CubeSuite+ Integrated Development Environment User's Manual: 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 "(g) Running the editor"). |
| Open with Internal Editor... | Opens the selected file with the Editor panel. |
| Open with Selected Application... | Opens the Open with Program dialog box to open the selected file with the designated application. |
| Open Folder with Explorer | Opens the folder that contains the selected file with Explorer. |
| Windows Explorer Menu | Displays the Windows Explorer menu corresponding to the selected file. |
| 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.<br>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.<br>Up to 200 characters can be specified.<br>The default category name is "New category".  You can also add a Category node with the same name as the existing Category node.<br>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.<br>The file itself is not deleted from the file system.<br>Note that this menu is disabled when the build tool is in operation. |
| Copy | Copies the selected file to the clipboard.<br>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.<br>The actual file is also renamed.<br>When the selected file is added to other project, that file name is also changed. |
| Change Extension... | Opens a message dialog box to confirm whether to change the file extension.<br>Clicking on the [Yes] button in the dialog box will open the Character String Input dialog box, in which the extension of the selected file can be changed.<br>When multiple files are selected, they are changed at one time.<br>Note that this menu item is disabled if the name of the selected file cannot be changed or the file cannot be removed from the project, or the build tool is in operation. |
| Property | Displays the selected file's property on the Property panel. |

**(9)  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.<br>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.<br>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.<br>Up to 200 characters can be specified.<br>The default category name is "New category".  You can also add a Category node with the same name as the existing Category node.<br>This menu is disabled when the build tool is in operation and when categories are nested 20 levels. |

| | |
|---|---|
| Open Folder with Explorer | Opens a shortcut to the folder that has been set for the selected category in the Explorer.<br>This menu item will be invalid if a shortcut to the folder has not been made. |
| Windows Explorer Menu | Shows the Windows Explorer menu for a shortcut to the folder that has been set for the selected category.<br>This menu item will be invalid if a shortcut to the folder has not been made. |
| Remove from Project | Removes the selected Category node from the project.<br>Note that this menu is disabled when the build tool is in operation. |
| Copy | Copies the selected Category node to the clipboard.<br>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.<br>However, this menu is disabled when the contents of the clipboard exist in the same project.<br>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-21.   Property Panel**
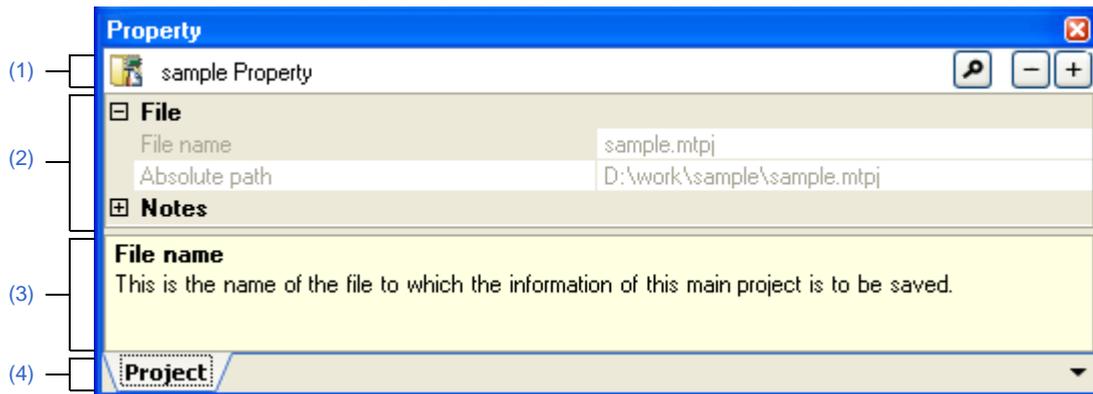


  The following items are explained here.
  - [How to open]
  - [Description of each area]
  - [Dialog boxes opened from the Property panel]
  - [[Edit] menu (Property panel-dedicated items)]
  - [[Help] menu (Property panel-dedicated items)]
  - [Context menu]

**[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, 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, or Category node is selected in Project Tree panel  while
             the Property panel is open, the detailed information of the selected node is displayed.

**[Description of each area]**

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

**(a)  Name for the selected node area**

In this area, the name of the selected node in Project Tree panel is displayed.

This area is left blank when multiple nodes are selected.

**(b)  button group**

| | |
|---|---|
| 🔓 🔒 | Locks properties to prevent the property value from being modified by wrong operation of a mouse or a keyboard.<br><br>🔓 : locks properties.<br>🔒 : unlocks properties.<br><br>This button is only on the following tabs.<br>- Tabs displayed when the Build Tool node is selected<br>- Tabs displayed when the C source file, C++ source file [CC-RX], assembly source file [CC-RH], assembler source file [CC-RX][CA850][CX][NC30 (Localised support)][CA78K0R][CA78K0], object file [CC-RH], object module file [CC-RX][CA850][CX][NC30 (Localised support)][CA78K0R][CA78K0], link directive file [CA850][CX][CA78K0R][CA78K0], symbol information file [CX], and library file is selected (except for [File Information] tab) |
| 🔍 | Opens the Find and Replace dialog box with the [Quick Find] tab target.<br>Property names, property values, and active-help character strings can be found. |
| − | Collapses the entire currently displayed detailed information display/change area. |
| + | Expands the entire currently displayed detailed information display/change area. |

**(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 categoraized 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, 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 HEX 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)  When the following nodes are selected in the Project Tree panel**

| | |
|---|---|
| Project node | - [Project] tab |

| Subproject node | - [Subproject] tab |
|---|---|
| Microcontroller node | - [Microcontroller Information] tab |
| Configuration Tool for Multi-core node | - [Boot Loader] tab [RH850] |
| Design Tool node | - [Pin Configurator Information] tab<br> - [Top View Setting] tab<br> - [Generation] tab<br> - [Macro Setting] tab<br>See the "CubeSuite+ Integrated Development Environment User's Manual: Design" for details about these tabs. |
| Build Tool node<br>(Except for a debug-dedicated project<sup>Note 1</sup>) | - [Common Options] tab<br> - [Compile Options] tab<br> - [Assemble Options] tab<br> - [Link Options] tab<br> - [ROMization Process Options] tab [CA850][CA78K0R]<br> - [ROMize Options] tab [CX]<br> - [Hex Output Options] tab [CC-RH][CC-RX][CX]<br> - [Hex Convert Options] tab [CA850]<br> - [Object Convert Options] tab [CA78K0R][CA78K0]<br> - [Librarian Options] tab [CC-RX]<br> - [Library Genetate Options] tab [CC-RX][NC30 (Localised support)]<br> - [Archive Options] tab [CA850]<br> - [Create Library Options] tab [CC-RH][CX][CA78K0R][CA78K0]<br> - [Section File Generate Options] tab [CA850]<br> - [Dump Options] tab [CA850]<br> - [Cross Reference Options] tab [CA850]<br> - [Memory Layout Visualization Options] tab [CA850]<br> - [Variables/Functions Relocation Options] tab [CA78K0R]<br> - [Variables Relocation Options] tab [CA78K0]<br> - [Memory Bank Relocation Options] tab [CA78K0]<br>See the "CubeSuite+ Integrated Development Environment User's Manual: Build" for details about these tab. |
| Build Tool node<br>(For a Debug-dedicated project<sup>Note 1</sup>) | - [Build Options] tab |
| Debug Tool node | - [Connect Settings] tab<br> - [Debug Tool Settings] tab<br> - [Flash Self Emulation Settings] tab [IECUBE]<br> - [DataFlash Emulation Settings] tab [IECUBE[RL78]] [IECUBE[78K0R]] [IECUBE[V850]]<br> - [Download File Settings] tab<br> - [Flash Options Settings] tab [V850E2]<br> - [Hook Transaction Settings] tab<br>See the "CubeSuite+ Integrated Development Environment User's Manual: Debug" for details about these tabs. |
| Analyze Tool node | - [Settings] tab<br>See the "CubeSuite+ Integrated Development Environment User's Manual: Analysis" for details about this tab. |

| | |
|---|---|
| File | - [Build Settings] tab (for C source file, C++ source file [CC-RX], assembly source file [CC-RH], assembler source file [CC-RX][CA850][CX][NC30 (Localised support)][CA78K0R][CA78K0], object file [CC-RH], object module file [CC-RX][CA850][CX][NC30 (Localised support)][CA78K0R][CA78K0], link directive file [CA850][CX][CA78K0R][CA78K0], symbol information file [CX], and library file) <br><br> - [Individual Compile Options] tab (for C source file) [CC-RH][CA850][CX][NC30 (Localised support)][CA78K0R][CA78K0] <br><br> - [Individual Compile Options(C)] tab (for C source file) [CC-RX] <br><br> - [Individual Compile Options(C++)] tab (for C++ source file) [CC-RX] <br><br> - [Individual Assemble Options] tab (for assembler source[Note 2]) <br><br> - [File Information] tab <br><br> See the "CubeSuite+ Integrated Development Environment User's Manual: Build" for details about the tabs other than the [File Information] tab. |
| Category node | - [Category Information] tab |

**Notes 1.**    For details on the debug-dedicated project, see "APPENDIX F   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) When the [Generate Code] button in the Code Generator panel is clicked**

---

 - [Macro Setting] tab

See the "CubeSuite+ Integrated Development Environment User's Manual: Design" for details about this tab.

---

**(c) When a file is selected on the Code Generator Preview panel**

---

 - [File Setting] tab

See the "CubeSuite+ Integrated Development Environment User's Manual: 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 "Character String Input dialog box", "CubeSuite+ Integrated Development Environment User's Manual: Build",

    or "CubeSuite+ Integrated Development Environment User's Manual: Debug" for details.

- Text Edit dialog box

    See "Text Edit dialog box", "CubeSuite+ Integrated Development Environment User's Manual: Build", or

    "CubeSuite+ Integrated Development Environment User's Manual: Debug" for details.

- Path Edit dialog box

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

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

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

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

| | |
|---|---|
| Open Help for Property Panel | Displays the help of this panel or the currently displayed tab. |

**[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.<br><br>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.<br><br>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-22.   Property Panel : [Project] Tab**
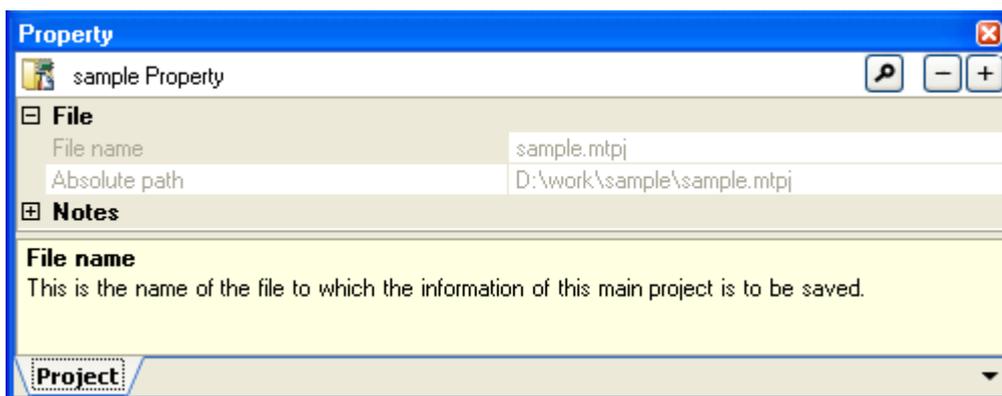


**[Description of each category]**

**(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 | *Name of the main project file* |
| | How to change | Changes not allowed |
| Absolute path | Displays the absolute path of the main project file to save the information on them. | |
| | Default | *Absolute name of main projects* |
| | 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.<br>Memo is added one item in one line.<br>The added memos are displayed as subproperty. | |
|---|---|---|
| | Default | Memo [*Number of items*] |
| | How to change | Edit by the Text Edit dialog box which appears when clicking the [...] button.<br>For the subproperty, you can use a text box directly enter the text. |
| | Restriction | Up to 256 characters<br>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-23.   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 | *Name of the subproject file* |
| | 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 | *Relative path from the main project* |
| | How to change | Changes not allowed |
| Absolute path | Displays the absolute path of the subproject file to save the information on them. | |
| | Default | *Absolute name of subprojects.* |
| | 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 [*Number of items*] |
| | 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)   [IOR Display Type] [RH850]
(4)   [Notes]

**Figure A-24.   Property Panel : [Microcontroller Information] Tab**
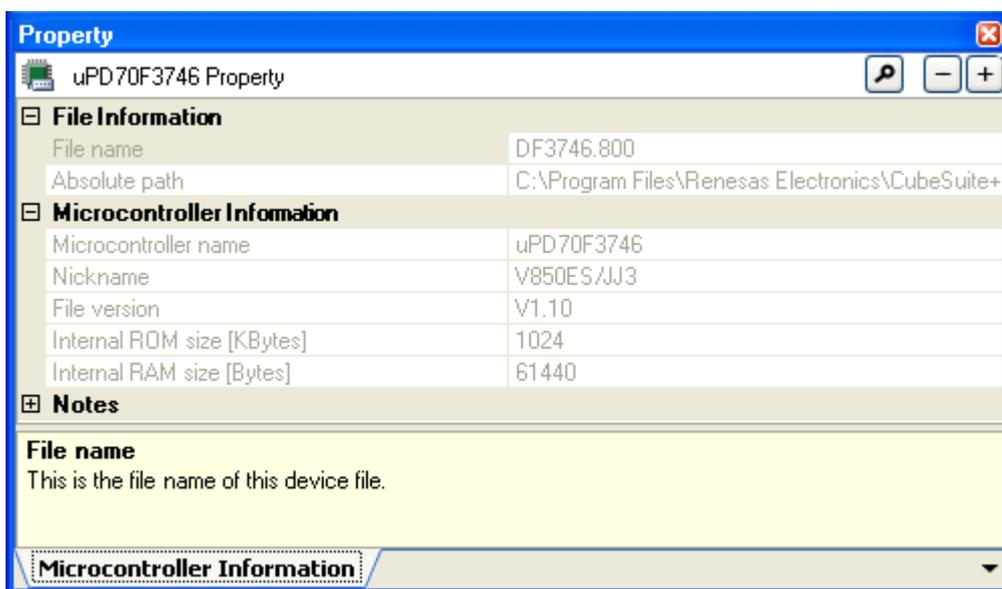


**[Description of each category]**

**(1)  [File Information]**

The detailed information on the file is shown.

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

**(3)  [IOR Display Type] [RH850]**

The detailed information on the IOR display type is shown.

| IOR display type | Displays the IOR display type of the project. | |
|---|---|---|
| | Default | *IOR display type of the project* |
| | How to change | Changes not allowed |

**(4)  [Notes]**

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

| Memo | Add memos to the microcontroller information.<br>The memo is added one item in one line.<br>The added memos are displayed as subproperty. | |
|---|---|---|
| | Default | Memo [*Number of items*] |
| | How to change | Edit by the Text Edit dialog box which appears when clicking the [...] button.<br>For the subproperty, you can use a text box directly enter the text. |
| | Restriction | Up to 256 characters<br>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 F   USING AN EXTERNAL BUILD TOOL") categorized by the following and the configuration can be changed.

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

**Figure A-25.   Property Panel: [Build Options] Tab**
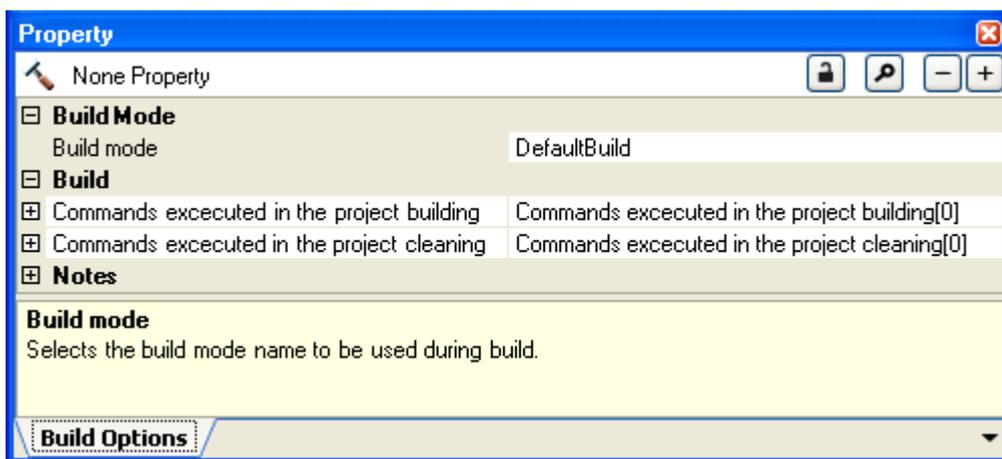


**[Description of each category]**

**(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. |
| | | *Build mode that is added to the project (other than DefaultBuild)* | 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 "F.5.1   Run a build") of the debug-dedicated project.<br><br>The following placeholders are supported.<br><br>%ActiveProjectDir%: Replaces with the absolute path of the active project folder.<br><br>%ActiveProjectName%: Replaces with the active project name.<br><br>%BuildModeName%: Replaces with the build mode name.<br><br>%MainProjectDir%: Replaces with the absolute path of the main project folder.<br><br>%MainProjectName%: Replaces with the main project name.<br><br>%MicomToolPath%: Replaces with the absolute path of the install folder of this product.<br><br>%ProjectDir%: Replaces with the absolute path of the project folder.<br><br>%ProjectName%: Replaces with the project name.<br><br>%TempDir%: Replaces with the absolute path of the temporary folder.<br><br>%WinDir%: Replaces with the absolute path of the Windows system folder.<br><br>When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed when running a build.<br><br>The placeholders can be described in the script. | | |
| | Default | Commands executed in the project building[0] | |
| | How to change | Edit by the Text Edit dialog box which appears when clicking the [...] button.<br>For the subproperty, you can enter directly in the text box. | |
| | Restriction | Up to 1023 characters<br>Up to 64 commands can be specified. | |
| Commands executed in the project cleaning | Specify the commands to be executed when running a clean (see "F.5.3   Run a clean") of the debug-dedicated project.<br><br>The following placeholders are supported.<br><br>%ActiveProjectDir%: Replaces with the absolute path of the active project folder.<br><br>%ActiveProjectName%: Replaces with the active project name.<br><br>%BuildModeName%: Replaces with the build mode name.<br><br>%MainProjectDir%: Replaces with the absolute path of the main project folder.<br><br>%MainProjectName%: Replaces with the main project name.<br><br>%MicomToolPath%: Replaces with the absolute path of the install folder of this product.<br><br>%ProjectDir%: Replaces with the absolute path of the project folder.<br><br>%ProjectName%: Replaces with the project name.<br><br>%TempDir%: Replaces with the absolute path of the temporary folder.<br><br>%WinDir%: Replaces with the absolute path of the Windows system folder.<br><br>When "#!python" is described in the first line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed when running a build.<br><br>The placeholders can be described in the script. | | |
| | Default | Commands executed in the project cleaning[0] | |
| | How to change | Edit by the Text Edit dialog box which appears when clicking the [...] button.<br>For the subproperty, you can enter directly in the text box. | |
| | Restriction | Up to 1023 characters<br>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[*number-of-items*] |
| | How to change | Edit by the Text Edit dialog box which appears when clicking the [...] button. |
| | | For the subproperty, you can enter directly in the text box. |
| | Restriction | Up to 256 characters |
| | | Up to 256 items can be specified. |

**[File Information] tab**

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

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

**Figure A-26.   Property Panel: [File Information] Tab**
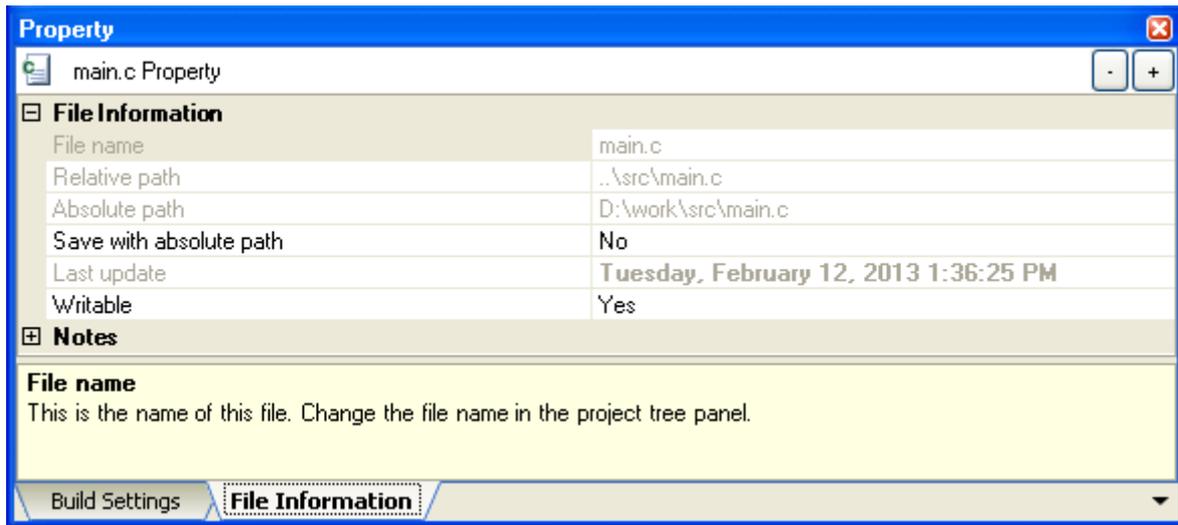


**[Description of each category]**

**(1)   [File Information]**

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

| File name | The file name is displayed. | |
|---|---|---|
| | Change the file name on the project tree. | |
| | Default | *file name* |
| | How to change | Changes not allowed |
| Relative path | The relative path from the project folder is displayed. | |
| | Default | *The relative path from the project folder* |
| | How to change | Changes not allowed |
| Absolute path | The absolute path of the file is displayed. | |
| | Default | *The absolute path of the file* |
| | How to change | Changes not allowed |

| Save with absolute path | Select whether to save the file location with the absolute path. This property is not displayed if the Property panel being displayed while a dependency file is selected on the project tree. | | |
|---|---|---|---|
| | Default | No | |
| | How to change | Select from the drop-down list. | |
| | Restriction | Yes | Saves the file location with the absolute path. |
| | | No | Saves the file location with the relative path. |
| Last update | The time and date on which this file was changed last is displayed. | | |
| | Default | *File updated time and date* | |
| | How to change | Changes not allowed | |
| Writable | Select whether to enable writing to the file. This property is not displayed if the Property panel being displayed while a dependency file is selected on the project tree. | | |
| | Default | Yes (when the file is write enabled) No (when the file is not write enabled) | |
| | How to change | Select from the drop-down list. | |
| | Restriction | Yes | Enables the file to write. |
| | | No | Does not enable the file to write. |

**(2) [Notes]**

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

This category is not displayed if the Property panel being displayed while a dependency file is selected on the project tree.

| Memo | Add memos to the file. Add one item in one line. The specified memo is displayed as the subproperty. | |
|---|---|---|
| | Default | Memo[*number of items*] |
| | How to change | Edit by the Text Edit dialog box which appears when clicking the [...] button. For the subproperty, you can enter directly in the text box. |
| | Restriction | Up to 256 characters Up to 256 items can be specified. |

**[Category Information] tab**

This tab shows the detailed information on the category node (the category that the user added), Files node, and Build tool generated files node categorized by the following and the configuration can be changed.

(1)   [Category Information]
(2)   [Notes]

**Figure A-27.   Property Panel: [Category Information] Tab**
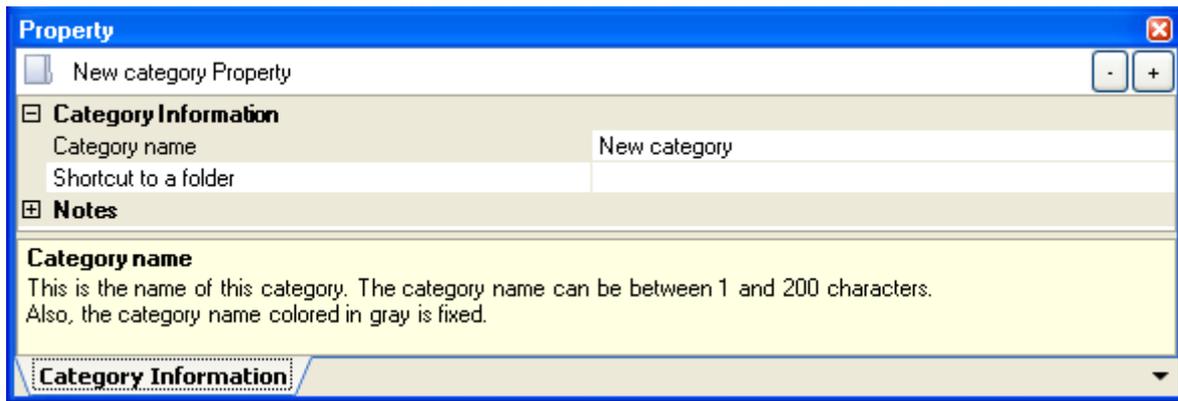


**[Description of each category]**

**(1)  [Category Information]**

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

| Category name | Specify the name of the category to categorize files. This property of the Files node and Build tool generated files node is displayed in gray and you cannot change the attribute. | |
|---|---|---|
| | Default | *Category name of files* |
| | How to change | Directly enter in the text box. |
| | Restriction | 1 to 200 characters |
| Shortcut to a folder | Specify a shortcut to a folder. If a relative path is specified, the reference point of the path is the main project or subproject folder. This property of the Files node and Build tool generated files node is not displayed. | |
| | Default | Blank |
| | How to change | Directly enter in the text box or edit by the Browse For Folder dialog box which appears when clicking the [...] button. |
| | Restriction | Up to 247 characters |

**(2)  [Notes]**

The detailed information on notes is displayed and the configuration can be changed.
This category of the Files node and Build tool generated files node is not displayed.

| Memo | Add memos to the category of files.<br>Add one item in one line.<br>The specified memo is displayed as the subproperty. | |
|---|---|---|
| | Default | Memo[*number of items*] |
| | How to change | Edit by the Text Edit dialog box which appears when clicking the [...] button.<br>For the subproperty, you can enter directly in the text box. |
| | Restriction | Up to 256 characters<br>Up to 256 items can be specified. |

**Editor panel**

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

See "CubeSuite+ Integrated Development Environment User's Manual: Coding" for details about this panel.

**Figure A-28.   Editor Panel**
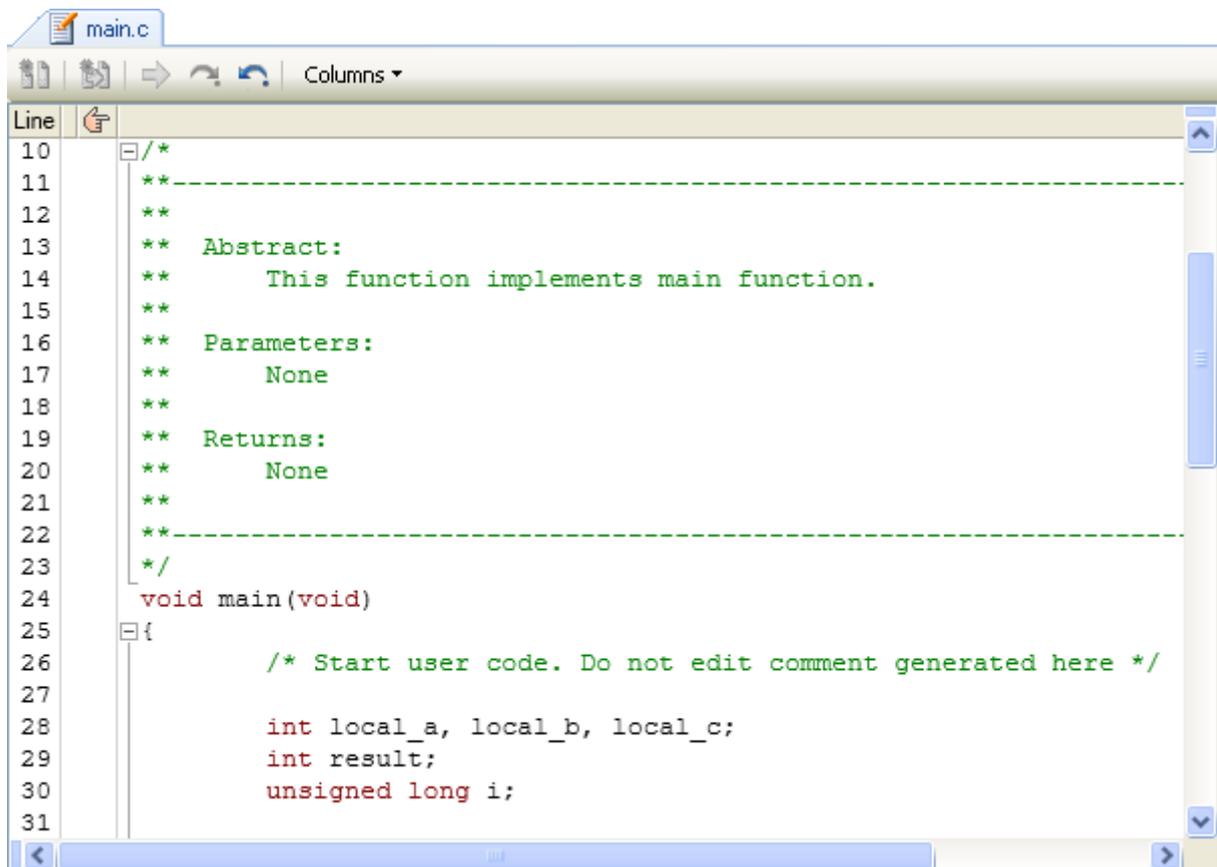
---

> **Output panel**

This panel is used to display the message that is output from the build tool/debug tool/each plug-in or the result of the Find In Files with the Find and Replace dialog box.
Messages are shown individually on the tab categorized by the output tool.
The [All Messages] tab consolidates and displays all output messages.

**Remark**    This panel can be zoomed in and out by `100%` ▼ in the tool bar, or by moving the mouse wheel forward or backward while holding down the [Ctrl] key.

**Figure A-29.   Output Panel**
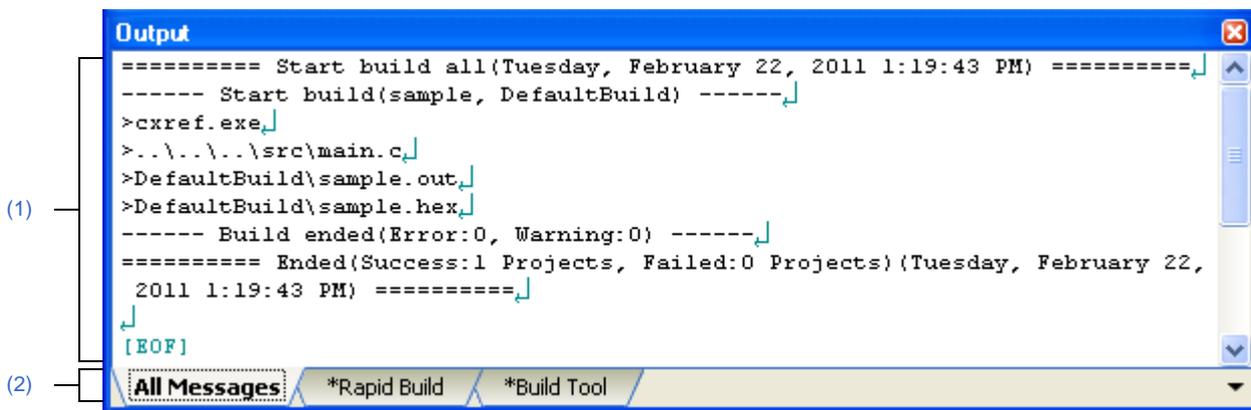


The following items are explained here.

- [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 | |

| Message type | Example (Default) | | | Description |
|---|---|---|---|---|
| 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. See the "CubeSuite+ Integrated Development Environment User's Manual: Analysis" for details about this tab. |
| Find and Replace | Displays the Find In Files result with the Find 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).

| Save Output-*tab name* | 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 *tab name* As...]. |
| | Note that this operation is invalid while building. |
| Save Output-*tab name* 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 Find and Replace dialog box with the [Quick Find] tab target. |
| Replace... | Opens the Find and Replace dialog box with the [Replace in Files] 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). |
| Jump to Previous Message | Jumps to the caret line in the editor indicated by the previous message (file and line). |
| Jump to Next Message | Jumps to the caret line in the editor indicated by the next message (file and line). |
| 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. |

---

Error List panel

This panel is used to list the information of messages output from the build tool (error information).

Remark   This panel can be zoomed in and out by [ 100% ▼] in the tool bar, or by moving the mouse wheel forward or backward while holding down the [Ctrl] key.

**Figure A-30.   Error List Panel**
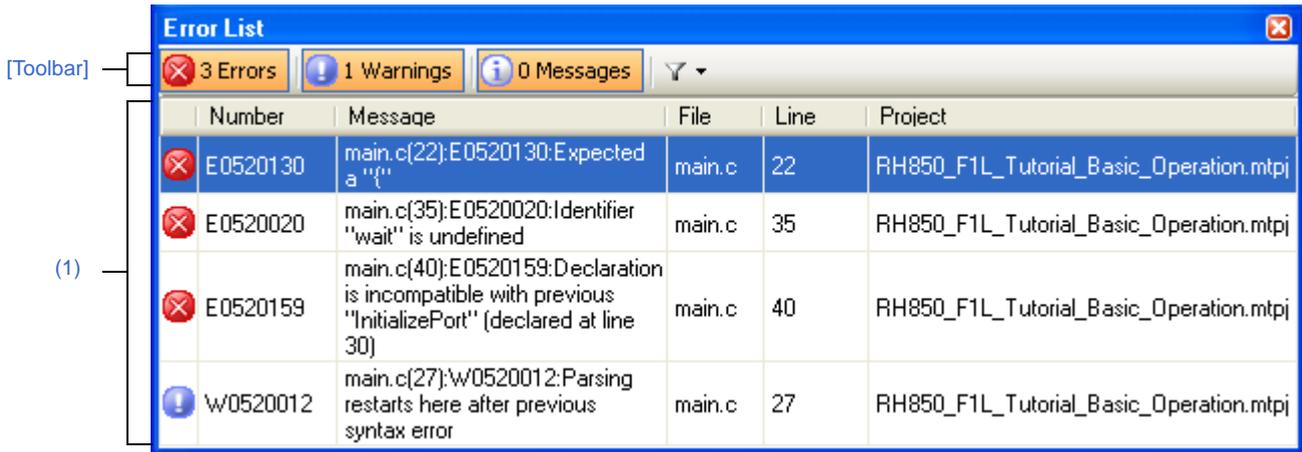


The following items are explained here.

- [How to open]
- [Description of each area]
- [[File] menu (Error List panel-dedicated items)]
- [[Edit] menu (Error List panel-dedicated items)]
- [Context menu]

**[How to open]**

- From the [View] menu, select [Error List].

**[Description of each area]**

**(1) Error information list area**

List the information of messages output from the build tool (error information).

The displayed error information can be sorted by clicking each column (ascending order / ascending order).

| Column | Description |
|---|---|
| - | Displays different icons in accord with the different types of message.<br>❌ : Error messaage<br>❗ : Warning message<br>ℹ️ : Normal message |
| Number | Displays the message number. |
| Message | Displays the message. |
| File | Displays the file name (without any path) of the target that the message indicates.<br>The absolute path will be shown in the tooltip. |

---

| Column | Description |
|--------|-------------|
| Line | Displays the line number of the target that the message indicates. |
| Project | Displays the project file name (without any path) of the target that the message indicates.<br>The absolute path will be shown in the tooltip. |

The error information displayed in this area will be deleted when one of the following events occurs.
- When the target file has been built without outputting the error or warning
- When the target file has been non build-target
- When the target file or subproject has been removed from the project
- When the build mode of the target project has been changed
- When the target project has been cleaned

This area has the following functions.

**(a) Tag jump**

When the error information is double-clicked, or the [Enter] key is pressed with the caret over the error information, 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 currently selected error information 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 currently on the panel can be saved in a text file (*.txt) by selecting [Save Error List As...] from [File] menu and opens the Save As dialog box.

**[Toolbar]**

| | |
|---|---|
| 3 Errors<br>3 Errors | Switches between display and non-display of error messages in the Error information list area.<br>The number following the icon is the number of error messages. |
| 1 Warnings<br>1 Warnings | Switches between display and non-display of warning messages in the Error information list area.<br>The number following the icon is the number of warning messages. |
| 0 Messages<br>0 Messages | Switches between display and non-display of normal messages in the Error information list area.<br>The number following the icon is the number of normal messages. |
| ▽ ▾ | Selects the target range of messages to be displayed in the Error information list area.<br>Current File: File opened with the currently active Editor panel<br>Active Project: Files registered in the active project<br>Opened File: All files opened with the Editor panel |
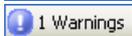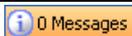
### [[File] menu (Error List panel-dedicated items)]

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

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

### [[Edit] menu (Error List panel-dedicated items)]

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

| | |
|---|---|
| Copy | Copies the selected error information to the clip board. |
| Select All | Selects all error information displayed on the panel. |
| Find... | Opens the Find and Replace dialog box with the [Find in Files] tab target. |
| Replace... | Opens the Find and Replace dialog box with the [Replace in Files] tab target. |

### [Context menu]

| | |
|---|---|
| Copy | Copies the selected error information to the clip board. |
| Select All | Selects all error information displayed on the panel. |
| Tag Jump | Jumps to the caret line in the editor indicated by the currently selected error information (file, line, and column). |
| Jump to Previous Message | Jumps to the caret line in the editor indicated by the error information prior to the currently selected error information (file and line). |
| Jump to Next Message | Jumps to the caret line in the editor indicated by the error information next to the currently selected error information (file and line). |
| Help for Message | Displays the help with regard to the message of the currently selected error information. <br><br> Note that the help is only for warning/error messages. |

---

| Change Microcontroller dialog box |
| --- |

This dialog box is used to change the microcontroller to be used in the project.

> **Remark**    See "2.7.13   Change the microcontroller" for details about changing the microcontroller.

**Figure A-31.   Change Microcontroller Dialog Box**
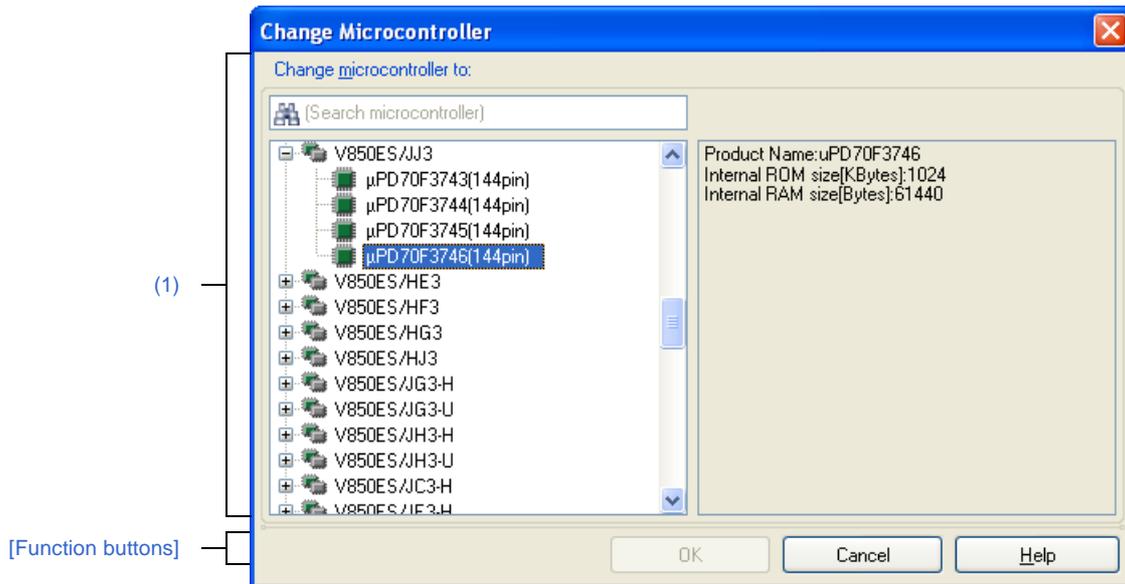


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]


**[How to open]**

- On the Project Tree panel, select the microcontroller node, and then select [Change microcontroller...] from the context menu.


**[Description of each area]**

**(1)  [Change microcontroller to] area**
Select a microcontroller to be used.

**(a)  Microcontroller search box**
You can search a microcontroller selected in the list of microcontrollers simply.

**(b)  List of microcontrollers**
Select a microcontroller.
The microcontrollers can be specified 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
- On-chip ROM size [Kbytes] [RX]

---

- Internal ROM size [Kbytes] [RH850][V850][R8C (Localised support)][RL78][78K0R][78K0]

- On-chip RAM size [Bytes] [RX]

- Internal RAM size [Bytes] [RH850][V850][R8C (Localised support)][RL78][78K0R][78K0]

- Additional Information**Note**


**Note**   Other information appears only when present.


**[Function buttons]**

| Button | Function |
|--------|----------|
| OK | Closes this dialog box and changes the microcontroller according to the designated condition.<br><br>Note that this button is invalid when the target microcontroller after the change is the same as the current microcontroller or is selected by its nickname from the list of microcontrollers. |
| Cancel | Cancels the designated condition and closes the dialog box. |
| Help | Displays the help of this dialog box. |

---

**Add File dialog box**

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

**Figure A-32.   Add File Dialog Box**
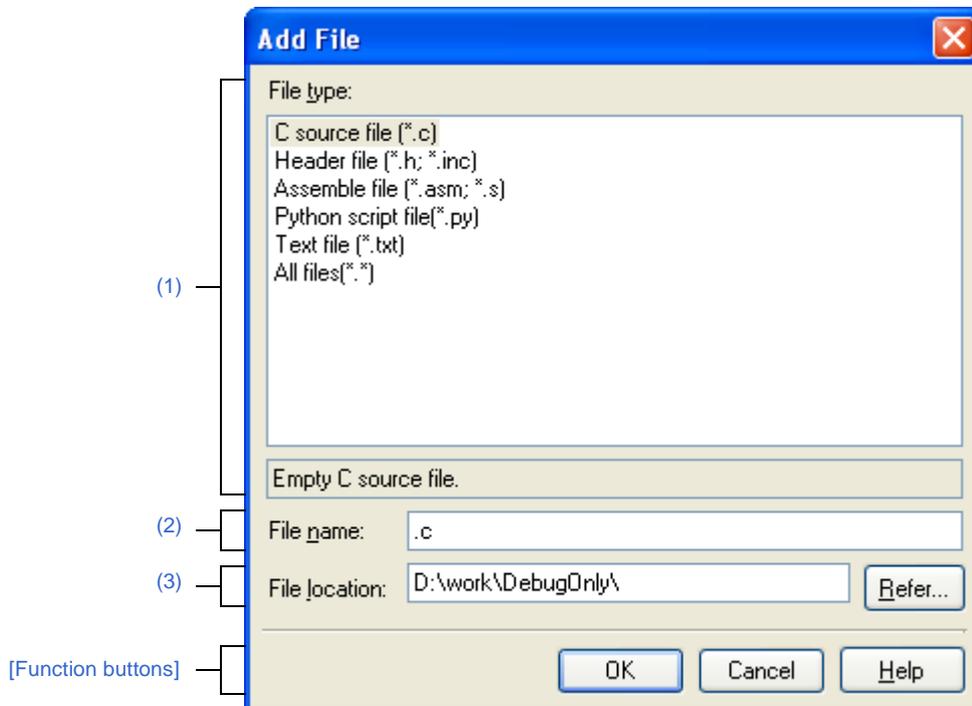


The following items are explained here.
   - [How to open]
   - [Description of each area]
   - [Function buttons]


**[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)
      - Python script file (*.py)
      - 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.

However, when this dialog box is opened from the context menu in the Category node (only when a shortcut to the

folder has been made and the folder exists), the path to the folder specified in the category is displayed.

**(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 loca-

tion] 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-33.   Add Folder and File Dialog Box**
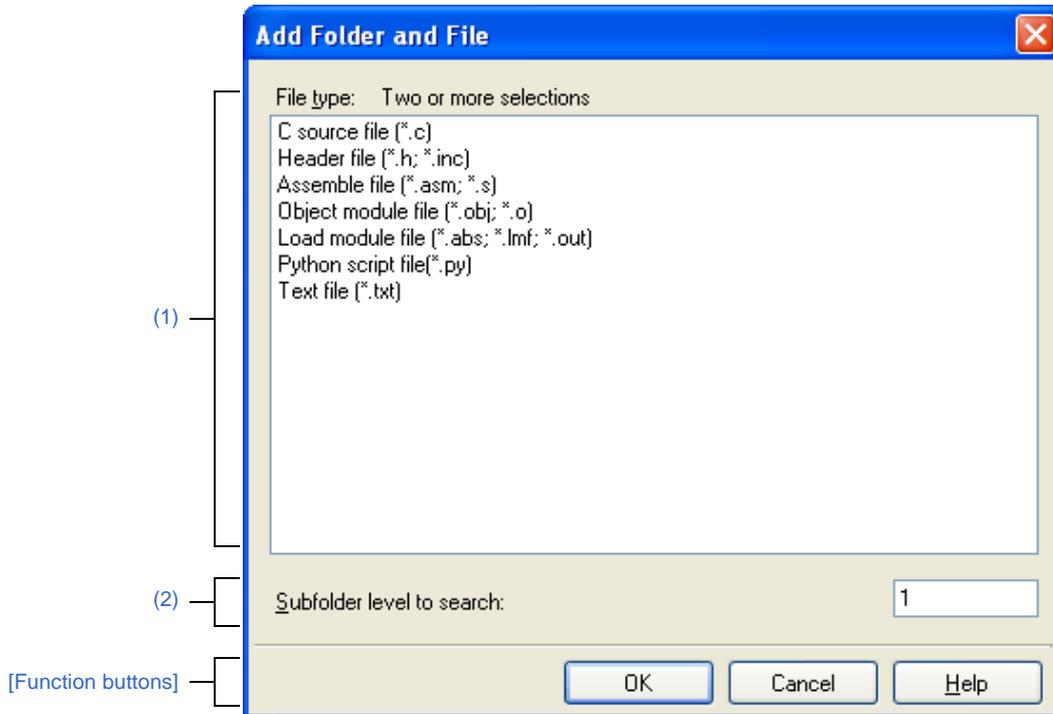


The following items are explained here.

- [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)
- Load module file (*.abs; *.lmf; *.out)
- Python script file (*.py)

- Text file (*.txt)

**(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. |

---

**Text Edit dialog box**

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

**Figure A-34.   Text Edit Dialog Box**
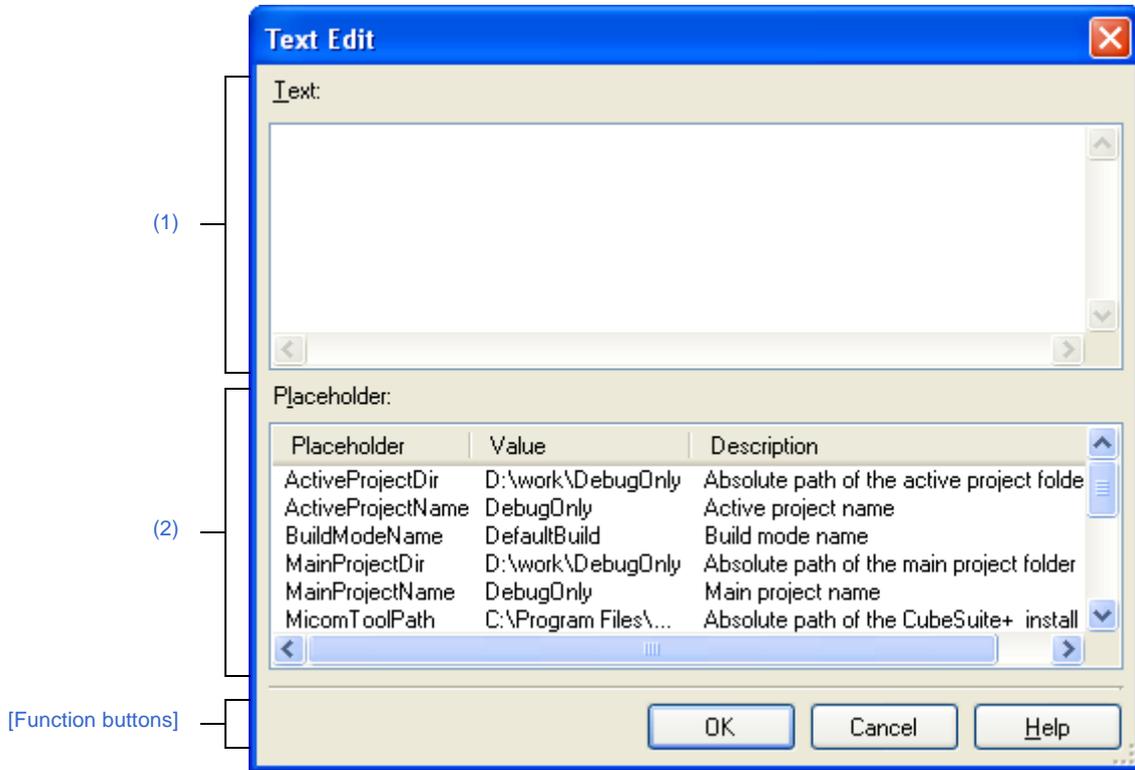


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[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]**
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 *maximum number of restriction in the property that called this dialog box* 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. |

**(2) [Placeholder]**

The list of placeholders which can be specified for the area that this dialog box is called from is displayed (ascending order).

Double click a row to surround the placeholder with percentage signs ("%"), and display it in [Text].

**(a) [Placeholder]**

This area displays the placeholder.

**(b) [Value]**

The string is displayed after replacement with the placeholder.

**(c) [Description]**

This area displays the description of the placeholder.

**Remark**   The placeholders which can be specified differ depending on the area that this dialog box is called from. For the specific placeholder, see the description of the area that this dialog box is called from.

**[Function buttons]**

| Button | Function |
|---|---|
| OK | Reflects the entered text to the text box that opened this dialog box and 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. |

---

| Encoding dialog box |
|---|

This dialog box is used to select a file-encoding.

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

**Figure A-35.   Encoding Dialog Box**
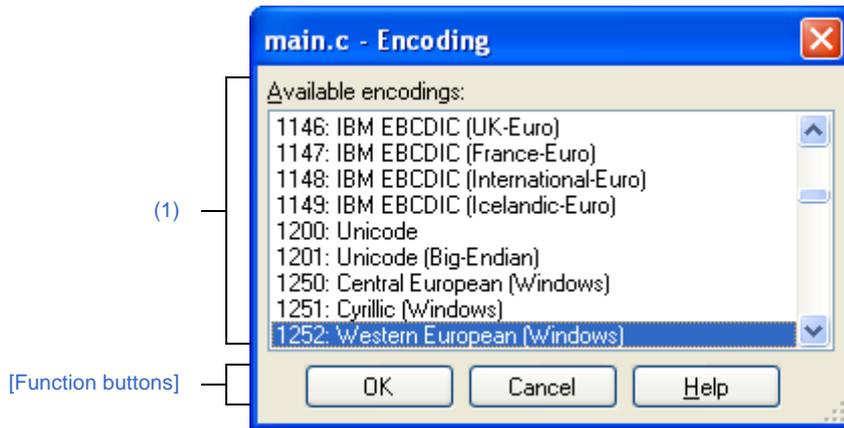


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

## [How to open]

- From the [File] menu, open the Open File dialog box by selecting [Open with Encoding...], and then click the
  [Open] button in the dialog box.

## [Description of each area]

**(1)  [Available encodings] area**
   Select the encoding to be set from this area.

## [Function buttons]

| Button | Function |
|---|---|
| OK | Opens the selected file in the Open File dialog box using a selected file encoding. |
| Cancel | Not open the selected file in the Open File dialog box and closes this dialog box. |
| Help | Displays the help for this dialog box. |

---

**Bookmarks dialog box**

This dialog box is used to display the position where a bookmark is to be set or to delete a bookmark.

**Figure A-36.　Bookmarks Dialog Box**
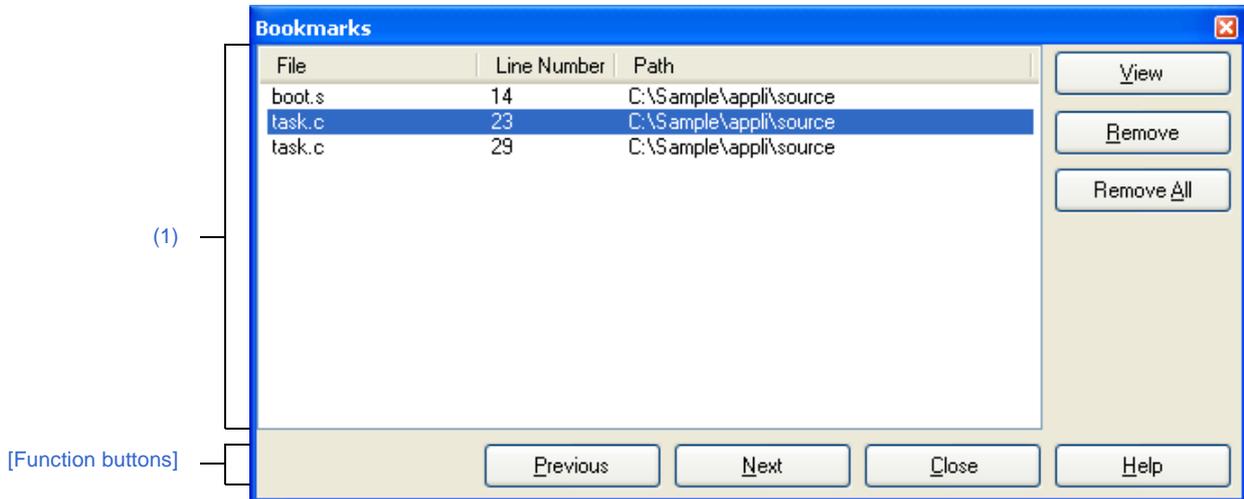


The following items are explained here.

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

**[How to open]**

- On the toolbar, click the [□] button.
- From the [Edit] menu, select [Bookmark] >> [List Bookmarks...].
- On the Editor panel, select [Bookmark] >> [List Bookmarks...] from the context menu.

**[Description of each area]**

**(1) Bookmark list area**

Display a list of bookmarks that have been registered.

The bookmarks are listed alphabetically for [File].  Bookmarks in the same file are listed in line number order.

When a bookmark is added to the Editor panel, a bookmark function is added.

In the bookmark list area, double-clicking on a line moves a caret to the corresponding position for the bookmark.

**(a)　[File]**

Display a file name (without any path) registered as a bookmark.

**(b)　[Line Number]**

Display a line number registered as a bookmark.

**(c)　[Path]**

Display a file path registered as a bookmark.

---

**(d)  Buttons**

| | |
|---|---|
| View | Moves a caret to the selected position for the bookmark. |
| | However, this button is disabled when no bookmark is selected, two or more bookmarks are selected, or no bookmark is registered. |
| Remove | Removes a selected bookmark.  When two or more bookmarks are selected, all of those selected are removed. |
| | However, this button is disabled when no bookmark is selected or no bookmark is registered. |
| Remove All | Removes all the registered bookmarks. |
| | This button is disabled when no bookmark is registered. |

**Caution**    **The registered bookmarks are not removed even if the Editor panel is closed.  However, when the Editor panel is closed without saving after a file is newly created, the registered bookmarks are removed.**

**[Function buttons]**

| Button | Function |
|---|---|
| Previous | Moves a caret to the position of the bookmark previous to the selected bookmark. |
| | This button is disabled in the following cases. |
| | - A bookmark shown in the first line has been selected. |
| | - No bookmark is selected. |
| | - Two or more bookmarks are selected. |
| | - No bookmark is registered. |
| | - Only one bookmark is registered. |
| Next | Moves a caret to the position of the bookmark next to the selected bookmark. |
| | This button is disabled in the following cases. |
| | - A bookmark shown in the last line has been selected. |
| | - No bookmark is selected. |
| | - Two or more bookmarks are selected. |
| | - No bookmark is registered. |
| | - Only one bookmark is registered. |
| Close | Closes this dialog box. |
| Help | Displays the help for this dialog box. |

---

**Go to Line dialog box**

---

This dialog box is used to move the caret to a specified line number, symbol, or address.

**Figure A-37. Go to Line Dialog Box**
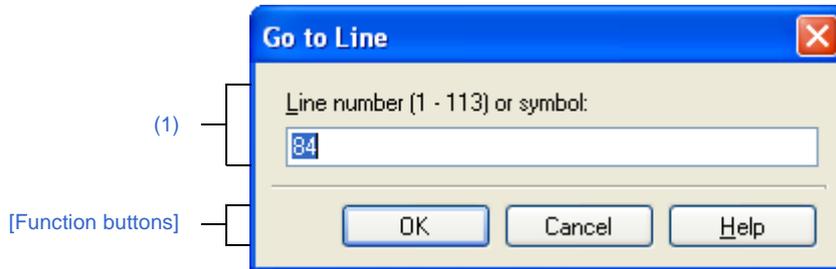
(1)

[Function buttons]

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

**[How to open]**

- From the [Edit] menu, select [Go To...].
- On the Editor panel, select [Go To...] from the context menu.

**[Description of each area]**

**(1) [Line number (*valid line range*) or symbol]**
"(*valid line range*)" shows the range of valid lines in the current file.
Specify the line number, symbol, or address that you want to move the caret to.
By default, the number of the line where the caret is currently located in the Editor panel is displayed.

**Remarks 1.** When a symbol (function name and variable name: [CC-RH], [CC-RX], [CX], or [NC30]) is speci-
fied, building must have been completed.
When the build tool used is CC-RH, CC-RX, CX, or NC30, cross-reference information must be
output. Execute a build after [Yes (-Xcref)] is selected in the following property of the build tool to
be used.
- [CC-RH][CX]
[Common Options] tab >> [Output File Type and Path] category >> [Output cross reference infor-
mation] property
- [CC-RX][NC30]
[Compile Options] tab >> [Others] category >> [Output cross reference information] property

**2.** When an address is specified, building must have been completed.
An address can be specified with a hexadecimal starting with "0x" or "0X". A decimal is interpreted
as a line number.

---

**[Function buttons]**

| Button | Function |
|--------|----------|
| OK | Places the caret at the start of the specified source line. |
| Cancel | Cancels the jump and closes this dialog box. |
| Help | Displays the help for this dialog box. |

**Jump to Function dialog box [CC-RH][CC-RX][CX][NC30]**

This dialog box is used to select a function to be jumped if there are some functions with the same names when a program jumps to the function specified on the Editor panel.

**Remarks 1.**   This dialog box is displayed only when there are some functions with the same names and [Yes (-Xcref)] is selected in the following property of the build tool to be used.
- [CC-RH][CX]
  [Common Options] tab >> [Output File Type and Path] category >> [Output cross reference information] property
- [CC-RX][NC30]
  [Compile Options] tab >> [Others] category >> [Output cross reference information] property

**2.**   This dialog box targets only files that have been registered in the project.

**Figure A-38.   Jump to Function Dialog Box**
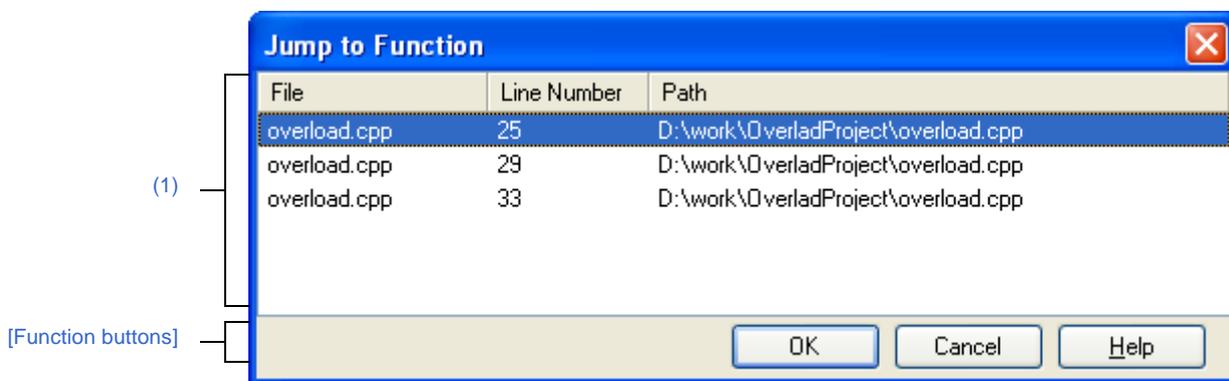


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[How to open]**

- On the Editor panel, select [Jump to Function] from the context menu.

**[Description of each area]**

**(1) Candidates in the jump destination display area**
List candidates in the jump destination.
Candidates are displayed in the alphabetical order of the names of [File]. If candidates are included in the same file, they are displayed in the order of line numbers.

**(a) [File]**
Display a file name (without any path) that a function is defined.

**(b) [Line Number]**
Display a line number that a function is defined.

**(c) [Path]**

Display a file path that a function is defined.

**[Function buttons]**

| Button | Function |
|--------|----------|
| OK | Jumps to the line that defines the target function after selecting the line in "(1)   Candidates in the jump destination display area" and clicking this button. |
| Cancel | Cancels the jump and closes this dialog box. |
| Help | Displays the help for this dialog box. |

---

**Find and Replace dialog box**

---

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

**Remark**    The ▶ button in [Search text] will be displayed only if this dialog box is opened from the Editor panel.

**Figure A-39.   Find and Replace Dialog Box**
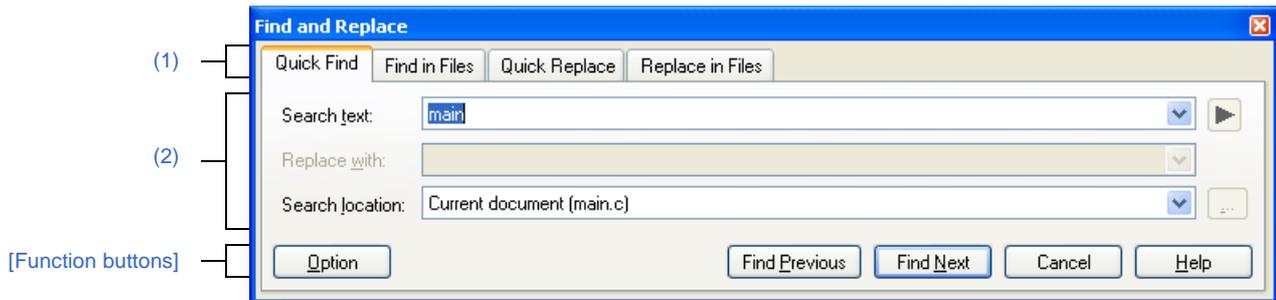


The following items are explained here.

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


## [How to open]

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


## [Description of each area]

**(1) Tab selection area**
Find/replace is switched when a tab is selected.
This dialog box has the following tabs.

- [Quick Find] tab
- [Find in Files] tab
- [Quick Replace] tab
- [Replace in Files] 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 find/replace.
Please see the description of the relevant buttons for details.

---

## [Quick Find] tab

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

**Remarks 1.** This tab will be disabled if you call the Find and Replace dialog box from a panel other than the Property panel, Output panel, or Editor panel.

**2.** The ▶ button in [Search text] will be displayed only if this dialog box is opened from the Editor panel.

**Figure A-40.　Find and Replace Dialog Box: [Quick Find] Tab**



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

## [How to open]

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

## [Description of each area]

**(1) [Search text]**
Designate characters to find.
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.

You can select the following wildcard by using the ▶ button. This is available when this dialog box is opened from the Editor panel and "Wild-card" is selected in [Search criteria].
- # Any single digit
- * Zero or more of any character

- ?  Any single character
- [!] Any one character not in the set
- [] Any one character in the set


You can select the following regular expressions by using the [▶] button. This is available when this dialog box is opened from the Editor panel and "Regular Expression" is selected in [Search criteria].
- $ End of line
- ( ) Group capture
- * Zero or more
- + One or more
- . Any single character
- [] Any one character not in the set
- [^] Any one character not in the set
- \ Escape special character
- \b Word boundary
- \n Line break
- \s Whitespace
- ^ Beginning of line
- | Or


**Remark**    For details on regular expressions, see "APPENDIX E   REGULAR EXPRESSIONS SYNTAX".


**(2)  [Replace with]**
This item is disabled.


**(3)  [Search location]**
Designate the location to find.
Select one of the following items from the drop-down list.

| Item | Operation |
| --- | --- |
| Selection area | Finds the selection in the search enabled panel which was active the last time. |
| | If this dialog box is opened from the Editor panel, or if there is no characters in selection in the panel which was last active, or the panel cannot be found, this item will be disabled. |
| Current document (*Panel Name*) | Finds in the panel which was last active and can be found. |
| | If the panel which was lastly active cannot be found 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 |
| --- | --- |
| Normal | Finds the characters designated in [Search text]. |

| Item | Operation |
|------|-----------|
| Wild-card | Finds using the wildcard designated in [Search text]. |
| Regular Expressions[Note] | Finds using the regular expressions designated in [Search text]. |

**Note**   This item is enabled only when the Editor panel is focused.

**(b) [File type]**

This item is disabled.

**(c) [Match case]**

| | |
|---|---|
| ☑ | Finds the designated characters in case-sensitive. |
| ☐ | Finds the designated characters in not case-sensitive (default). |

**(d) [Match whole word]**

| | |
|---|---|
| ☑ | Finds a designated exact word. |
| ☐ | Finds 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. |
| Find Previous | Finds 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]. The operation is the same as when the [Shift] + [Enter] key is pressed. |
| Find Next | Finds 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]. The operation is the same as when the [Enter] key is pressed. |
| 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, "*Search text* was not found." is displayed on the status bar of
the Main window.

**[Find in Files] tab**

In this tab, the designated characters are found 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.

**Remark** The ▶ button in [Search text] will be displayed only if this dialog box is opened from the Editor panel.

**Figure A-41. Find and Replace Dialog Box: [Find in Files] Tab**



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

**[How to open]**

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

**[Description of each area]**

**(1) [Search text]**

Designate characters to find.

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.

You can select the following wildcard by using the ▶ button. This is available when this dialog box is opened from the Editor panel and "Wild-card" is selected in [Search criteria].
- # Any single digit
- * Zero or more of any character
- ? Any single character

- [!] Any one character not in the set
- [] Any one character in the set


You can select the following regular expressions by using the [▶] button. This is available when this dialog box is opened from the Editor panel and "Regular Expression" is selected in [Search criteria].
- $ End of line
- ( ) Group capture
- * Zero or more
- + One or more
- . Any single character
- [] Any one character not in the set
- [^] Any one character not in the set
- \ Escape special character
- \b Word boundary
- \n Line break
- \s Whitespace
- ^ Beginning of line
- | Or


> **Remark**    For details on regular expressions, see "APPENDIX E   REGULAR EXPRESSIONS SYNTAX".


**(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 document (*Panel Name*) | Finds within the current Editor panel. |
| All open documents | Finds all the opening the Editor panel. <br> If no file is opened in the Editor panel, this item is disabled. |
| Active project | Finds the text file included in the active project. <br> When [File type] is specified, searches only the specified type. <br> Note that is the current project does not exist, this item is disabled. |
| Main project and subprojects | Finds within the text file included in the main project and subproject. <br> When [File type] is specified, searches only the specified type. <br> Note that is the current project does not exist, this item is disabled. |
| *Folder Name* | Finds 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, finds 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 |
|---|---|
| Normal | Finds the characters designated in [Search text]. |
| Wild-card | Finds using the wildcard designated in [Search text]. |
| Regular Expressions[Note] | Finds using the regular expressions designated in [Search text]. |

**Note**   This item is enabled only when the Editor panel is focused.

**(b) [File type]**

Specify File types to search.

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

| Item | Operation |
|---|---|
| Source files (Extensions[Note]) | Files to find are limited to the source files. |
| *.txt | Files to find are limited to the text files. |
| *.* | Finds 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) [Match case]**

| ☑ | Finds the designated characters in case-sensitive. |
|---|---|
| ☐ | Finds the designated characters in not case-sensitive (default). |

**(d) [Match whole word]**

| ☑ | Finds a designated exact word. |
|---|---|
| ☐ | Finds 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. |
| Find All | Finds 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.

Remark    This tab will be disabled if you call the Find and Replace dialog box from a panel other than the Editor panel.

**Figure A-42.   Find and Replace Dialog Box: [Quick Replace] Tab**
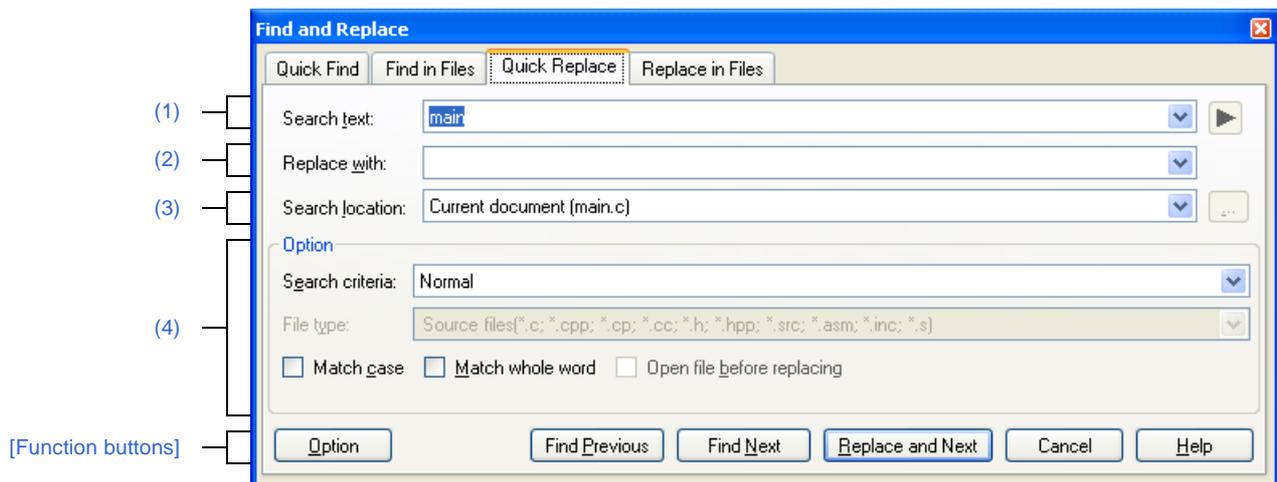


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[How to open]**

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

**[Description of each area]**

**(1)  [Search text]**
Designate characters to find.
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.
Words (variable/function) at the caret position are shown by default.

You can select the following wildcard by using the  ▶  button. This is available when "Wild-card" is selected in [Search criteria].
- # Any single digit
- * Zero or more of any character
- ?  Any single character
- [!] Any one character not in the set
- [] Any one character in the set

You can select the following regular expressions by using the [▶] button. This is available when "Regular Expression" is selected in [Search criteria].

- $ End of line
- ( ) Group capture
- * Zero or more
- + One or more
- . Any single character
- [] Any one character not in the set
- [^] Any one character not in the set
- \ Escape special character
- \b Word boundary
- \n Line break
- \s Whitespace
- ^ Beginning of line
- | Or

**Remark**    For details on regular expressions, see "APPENDIX E   REGULAR EXPRESSIONS SYNTAX".

**(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]**

The following item will appear in the drop-down list.

| Item | Operation |
|---|---|
| Current document (*Panel Name*) | Finds in the panel which was last active and can be found. |
| | If the panel which was lastly active cannot be found 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 |
|---|---|
| Normal | Finds the characters designated in [Search text]. |
| Wild-card | Finds using the wildcard designated in [Search text]. |
| Regular Expressions | Finds using the regular expressions designated in [Search text]. |

**(b)  [File type]**

This item is disabled.

**(c) [Match case]**

| | |
|---|---|
| ☑ | Finds the designated characters in case-sensitive. |
| ☐ | Finds the designated characters in not case-sensitive (default). |

**(d) [Match whole word]**

| | |
|---|---|
| ☑ | Finds with a designated exact word. |
| ☐ | Finds 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. |
| Find Previous | Finds from the current caret position to the top of the file with the designated criteria. Selects the characters that are found and moves the caret[Note]. |
| Find Next | Finds from the current caret position to the end of the file with the designated criteria. Selects the characters that are found and moves the caret[Note]. |
| Replace and Next (Replace and Previous) | Replaces the selected characters to the characters to be replaced then searches the next (previous) 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, "*Search text* was not found." is displayed on the status bar of
the Main window.

**[Replace in Files] 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.

> **Remarks 1.** This tab will be disabled if you call the Find and Replace dialog box from a panel other than the Editor panel.
>
> **2.** The ▶ button in [Search text] will be displayed only if this dialog box is opened from the Editor panel.

**Figure A-43.   Find and Replace Dialog Box: [Replace in Files] Tab**
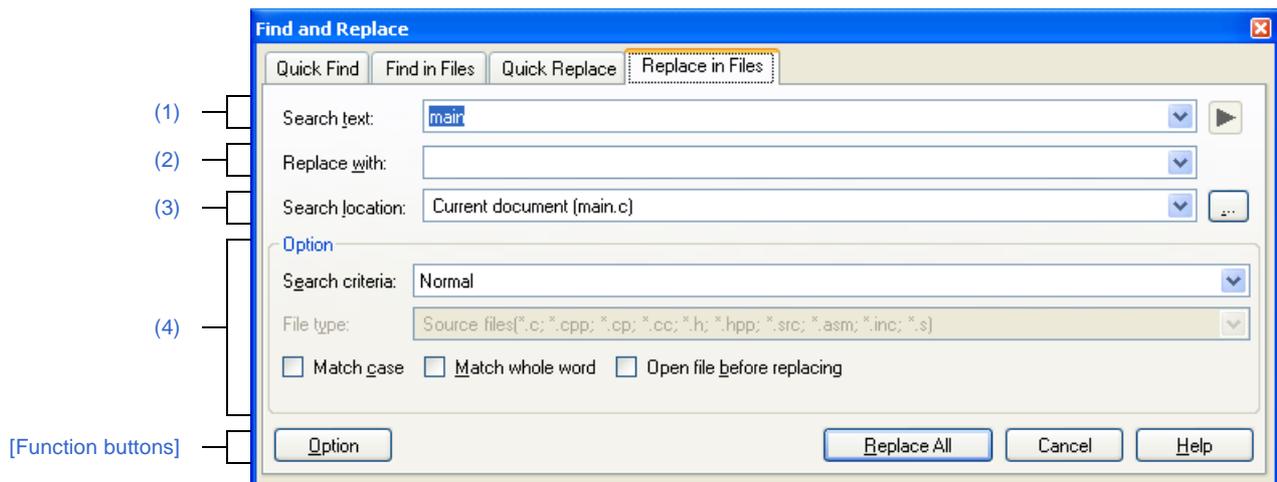


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[How to open]**

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

**[Description of each area]**

**(1)  [Search text]**
Designate characters to find.
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.

You can select the following wildcard by using the ▶ button. This is available when this dialog box is opened from the Editor panel and "Wild-card" is selected in [Search criteria].
- # Any single digit
- * Zero or more of any character

- ?  Any single character
- [!] Any one character not in the set
- [] Any one character in the set

You can select the following regular expressions by using the ▶ button. This is available when this dialog box is opened from the Editor panel and "Regular Expression" is selected in [Search criteria].

- $ End of line
- ( ) Group capture
- * Zero or more
- + One or more
- . Any single character
- [] Any one character not in the set
- [^] Any one character not in the set
- \ Escape special character
- \b Word boundary
- \n Line break
- \s Whitespace
- ^ Beginning of line
- | Or

> **Remark**    For details on regular expressions, see "APPENDIX E   REGULAR EXPRESSIONS SYNTAX".

**(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 find.

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 document (*Panel Name*) | Finds within the current Editor panel. |
| All open documents | Finds within all the opening the Editor panel.<br>If no file is opened in the Editor panel, this item is disabled. |
| Active project | Finds within the text file included in the active project.<br>When [File type] is specified, finds only the specified type.<br>Note that is the current project does not exist, this item is disabled. |
| Main project and subprojects | Finds within the text file included in the main project and subproject.<br>When [File type] is specified, finds only the specified type.<br>Note that if the current project does not exist, this item is disabled. |
| *Folder Name* | Finds 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 |
|---|---|
| Normal | Finds the characters designated in [Search text]. |
| Wild-card | Finds using the wildcard designated in [Search text]. |
| Regular Expressions[Note] | Finds using the regular expressions designated in [Search text]. |

**Note**   This item is enabled only when the Editor panel is focused.

**(b) [File type]**

Specify File types to search.

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

| Item | Operation |
|---|---|
| Source files (*Extensions*[Note]) | Files to find are limited to the source files. |
| Text files (*.txt) | Files to find are limited to the text files. |
| All files (*.*) | Finds all the files. |

**Note**   Shows extensions of the source file added to the Project Tree panel.

Note that the finds 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) [Match case]**

| | |
|---|---|
| ☑ | Finds with the designated characters in case-sensitive. |
| ☐ | Finds with the designated characters in not case-sensitive (default). |

**(d) [Match whole word]**

| | |
|---|---|
| ☑ | Finds with a designated exact word. |
| ☐ | Finds with at least one of the words (default). |

**(e)  [Open file before replacing]**

| | |
|---|---|
| ☑ | Replace is done after opening the file to find/replace characters in the Editor panel. |
| ☐ | Replace is done without opening the file to find/replace characters (default). |

**[Function buttons]**

| Button | Function |
|---|---|
| Option | Switches between display/hide the [Option] area in this tab. |
| Replace All | Finds 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. |

---

**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-44. Save Settings Dialog Box**
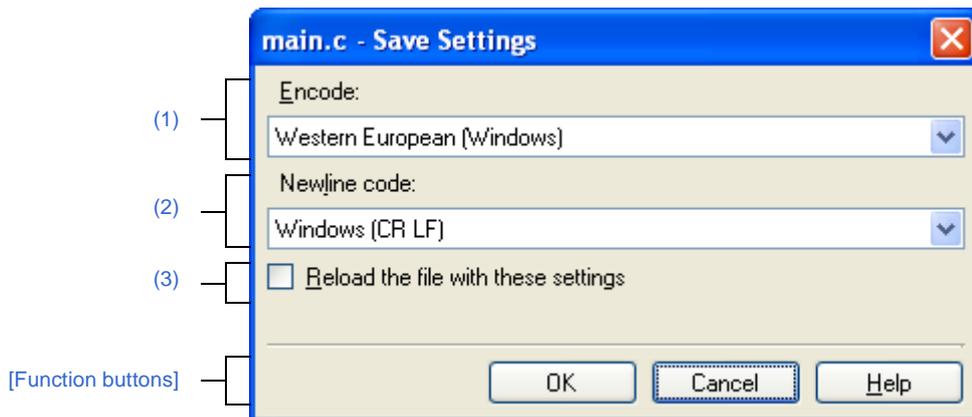


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[How to open]**

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

**[Description of each area]**

**(1) [Encode]**
Select the encoding to be set from the drop-down list.
The items of the drop-down list are displayed according to the following sequence.
Note that the same encoding and encoding which are not supported by the current OS will not be displayed.
    - *Current encoding of the file (default)*
    - *Default encoding of the current OS*
    - *Most recently used encodings (maximum 4)*
    - *Popular encodings for current locale*
      e.g. for United States locale it will be:
        - Western European (Windows)
        - Unicode (UTF-8)
    - *All other encodings supported by the OS (in alphabetical order)*

**(2) [Newline code]**
Select the newline code to be set from the drop-down list.
You can select any of items below.
    - Windows (CR LF)
    - Macintosh (CR)

---

- Unix (LF)

An active newline entry is selected by default.

**(3)  [Reload the file with these settings]**

| | |
|---|---|
| ☑ | Reloads the file with the selected encoding and newline code when the [OK] button is clicked. |
| ☐ | Does not reload the file when the [OK] button is clicked (default). |

**[Function buttons]**

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

---

**Print Preview window**

This window is used to preview the file currently being displayed in the Editor panel before printing.

**Remark**   This window can be zoomed in and out by moving the mouse wheel forward or backward while holding down the [Ctrl] key.

**Figure A-45.   Print Preview Window**
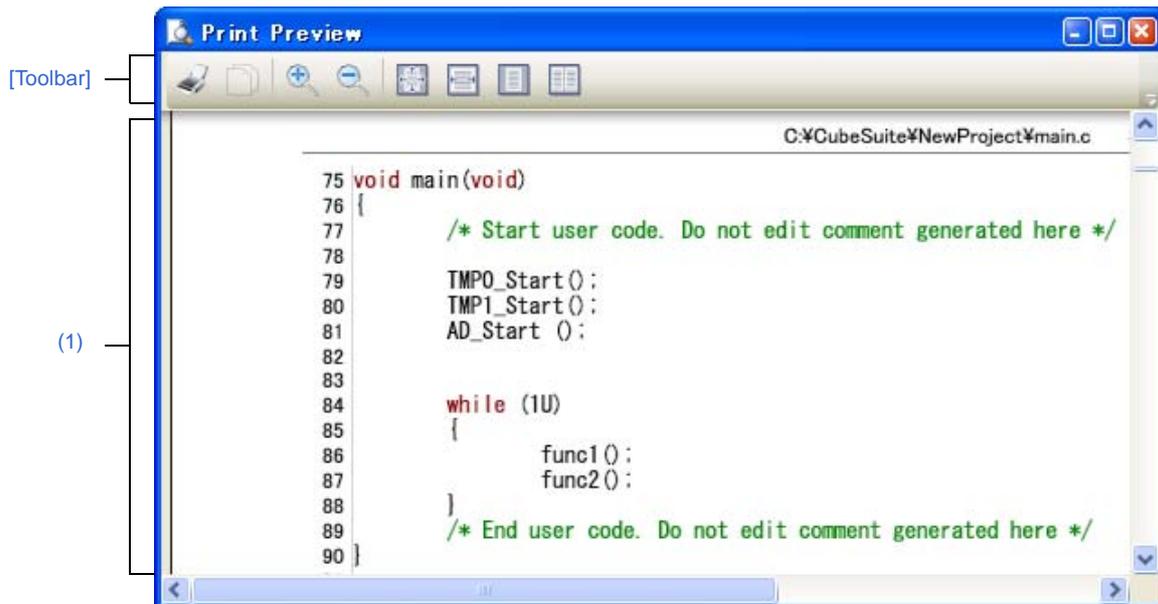


The following items are explained here.
- [How to open]
- [Description of each area]
- [Toolbar]
- [Context menu]

**[How to open]**

- Focus the Editor panel, and then select [Print Preview] from the [File] menu.

**[Description of each area]**

**(1)  Preview area**
This window displays a form showing a preview of how and what is printed.
The file name (fully qualified path) and the page number are displayed at the page header and page footer.
The display differs according to whether the debug tool is or is not connected, and when it is connected, to whether the display is in normal display mode or mixed display mode.  Note, however, that columns that are hidden on the Editor panel are not displayed (these columns are not printed).

---

**[Toolbar]**

|  | |
|---|---|
|  | Opens the Print dialog box provided by Windows to print the current Editor panel as shown by the print preview form. |
|  | Copies the selection into the clipboard. |
|  | Increases the size of the content. |
|  | Decreases the size of the content. |
|  | Displays the preview at 100-percent zoom (default). |
|  | Fits the preview to the width of this window. |
|  | Displays the whole page. |
|  | Displays facing pages. |

**[Context menu]**

| | |
|---|---|
| Increase Zoom | Increases the size of the content. |
| Decrease Zoom | Decreases the size of the content. |

---

Dependent Projects Settings dialog box

This dialog box is used to reference/set the dependent projects.

**Figure A-46.   Dependent Projects Settings Dialog Box**
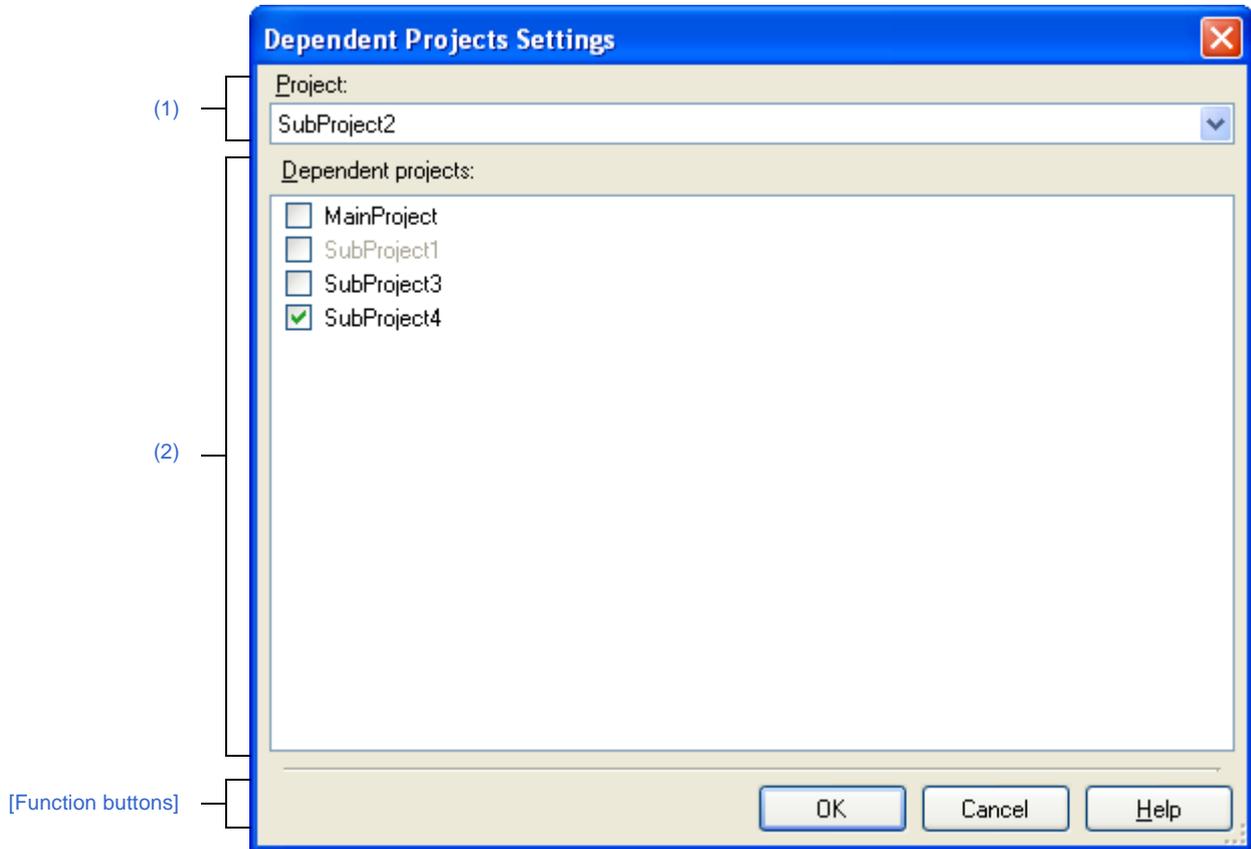


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[How to open]**

- From the [Project] menu, select [Dependent Projects Settings...].

Caution    **This item is only enabled when a project where a subproject exists is opened.**

**[Description of each area]**

**(1)  [Project]**
Select a project (dependent-source project) to be targeted when a dependent project is set.
The active project is selected by default.

**(2)  [Dependent projects]**
Select a project to be referenced as the dependent project at build in the check box.
A cyclic-reference project is displayed in gray because it cannot be selected.

---

**Remark**   When an application project is related to the RH850 boot loader project, the boot loader project and the application project are fixed to the dependent-source project and the dependent project, respectively, and they cannot be changed.

Note that the boot loader project and the application project are related in the Select Constituent Application Projects dialog box that is opened from the [Constituent application projects] property of the Configuration Tool for Multi-core node.

### [Function buttons]

| Button | Function |
|---|---|
| OK | Applies all setting and closes this dialog box. |
| Cancel | Cancels the designated condition and closes the dialog box. |
| Help | Displays the help of this dialog box. |

> **Build Mode Settings dialog box**

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

**Figure A-47.   Build Mode Settings Dialog Box**
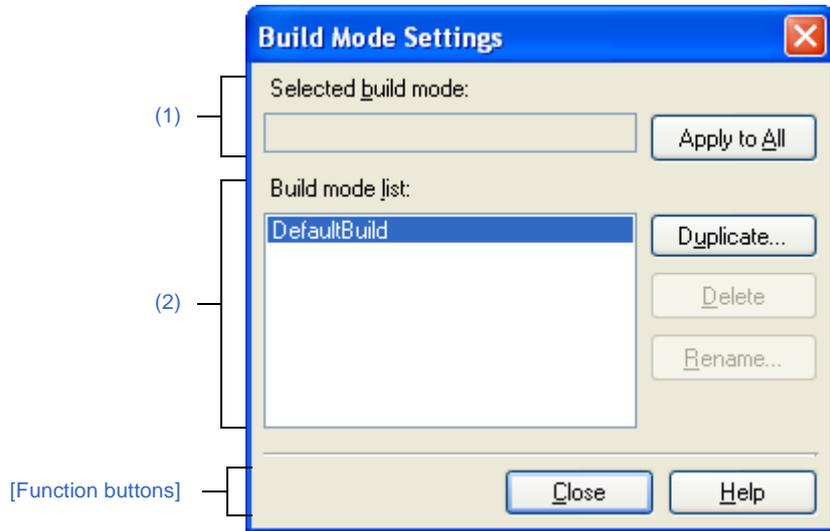


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[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... | Duplicates the selected build mode.<br><br>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.<br><br>When the build mode with "*" mark does not exist in the main project or subproject and duplicate the build mode, "DefaultBuild" will be duplicated.<br><br>Up to 20 build modes can be added. |
|---|---|
| Delete | Deletes the selected build mode.<br><br>Note that "DefaultBuild" cannot be deleted.<br><br>If the currently set build mode is deleted, "DefaultBuild" is set. |
| Rename... | Renames the selected build mode.<br><br>Rename the build mode with entered name in the opening the Character String Input dialog box. |

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

**Remarks 1.**   Up to 127 characters can be 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. |

---

**Character String Input dialog box**

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

**Figure A-48.　Character String Input Dialog Box**
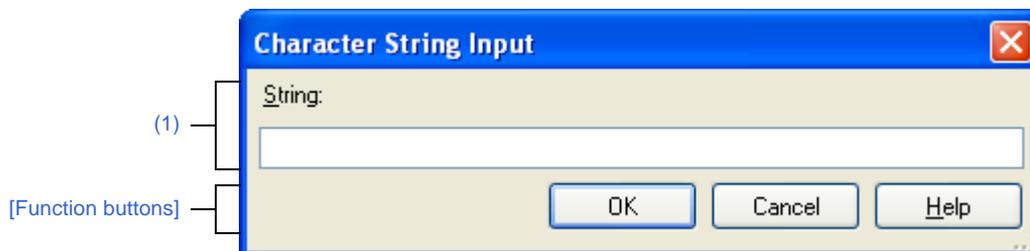


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[How to open]**

- On the Project Tree panel, after selecting a file, open the message dialog box by selecting [Change Extension...] from the context menu.
  And then click the [Yes] button in the dialog box.
- 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 *maximum number of restriction in the property that called this dialog box* 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. |

---

| Button | Function |
|--------|----------|
| 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-49.   Batch Build Dialog Box**
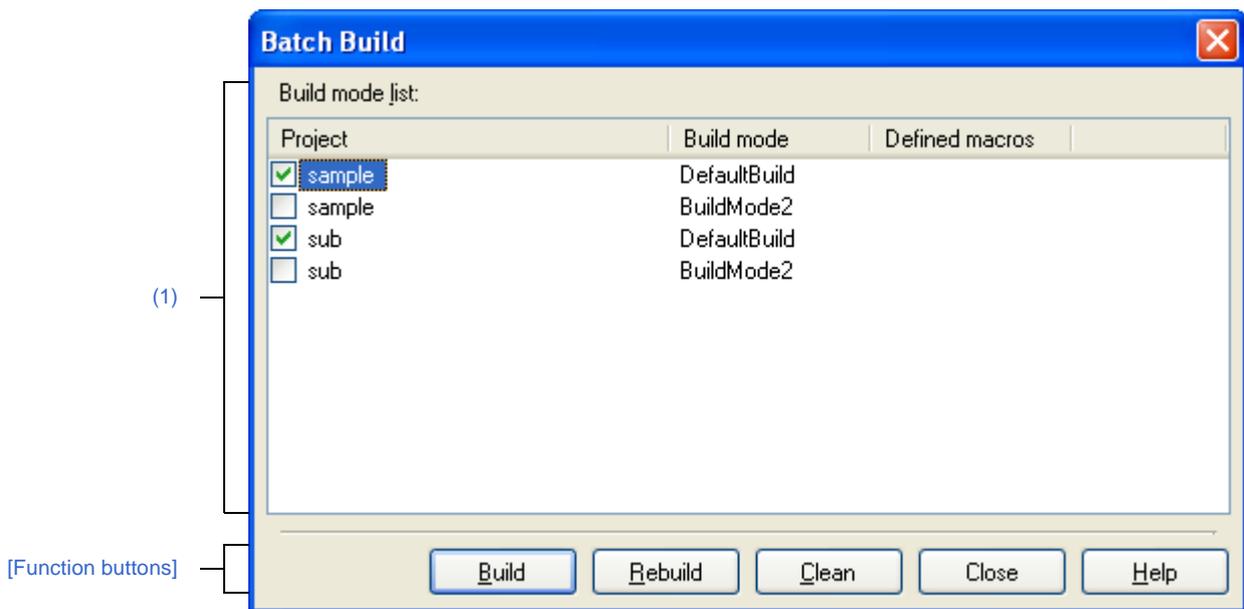


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]


**[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.

Remarks 1.　　Use one of the following methods to select multiple rows.

　　-Left-click the mouse with pressing the [Ctrl] key.

　　-Left-click the mouse with pressing the [Shift] key.

　　-Drag the mouse to select a range of rows.

Pressing the space key while a row is selected switches the selected/non-selected state of the checkbox.

　　2.　　Clicking on the header of each column sorts the entries (in ascending order by default).

Clicking on the same header again sorts the entries in the reverse order.

**[Function buttons]**

| Button | Function |
|---|---|
| Build | Closes this dialog box and runs builds of the selected projects in the respective build modes.<br>The execution result of the builds is displayed on the Output panel.<br>After the builds are complete, the build mode configuration restores to the one before this dialog box was opened.<br>If any project is not selected, this button will be disabled. |
| Rebuild | Closes this dialog box and runs rebuilds of the selected projects in the respective build modes.<br>The execution result of the rebuilds is displayed on the Output panel.<br>After the rebuilds are complete, the build mode configuration restores to the one before this dialog box was opened.<br>If any project is not selected, this button will be disabled. |
| Clean | Closes this dialog box and deletes the files which are built in the respective build modes set for the selected projects.<br>The execution result of the cleans is displayed on the Output panel.<br>After the cleans are complete, the build mode configuration restores to the one before this dialog box was opened.<br>If any project is not selected, this button will be disabled. |
| Close | 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-50.   Progress Status Dialog Box**
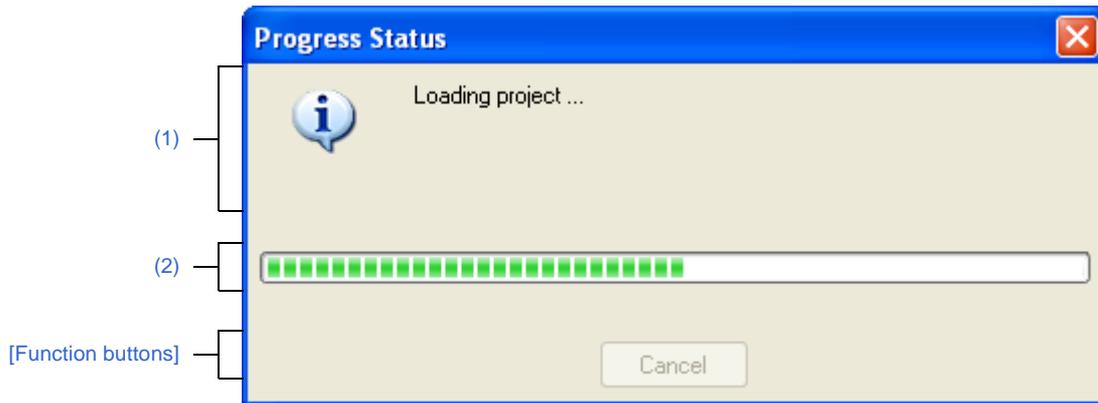


The following items are explained here.

- [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. <br> Note that if the process termination is impossible, this button is disabled. |

---

**Pack Settings dialog box**

This dialog box is used to pack and save the project and this product.

**Figure A-51.   Pack Settings Dialog Box**
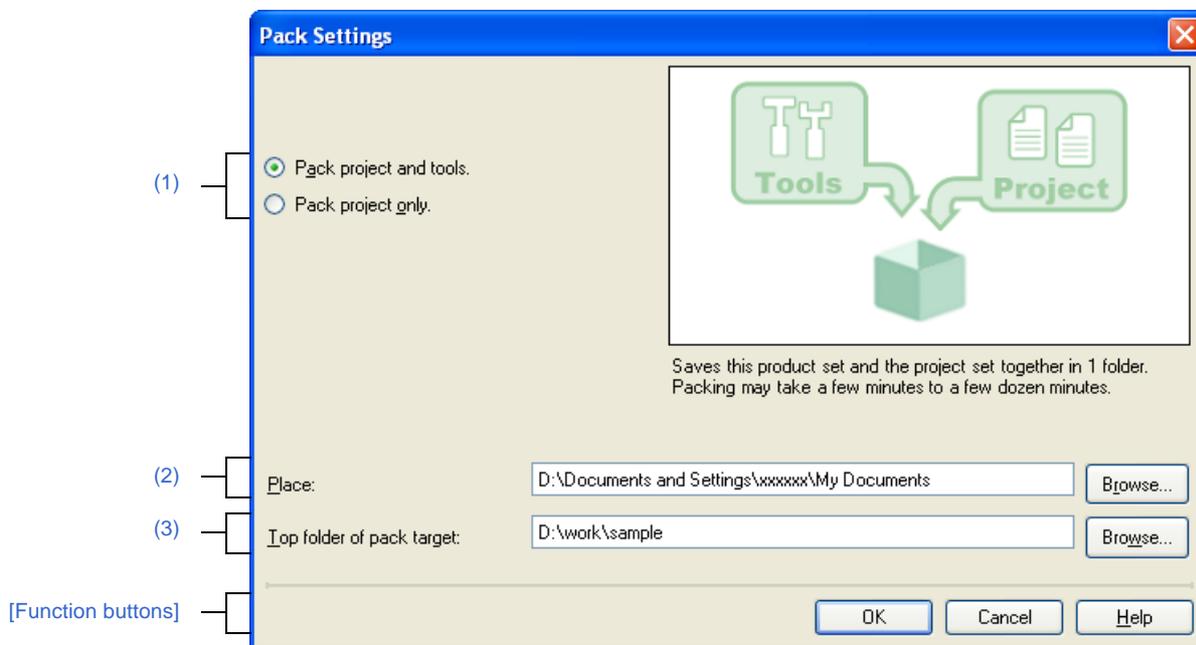


The following items are explained here.

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

**[How to open]**

- On the Project Tree panel, select the Project node, and then select [Save Project and Development Tools as Package...] from the context menu.

**[Description of each area]**

**(1)  Pack type selection area**
Select the pack type.

| Pack project and tools | Packs the project and this product (default). |
| --- | --- |
| Pack project only | Packs the project only. |

**(2)  [Place] area**
Specify the location in which to save the project and this product.
Enter the absolute path directly, or click the [Browse...] button, and select the location via the Browse For Folder dialog box.

**(a) 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] area. 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 project folder. | A folder inside the project folder is specified. |
| The folder that you will save the project is the same to the project folder. Set another place. | The target project folder for packing is in the folder specified as the location to save the project (Example: A project folder is specified as the top folder of the packing target and a folder above the project folder is specified as the location to save the project folder). |

**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 ▶.

**(3) [Top folder of pack target] area**
Specify the top folder of the packing target.
Enter the absolute path directly, or click the [Browse...] button, and select the location via the Browse For Folder dialog box.

**(a) 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 [Top folder of pack target] area. 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. |
| Specify a folder that contains the project folder. | A folder that does not contain the 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 packs and saves the project and this product to the designated condition. |
| Cancel | Cancels the designated condition and closes the dialog box. |
| Help | Displays the help of this dialog box. |

---

**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-52.   Option Dialog Box**
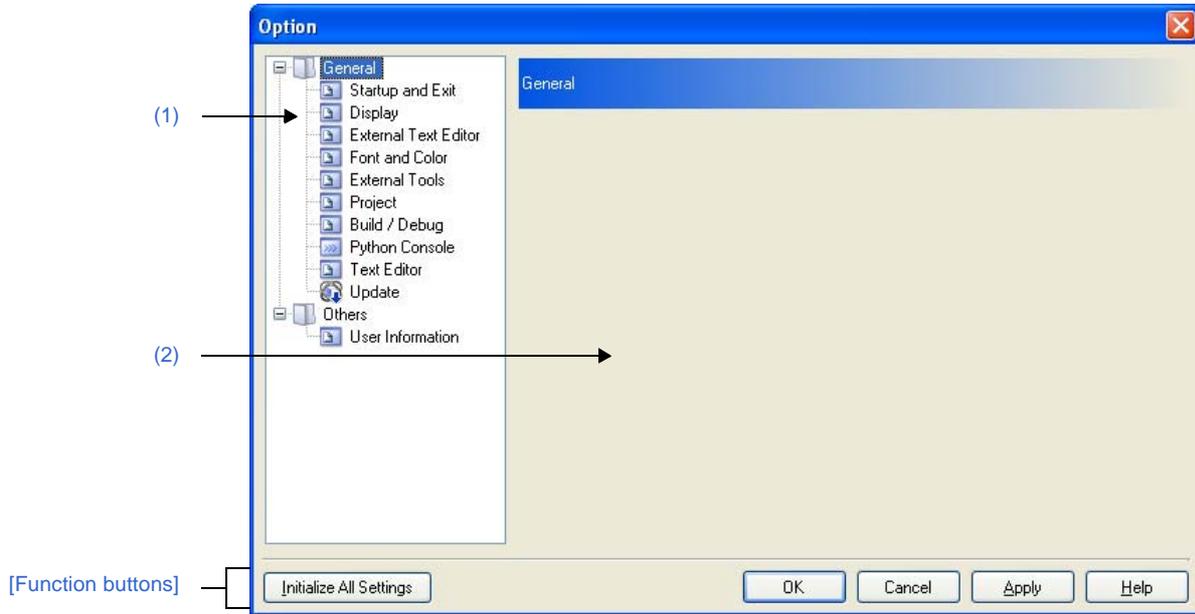


The following items are explained here.

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

**[How to open]**

- From the [Tool] menu, select [Options...].

**[Description of each area]**

**(1)  Category selection area**

Select the items to configure from the following categories.

| Category | Description |
|---|---|
| [General - Startup and Exit] category | Configure startup and shutdown. |
| [General - Display] category | Configure messages from the application. |
| [General - External Text Editor] category | Configure the external text editor. |
| [General - Font and Color] category | Configure the fonts and colors shown on each panel. |
| [General - External Tools] category | Configure the startup of external tools. |
| [General - Project] category | Configure the project. |
| [General - Build/Debug] category | Configure building and debugging. |
| [General - Python Console] category | Configure the Python console. |

---

| Category | Description |
|---|---|
| [General - Text Editor] category | Configure the text editor. |
| [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.<br><br>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-53.   Option Dialog Box ([General - Startup and Exit] Category)**
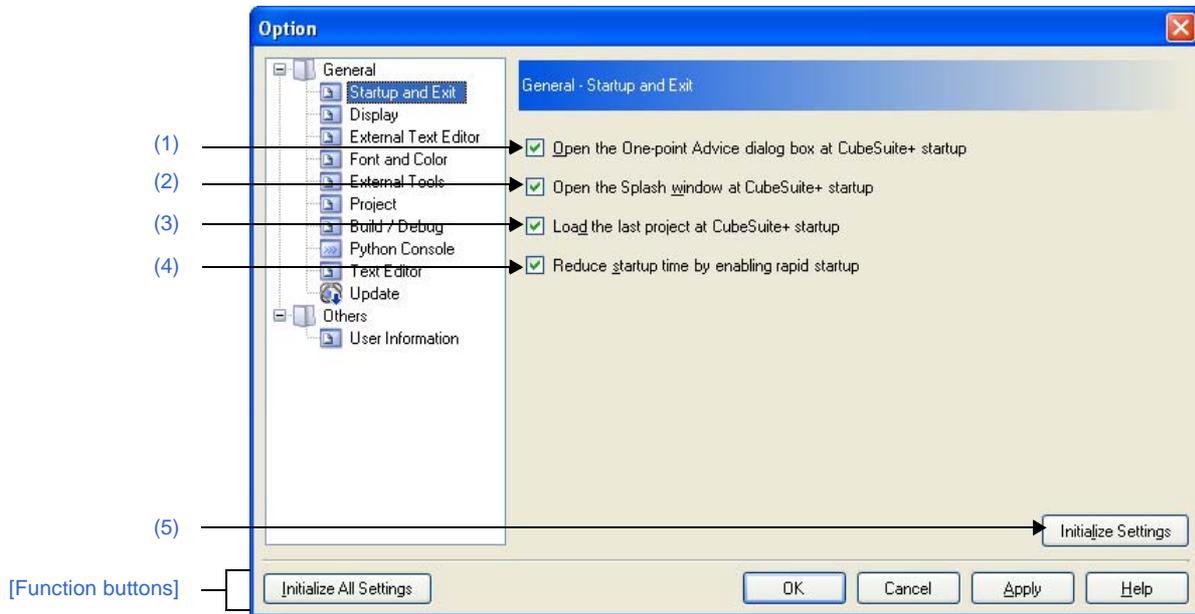


The following items are explained here.

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

**[How to open]**

- From the [Tool] menu, select [Options...].

**[Description of each area]**

**(1) [Open the One-point Advice dialog box at CubeSuite+ startup]**

| | |
|---|---|
| ☑ | Shows the FormOnePoint dialog box on startup (default). |
| ☐ | Does not show the FormOnePoint dialog box on startup. |

**(2) [Open the Splash window at CubeSuite+ startup]**

| | |
|---|---|
| ☑ | Shows the Splash window on startup (default). |
| ☐ | Does not show the Splash window on startup. |

**(3) [Load the last project at CubeSuite+ startup]**

| | |
|---|---|
| ☑ | Automatically loads the last project on startup (default). |
| ☐ | Does not automatically load the last project on startup. |

**(4) [Reduce startup time by enabling rapid startup]**

| | |
|---|---|
| ☑ | Reduces startup time by enabling rapid startup (default). |
| ☐ | Disables the rapid start feature. |

**Remark**    See "2.10.1   Use rapid start" for details about the rapid start.

**Caution**    **This item is only enabled when this product is installed using the installer. It is disabled when a packaged item is being used.**

**(5) 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-54.   Option Dialog Box ([General - Display] Category)**
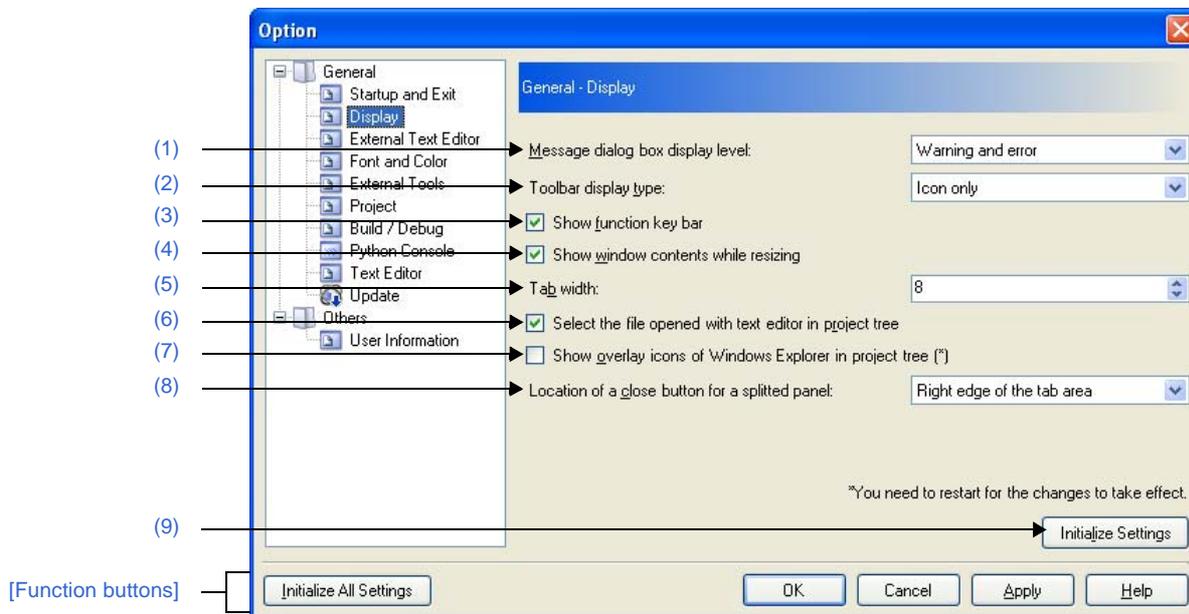


The following items are explained here.

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

### [How to open]

- From the [Tool] menu, select [Options...].

### [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]**

| | |
|---|---|
| ☑ | Displays the Function Key bar in the Main window (default). |
| ☐ | Does not display the Function Key bar in the Main window. |

**(4) [Show window contents while resizing]**

| | |
|---|---|
| ☑ | Resizes the window while displaying areas that were not visible before. |
| ☐ | 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.

Note that clicking on the [OK] or [Apply] button while this item remains blank causes the tab setting to remain at the previously saved value.

**(6) [Select the file opened with text editor in project tree]**

| | |
|---|---|
| ☑ | When the Editor panel is activated, the file currently being opened is selected in the Project Tree panel (default). |
| ☐ | Even if the Editor panel is activated, the file currently being opened is not selected in the Project Tree panel. |

**(7) [Show overlay icons for Windows Explorer in project tree]**

| | |
|---|---|
| ☑ | Displays overlay icons for Windows Explorer in the Project Tree panel. |
| ☐ | Does not display overlay icons for Windows Explorer in the Project Tree panel (default). |

**Caution    The change of the setting is reflected after this product restarts.**

**(8) [Location of a close button for a splitted panel]**

Use this area to select the location of a close button for a splitted panel, via the following drop-down list.

| | |
|---|---|
| Right edge of the tab area | Displays to the right edge in the tab area  (default). |
| All tabs | Displays to the right edge of the title in all tabs. |
| Active tab | Displays to the right edge of the title in the active tab. |

**(9) 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.<br>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 Text Editor] category**

Use this category to configure general settings relating to the external text editor.

**Figure A-55.　Option Dialog Box ([General - External Text Editor] Category)**
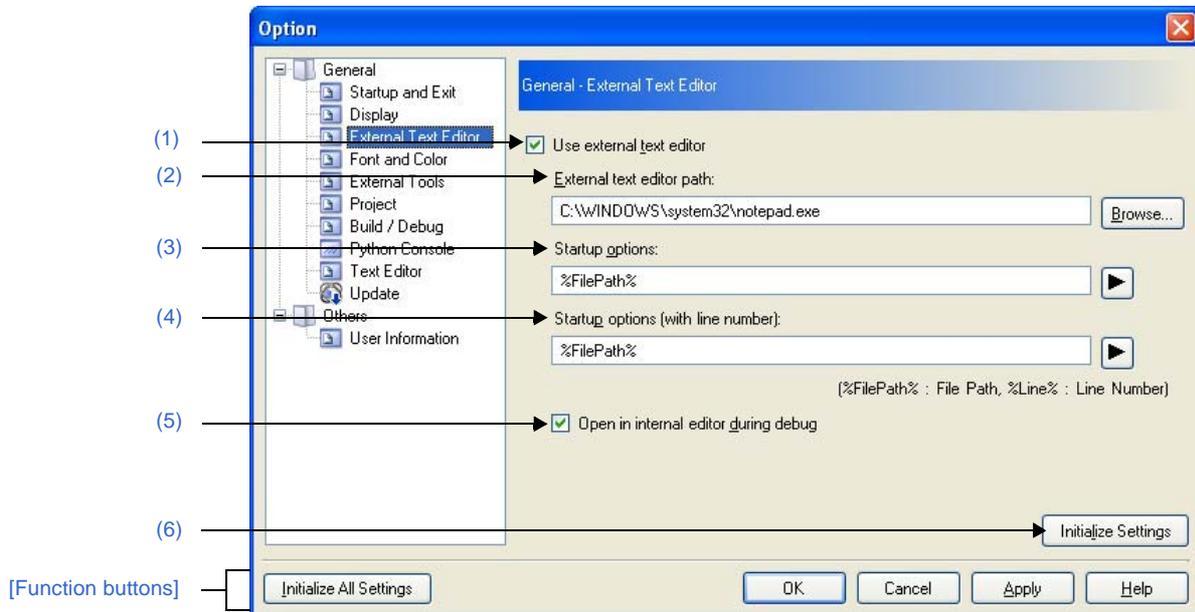


The following items are explained here.

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

**[How to open]**

- From the [Tool] menu, select [Options...].

**[Description of each area]**

**(1)　[Use external text editor]**

| | |
|---|---|
| ☑ | Uses an external text editor as the text editor. |
| ☐ | Uses the Editor panel as the text editor (default). |

**(2)　[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.

This item is only enabled if the [Use external text editor] check box is selected.

**(3) [Startup options]**

Specify the startup options (without the line number) for the external text editor (up to 256 characters).

This item is only enabled if the [Use external text editor] check box is selected.

The following placeholders can be specified by the ▶ button.

If you wish to specify more than one option, separate them by spaces.

| %FileName% | Passes the name of the file to display to the external text editor. |
|---|---|
| %FilePath% | Passes the absolute path of the file to display to the external text editor. |
| %FileRelativePath% | Passes the absolute path of the file from the project folder to display to the external text editor. |
| %Line% | Passes the caret position (line number) to the external text editor. |

**(4) [Startup options (with line number)]**

Specify the startup options (with the line number) for the external text editor (up to 256 characters).

This item is only enabled if the [Use external text editor] check box is selected.

The following placeholders can be specified by the ▶ button.

If you wish to specify more than one option, separate them by spaces.

| %FileName% | Passes the name of the file to display to the external text editor. |
|---|---|
| %FilePath% | Passes the absolute path of the file to display to the external text editor. |
| %FileRelativePath% | Passes the absolute path of the file from the project folder to display to the external text editor. |
| %Line% | Passes the caret position (line number) to the external text editor. |

**(5) [Open in internal editor during debug]**

This item is only enabled if the [Use external text editor] check box is selected.

| ☑ | Suppresses starting an external text editor and opens the file to display in the Editor panel during debugging (default). |
|---|---|
| ☐ | Opens the file to display in an external text editorl. |

**(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.<br>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-56.   Option Dialog Box ([General - Font and Color] Category)**
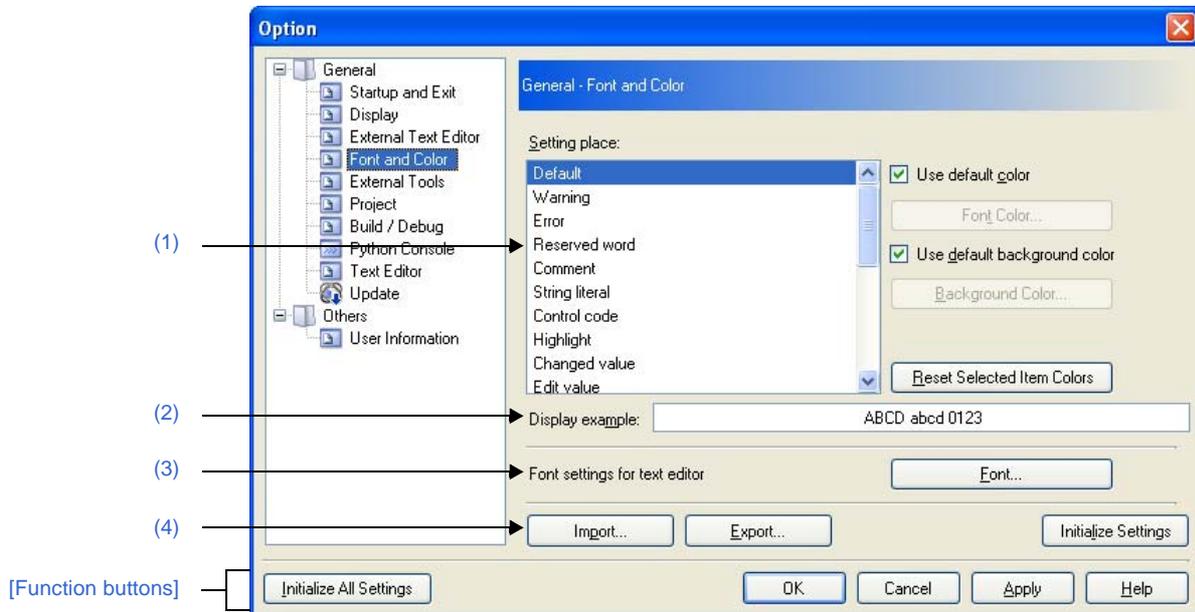


The following items are explained here.

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

**[How to open]**

- From the [Tool] menu, select [Options...].

**[Description of each area]**

**(1)  Color options area**

Use this area to configure the colors.

**(a)  [Setting place] area**

Select a location from the list for which the color will be specified.
The relationships between the list items and default color settings are as follows.

| Item | Example | | | Description |
|------|---------|--|--|-------------|
| Default[Note] | AaBbCc | Font color | Black | 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 | |

| Item | Example | | | Description |
|------|---------|------|------|-------------|
| 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, "/* ... */") 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 Output 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] [IECUBE2] [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] [IECUBE2] [Simulator]. |
| | | Background color | Orange | |
| Read and write | AaBbCc | Font color | Default color | Locations that have been read and written are displayed in this color in the Trace panel [IECUBE] [IECUBE2] [Simulator]. |
| | | Background color | LightSkyBlue | |
| Lost | AaBbCc | Font color | White | Locations that the value have been got from the debug tool is not correct in the Memory panel. |
| | | Background color | LightGray | |

| Item | Example | | | Description |
|---|---|---|---|---|
| Coverage 100% | AaBbCc | Font color | Default color | The line for which code coverage is at 100 % is displayed in this color in the Editor panel and Disassemble panel. |
| | | Background color | LightGreen | |
| Coverage 1 - 99% | AaBbCc | Font color | Default color | The line for which code coverage is at 1 to 99 % is displayed in this color in the Editor panel and Disassemble panel. |
| | | Background color | LightPink | |
| Coverage 0% | AaBbCc | Font color | Default color | The line for which code coverage is at 0 % (not yet executed) is displayed in this color in the Editor panel and Disassemble panel. |
| | | Background color | LightGray | |
| Invalid | AaBbCc | Font color | Gray | Non memory-mapped areas in the Memory panel and filenames in the Project Tree panel that do not actually exist are displayed in this color. |
| | | 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]**

| ☑ | Displays items selected via the [Setting place] area using the standard text color. |
|---|---|
| ☐ | 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]**

| ☑ | Displays items selected via the [Setting place] area using the standard background color. |
|---|---|
| ☐ | 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-57.   Edit Colors Dialog Box**



**(2)  [Display example] area**

Display sample text using the color and font settings from the Color options area.

By default the string "AaBbCc" is shown, but you can type an arbitrary string directly into the text box.

**(3)  [Font settings for text editor] area**

Click the [Font...] button to open the Font Dialog Box and configure the fonts for your text editor.

**Figure A-58.   Font Dialog Box**
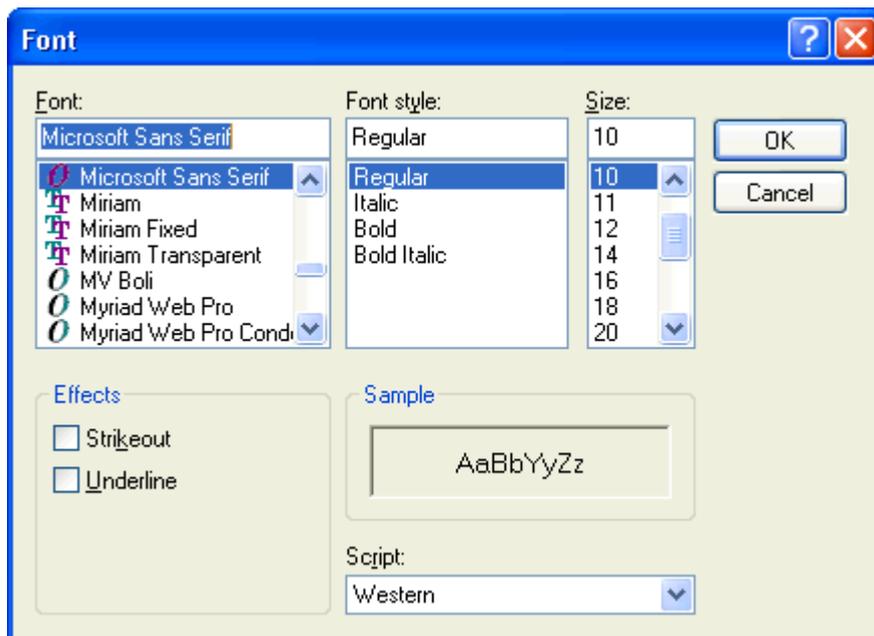
**(4) Buttons**

| | |
|---|---|
| Import... | Opens the Open Option Setting File dialog box to refer the settings saved in the file to this category. |
| Export... | Opens the Save Option Setting File dialog box to save the settings of this category to a file. |
| 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.<br>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-59.   Option Dialog Box ([General - External Tools] Category)**
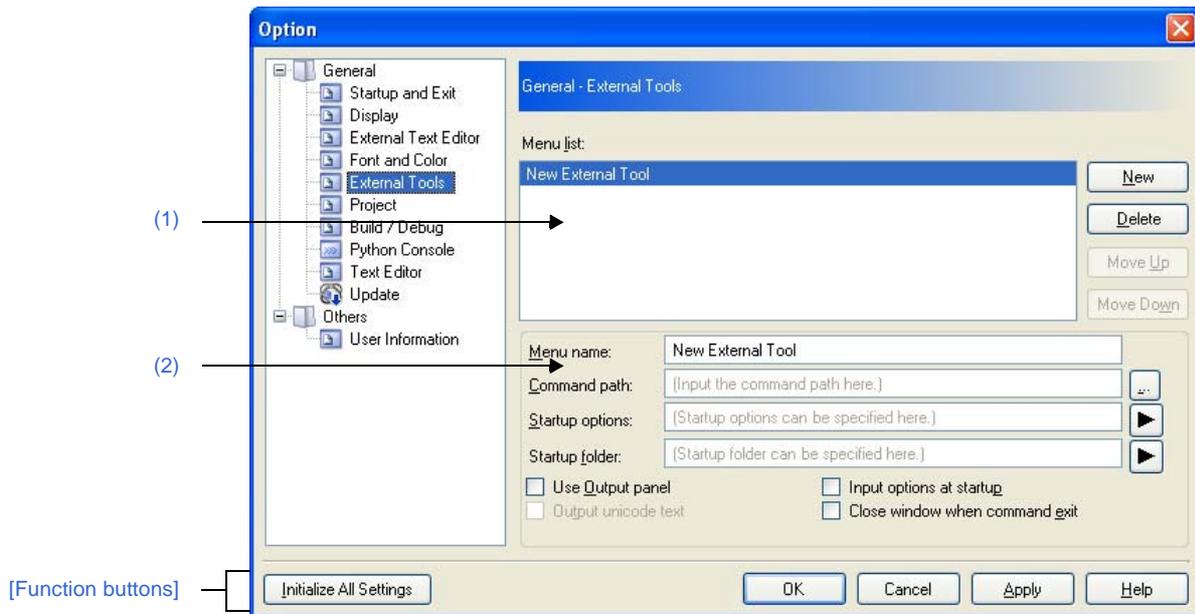


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

## [How to open]

- From the [Tool] menu, select [Options...].

## [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.<br>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 256 characters).

The following placeholders can be specified by the ▶ button.

If you wish to specify more than one option, separate them by spaces.

| %FileName% | The name of the file currently selected in the Project Tree panel is passed to the external tool. |
|---|---|
| %FilePath% | The absolute path of the file currently selected in the Project Tree panel is passed to the external tool. |
| %FileRelativePath% | The absolute path from the project folder of the file currently selected in the Project Tree panel is passed to the external tool. |
| %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). |
| %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). |

**(d) [Startup folder]**

Specify the absolute path to the folder for launching the external tool (up to 247 characters).

The following placeholders can be specified by the ▶ button.

If you wish to specify more than one folder, separate them by spaces.

| %ActiveProjectDir% | The absolute path of the active project folder of the item currently selected in the Project Tree panel is passed to the external tool (if the active project does not exist, then nothing is passed). |
|---|---|
| %FileDir% | The absolute path of the folder where the file currently selected in the Project Tree panel exists is passed to the external tool. |
| %MainProjectDir% | The absolute path of the main project folder of the item currently selected in the Project Tree panel is passed to the external tool (if the main project does not exist, then nothing is passed). |
| %MicomToolPath% | The absolute path of the install folder of this product is passed to the external tool. |
| %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 absolute path of the project folder of the item currently selected in the Project Tree panel is passed to the external tool. |
| %TempDir% | The absolute path of the temporary folder is passed to the external tool. |
| %WinDir% | The absolute path of the Windows system folder is passed to the external tool. |

**(e) [Use Output panel]**

| ☑ | Displays messages that the new external tool outputs to stdout and stderr in the Output panel.<br>Messages will be output to the Output panel's [*Tool Name*] tab ("*Tool Name*" is the string specified in [Menu name]). |
|---|---|
| ☐ | Uses the Windows command prompt for messages that the new external tool outputs to stdout and stderr (default). |

**(f) [Input options at startup]**

| ☑ | The Character String Input opens, enabling you to specify startup parameters for the new external tool. |
|---|---|
| ☐ | 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.

| ☑ | Displays messages that the new external tool outputs to stdout and stderr in the Output panel using the Unicode character set. |
|---|---|
| ☐ | 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.

| ☑ | Closes the Windows command prompt when the new external tool exits. |
|---|---|
| ☐ | 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.<br>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 - Project] category**

Use this category to configure general settings relating to the project.

**Figure A-60.   Option Dialog Box ([General - Project] Category)**
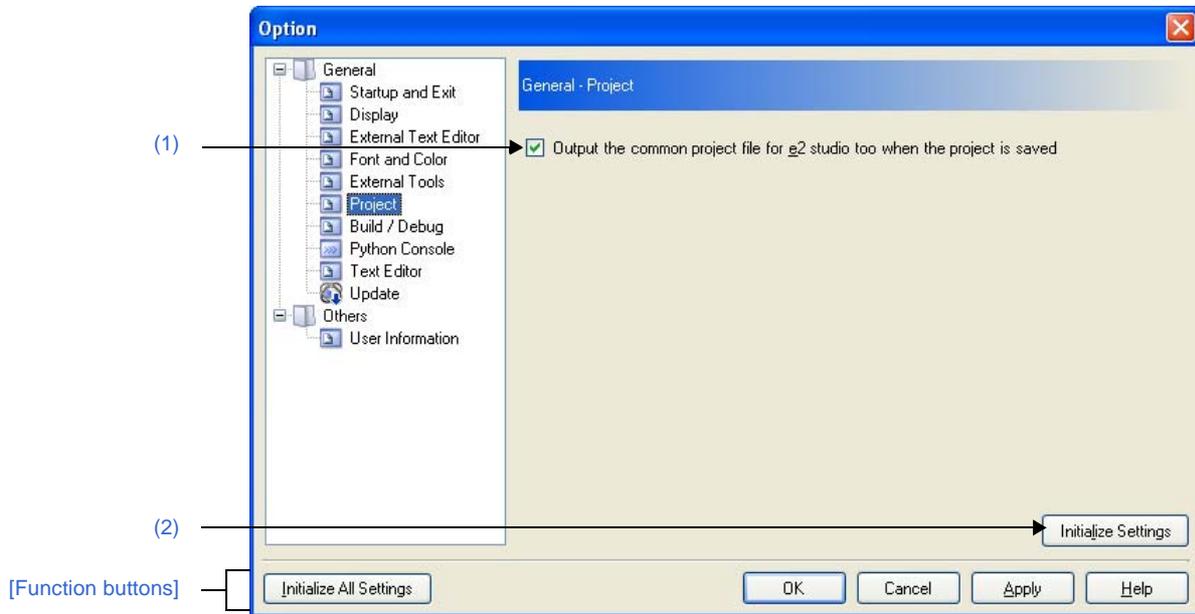


The following items are explained here.

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

**[How to open]**

   - From the [Tool] menu, select [Options...].

**[Description of each area]**

**(1) [Output the common project file for e2 studio too when the project is saved]**

| | |
|---|---|
| ☑ | Outputs the common project file for e$^2$ studio too when the project is saved (default). |
| ☐ | Does not output the common project file for e$^2$ studio when the project is saved. |

**(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. |

## [General - Build/Debug] category

Use this category to configure general setting relating to building and debugging.

**Figure A-61.   Option Dialog Box ([General - Build/Debug] Category)**
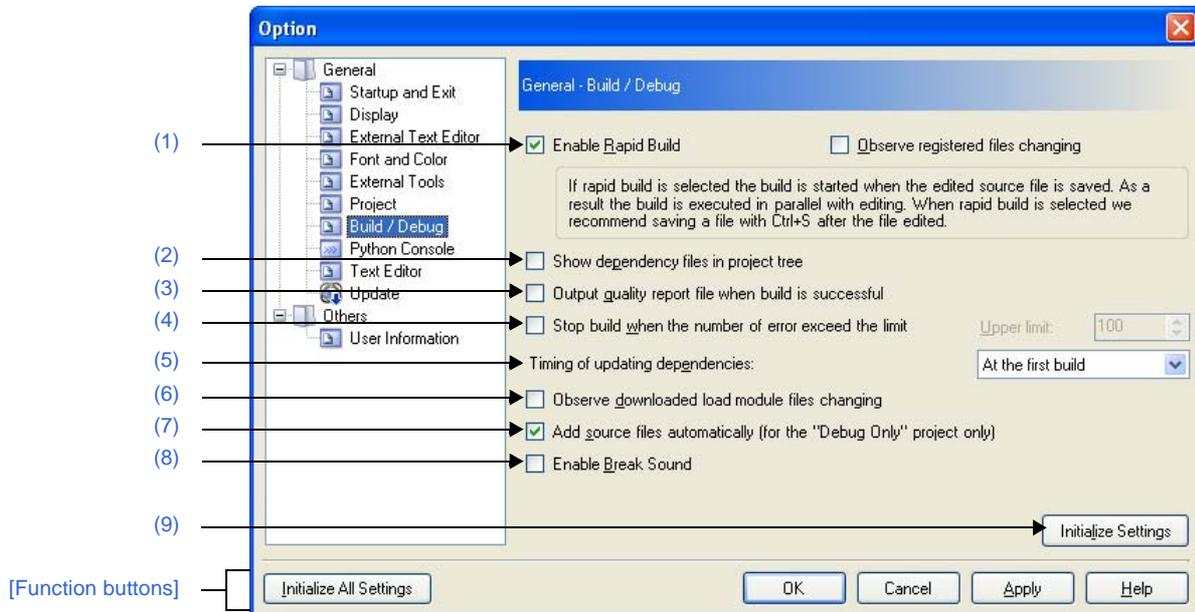


The following items are explained here.

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

### [How to open]

- From the [Tool] menu, select [Options...].

### [Description of each area]

**(1)  [Enable Rapid Build]**

| | |
|---|---|
| ☑ | Enables the rapid build Note feature (default). |
| ☐ | 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+ Integrated Development Environment User's Manual: Build"
- When the target project type is a debug-dedicated project: "F.5.4   Run a rapid build"

**(a) [Observe registered files changing]**

| | |
|---|---|
| ☑ | 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. |
| ☐ | 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.

**Cautions 1.    The rapid build will not finish if this item is selected, and the files to be built have been registered for automatic editing or overwriting (e.g. by commands executed before or after the build).**
**If the rapid build does not finish, unselect this item, and stop the rapid build.**
**2.    If this item is selected, a file that is registered in the project but does not exist (a file grayed out) will not be observed even if it is registered again by the Explorer etc.**
**To observe the file, reload the project file, or select this item again after unselecting this item and closing this dialog box.**

**(2) [Show dependency files in project tree]**

| | |
|---|---|
| ☑ | Displays the file group on which the source file depends on the project tree. |
| ☐ | Does not display the file group on which the source file depends on the project tree (default). |

**(3) [Output quality report file when build is successful]**

| | |
|---|---|
| ☑ | Output the quality report file when a build is successful. |
| ☐ | Does not output the quality report file when a build is successful (default). |

Remarks 1.    The quality report file is not output when a rapid build is executed, a debug-dedicated project is built, and compiling or assembling is executed in file units.
2.    The following information item is output to the quality report file.
   - Time and date on which the file is created
   - Log of the build results
   - Information on the command file which is used during building
   - Information on the detailed version of this product or the current project
3.    The quality report file is output with the file name "QualityReport(*project-name.build-mode-name*).text" to the project folder of each project.
If a file having the same name exists, it will be overwritten.
It is also shown on the project tree, under the Build tool generated files node.

**(4) [Stop build when the number of error exceed the limit]**

| | |
|---|---|
| ☑ | Stops the build if the total number of errors at the build reaches the number specified in [Upper limit]. |
| ☐ | Does not stop the build even if the total number of errors at the build reaches the number specified in [Upper limit] (default). |

**(a)  [Upper limit]**

Specify the upper limit of the number of errors.

Either enter a number between 1 and 10000 directly via the keyboard, or specify a number via the 🔼 buttons. The default is 100.

Note that clicking on the [OK] or [Apply] button while this item remains blank causes the tab setting to remain at the previously saved value.

**Remark**  This item is only enabled if the [Stop build when the number of error exceed the limit] check box is selected.

**(5)  [Timing of updating dependencies]**

Select the timing for automatically updating dependencies from the drop-down list of either of the following:

| At the first build | Updates dependencies immediately before executing the first build after opening the project (default). |
|---|---|
| At every build | Updates dependencies immediately before executing the build. |

**(6)  [Observe downloaded load module files changing]**

| ☑ | Monitors load module files downloaded to the debugging tool for changes.  When there is a change, a message dialog box confirming whether to execute the download is displayed. |
|---|---|
| ☐ | Does not monitor load module files downloaded to the debugging tool for changes (default). |

**(7)  [Add source files automatically (for the "Debug Only" project only)]**

| ☑ | In a debug-dedicated project, when downloading a load module file into the debug tool, source files are automatically added to the project tree (default). |
|---|---|
| ☐ | In a debug-dedicated project, when downloading a load module file into the debug tool, source files are not automatically added to the project tree. |

**Caution  This function is only valid when a load module file has been added to the  Download files node of the project tree.**

**When a load module file has been added in the [Download File Settings] tab on the Property panel of the debug tool, source files are not added to the project tree.**

**(8)  [Enable Break Sound]**

| ☑ | Beeps when the execution of a user program is halted due to a break event (hardware or software break). |
|---|---|
| ☐ | Does not beep when the execution of a user program is halted due to a break event (hardware or software break) (default). |

**(9)  Buttons**

| Initialize Settings | Returns all currently displayed setting to their default values. |
|---|---|

**[Function buttons]**

| Button | Function |
|---|---|
| Initialize All Settings | Restores all settings on this dialog box to their default values.<br><br>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 - Python Console] category

Use this category to configure general setting relating to the Python console.

**Figure A-62.   Option Dialog Box ([General - Python Console] Category)**
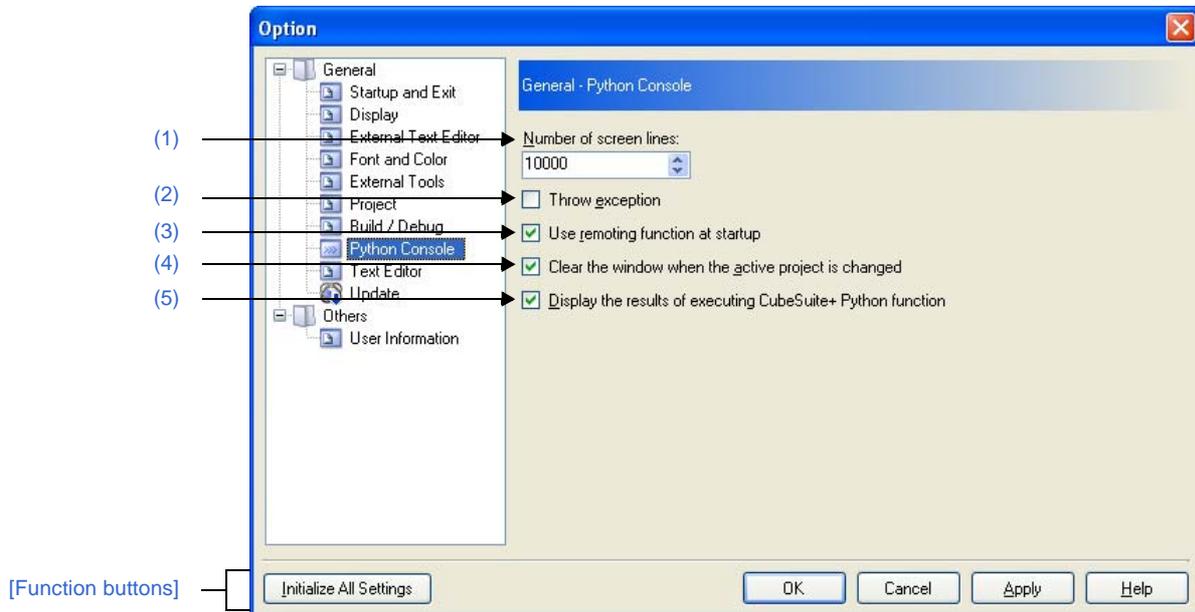


The following items are explained here.

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

### [How to open]

- From the [Tool] menu, select [Options...].

### [Description of each area]

**(1)  [Number of screen lines]**

Specify the number of screen lines for the Python console.

Either enter a number between 5000 and 100000 directly via the keyboard, or specify a number via the ⬍ buttons.

The default is 10000.

**Remark**     This item can be set or referred by using CubeSuite+ Python property "common.ViewLine".

**(2)  [Throw exception]**

| | |
|---|---|
| ☑ | Throws an exception. |
| ☐ | Does not throws an exception (default). |

**Remark**     This item can be set or referred by using CubeSuite+ Python property "common.ThrowExcept".

**(3) [Use remoting function at startup]**

| | |
|---|---|
| ☑ | Enables the function for linking to external tools when starting the Python console (default). |
| ☐ | Disables the function for linking to external tools when starting the Python console. |

**Remark**    This item can be set or referred by using CubeSuite+ Python property "common.UseRemoting".

**(4) [Clear the window when the active project is changed]**

| | |
|---|---|
| ☑ | Clears the window when the active project is changed (default). |
| ☐ | Does not clear the window when the active project is changed. |

**Remark**    This item can be set or referred by using CubeSuite+ Python property "common.ConsoleClear".

**(5) [Display the results of executing CubeSuite+ Python function]**

| | |
|---|---|
| ☑ | Displays the results of executing a CubeSuite+ Python function in the Python console (default). |
| ☐ | Does not display the results of executing a CubeSuite+ Python function in the Python console. |

**Remark**    This item can be set or referred by using CubeSuite+ Python property "common.ViewOutput".

**[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-63.   Option Dialog Box ([General - Text Editor] Category)**
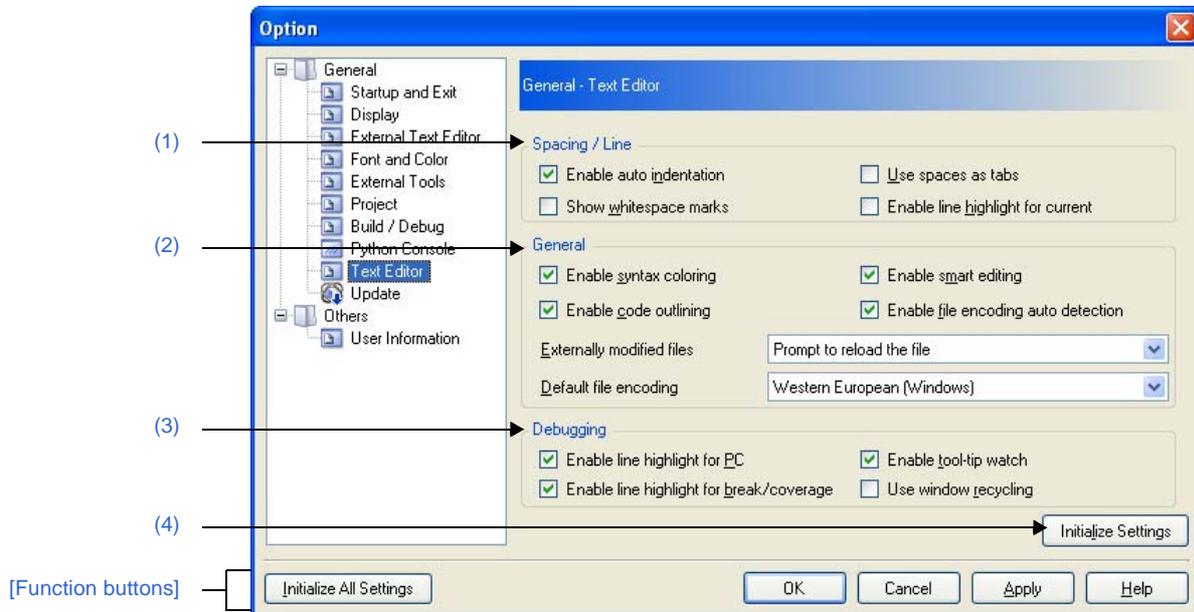


The following items are explained here.

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

**[How to open]**

- From the [Tool] menu, select [Options...].

**[Description of each area]**

**(1)  [Spacing]**

Use this area to configure the spacing-related settings in the text editor.

**(a)  [Enable auto indentation]**

| | |
|---|---|
| ☑ | Code is intelligently indented depending on the code being typed in the text editor. E.g. Inside a function or not or coding a loop (default). |
| ☐ | Code is not indented depending on the code being typed in the text editor. |

**(b)  [Use spaces as tabs]**

| | |
|---|---|
| ☑ | When you press the [Tab] key in the text editor, a group of space characters is inserted equal to the width of a tab. |
| ☐ | When you press the [Tab] key in the text editor, the tab character is inserted (default). |

**(c) [Show whitespace marks]**

| | |
|---|---|
| ☑ | Whitespace marks such as space and tab are shown in the text editor. |
| ☐ | Whitespace marks such as space and tab are not shown in the text editor (default). |

**(d) [Enable line highlight for current]**

| | |
|---|---|
| ☑ | Highlights the current line in the text editor. |
| ☐ | Does not highlight the current line in the text editor (default). |

**(2) [General]**

Use this area to configure general settings in the text editor.

**(a) [Enable syntax coloring]**

| | |
|---|---|
| ☑ | Enables the coloring of keywords in the text editor that have supported file-types (default). |
| ☐ | No coloring of keywords is displayed in the text editor. |

**(b) [Emable smart editing]**

| | |
|---|---|
| ☑ | Enables the smart edit function (default). |
| ☐ | Disables the smart edit function. |

**Remark**   This function is disabled when the build tool is CA850, CA780R, or CA78K0.

**(c) [Enable code outlining]**

This item is only available if the [Enable syntax coloring] check box is selected.

| | |
|---|---|
| ☑ | Enables the code outlining feature in the text editor.<br>This enables the expansion and collapse of code blocks for file-types that support it (default). |
| ☐ | Does not use the code outlining feature in the text editor. |

**(d) [Enable file encoding auto detection]**

| | |
|---|---|
| ☑ | Enables automatic determination of the encoding when a file is read (default). |
| ☐ | Disables automatic determination of the encoding when a file is read. |

**(e) [Externally modified files]**

This item allows you to specify the action to be taken for source files that are modified externally to the Cube-Suite+.

| Prompt to reload the file | When a file open in the text editor is modified externally to the CubeSuite+, a message dialog box will be displayed which asks you whether you wish to reload the file or not (default).<br><br>If you click the [Yes] button on the message dialog box, this will reload the file from disk, losing all current changes to that file in the text editor.<br><br>If you click the [No] button on the message dialog box, this will ignore the external changes and any existing modifications to that file in the text editor will remain. |
|---|---|
| Always reload the file | This will automatically reload the file when it is modified externally. |
| Never reload the file | This will ignore any externally-modified changes to source files and will not reload the file. |

**(f)  [Default file encoding]**

This option is for the default file encoding that should be used when creating a new file in the Editor panel and automatic determination of encoding is disabled.  In addition this encoding is used when the file being opened has a file encoding that cannot be detected or is not supported in the Editor panel currently.

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.

- *Default encoding of the current OS* (default)
- *Most recently used encodings (maximum 4)*
- *Popular encodings for current locale*
    e.g. for United States locale it will be:
        - Western European (Windows)
        - Unicode (UTF-8)
- All other encodings supported by the OS (in alphabetical order)

**(3)  [Debugging]**

Use this area to configure general settings relating to debugging in the text editor.

**(a)  [Enable line highlight for PC]**

| ☑ | Shows the current PC as a full line highlight in the text editor (default). |
|---|---|
| ☐ | Does not show the current PC as a full line highlight in the text editor. |

**(b)  [Enable tool-tip watch]**

| ☑ | Shows the value of variables in a tool-tip when hovering the mouse over the variable in the text editor (default). |
|---|---|
| ☐ | Does not show the value of variables in a tool-tip. |

**(c)  [Enable line highlight for break/coverage]**

| ☑ | Shows the breakand coverage as a full line highlight in the text editor (default). |
|---|---|
| ☐ | Does not show the break and coverage as a full line highlight in the text editor. |

**(d) [Use window recycling]**

| | |
|---|---|
| ☑ | Uses window recycling. |
| ☐ | Does not use window recycling (default). |

> **Remark**    See "CubeSuite+ Integrated Development Environment User's Manual: Debug" for details about
> window recycling.

**(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-64.   Option Dialog Box ([General - Update] Category)**
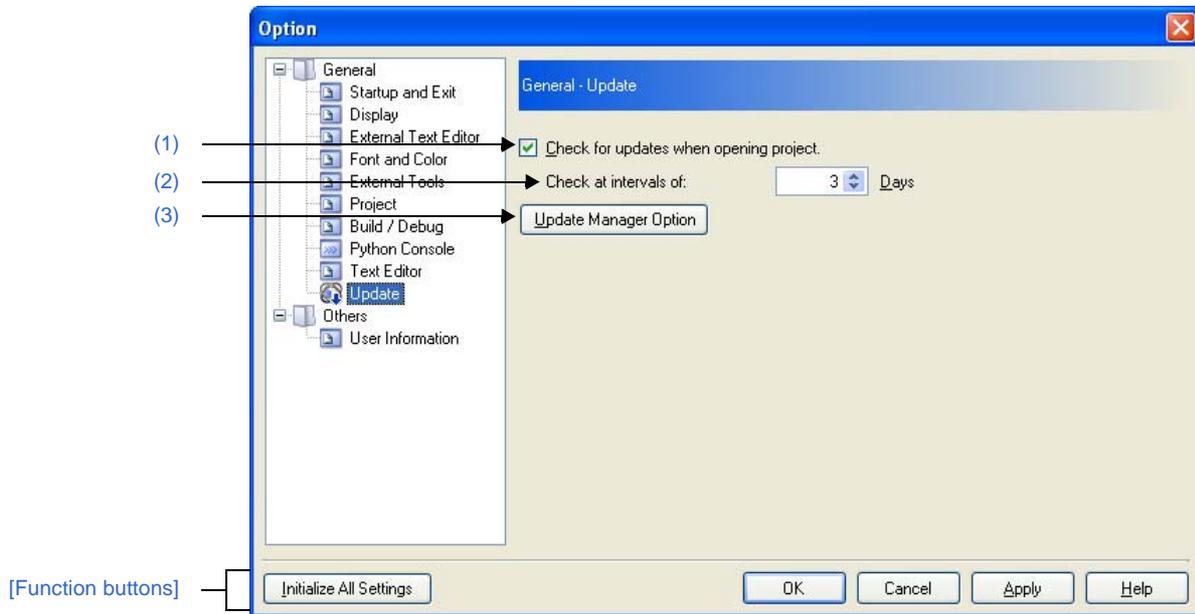


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

## [How to open]

- From the [Tool] menu, select [Options...].

## [Description of each area]

**(1)  [Check for updates when opening project.]**

| | |
|---|---|
| ☑ | Automatically checks for updates when a project is opened. |
| ☐ | 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.<br>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-65.   Option Dialog Box ([Other - User Information] Category)**
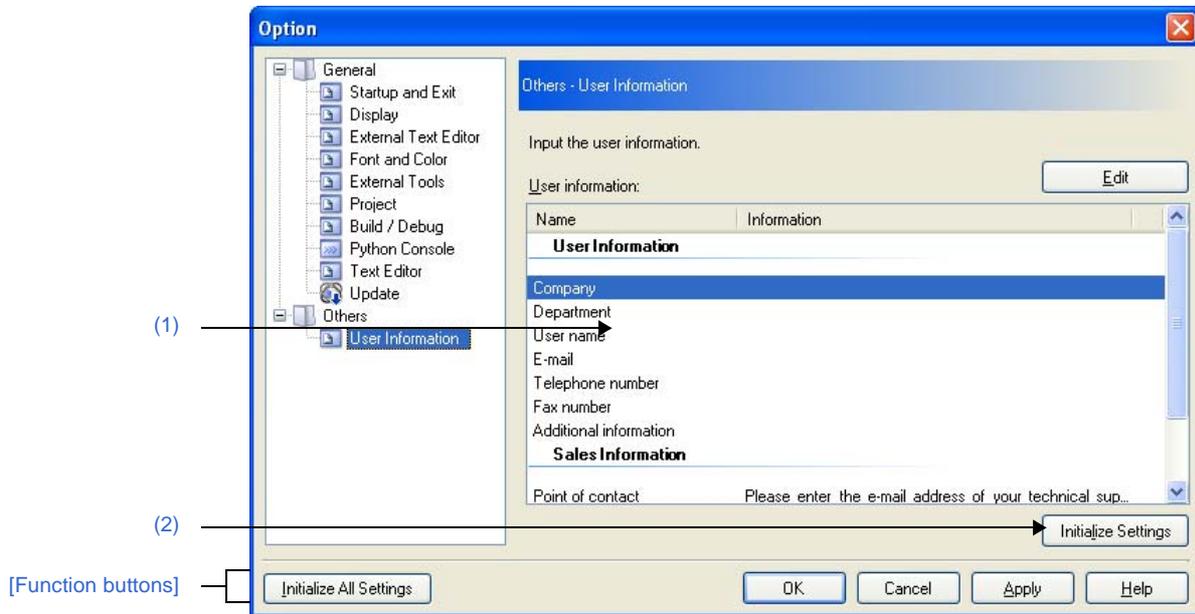


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[How to open]**

- From the [Tool] menu, select [Options...].

**[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. |

---

**Plug-in Manager dialog box**

This dialog box is used to set the plug-ins to be read in when this product is started.

**Caution    The settings in this dialog box are effective from the next time this product is started.**

**Figure A-66.   Plug-in Manager Dialog Box**
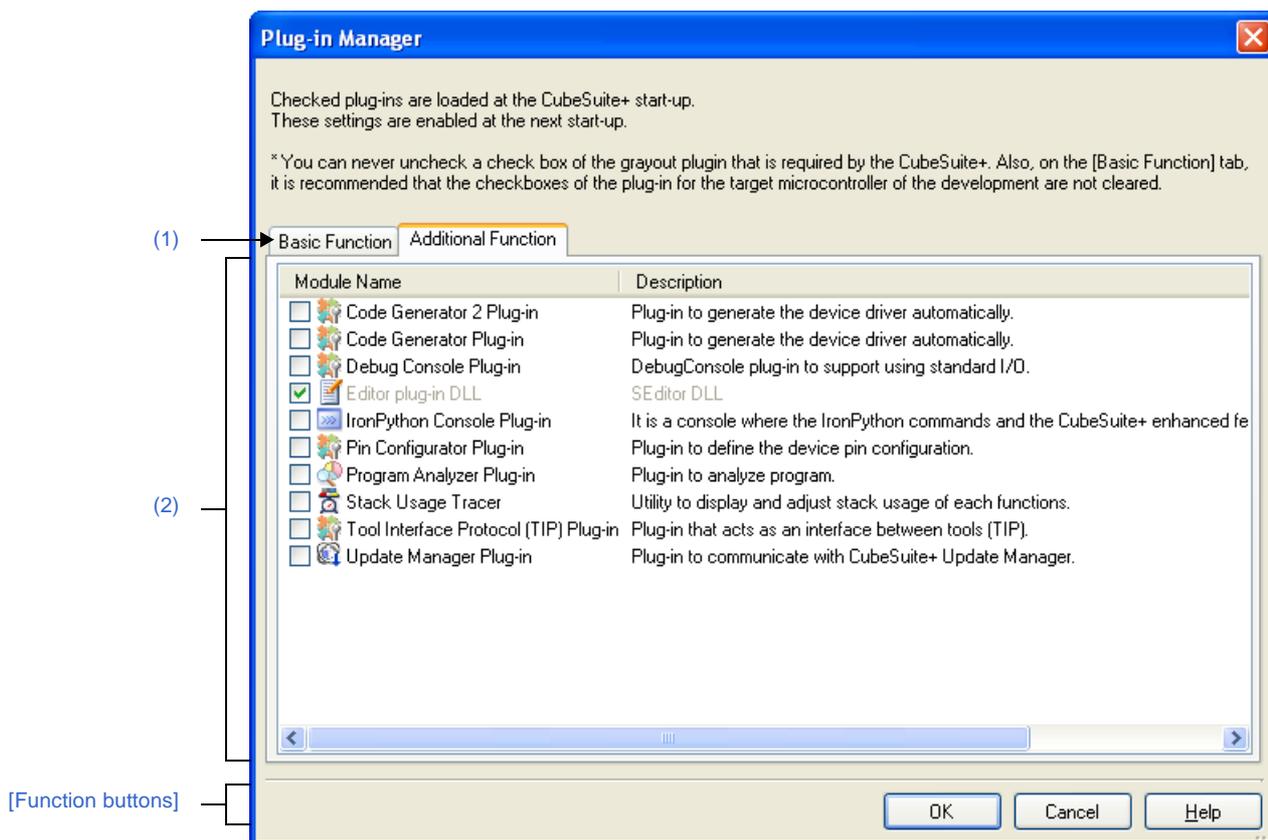
The following items are explained here.

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

**[How to open]**

- From the [Tool] menu, select [Plug-in Setting...].

**[Description of each area]**

**(1)  Tab selection area**

By selecting a tab, the plug-ins displayed in the plug-in list area are switched.

This dialog box has the following tabs:

- [Basic Function] tab
- [Additional Function] tab

---

**(2) Plug-in list area**

Select the plug-ins read when this product is started with check boxes.

For details of displayed items/setting method, see the description of each tab.

**[Function buttons]**

| Button | Function |
|--------|----------|
| OK | Applies all setting and closes this dialog box. |
| Cancel | Cancels the designated condition and closes the dialog box. |
| Help | Displays the help of this dialog box. |

**[Basic Function] tab**

This tab is used to set the build tool and debug tool plug-ins to be read in when this product is started.

**Figure A-67.   Plug-in Manager Dialog Box: [Basic Function] Tab**
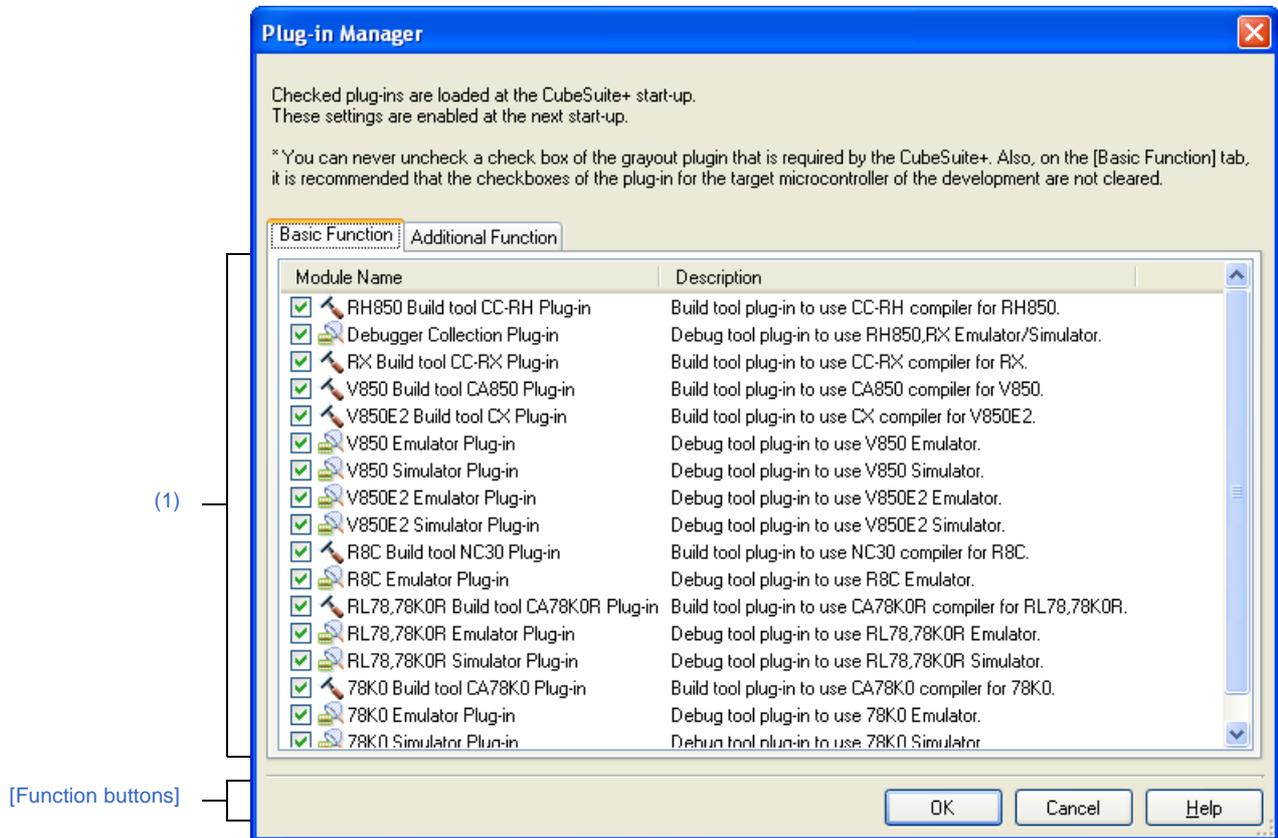


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[How to open]**

- From the [Tool] menu, select [Plug-in Setting...].

**[Description of each area]**

**(1)  Plug-in list area**
This area shows a list of the build tool and debug tool plug-ins installed in this product, in the following format.

| Module Name | Shows the plug-in module names. |
|---|---|
| | The plug-ins read in when this product is started are selected with check boxes. |
| Description | Shows a simple description of the plug-in. |

**[Function buttons]**

| Button | Function |
|--------|----------|
| OK | Applies all setting and closes this dialog box. |
| Cancel | Cancels the designated condition and closes the dialog box. |
| Help | Displays the help of this dialog box. |

**[Additional Function] tab**

This tab is used to set the plug-ins other than build tool and debug tool plug-ins to be read in when this product is started (plug-ins present in the Plugins folder of this product).

> **Caution**   **The plug-ins except for the Editor panel are disabled on this tab after default installation.**
> **To use each plug-in, enable it on this tab and then restart CubeSuite+.**

**Figure A-68.   Plug-in Manager Dialog Box: [Additional Function] Tab**
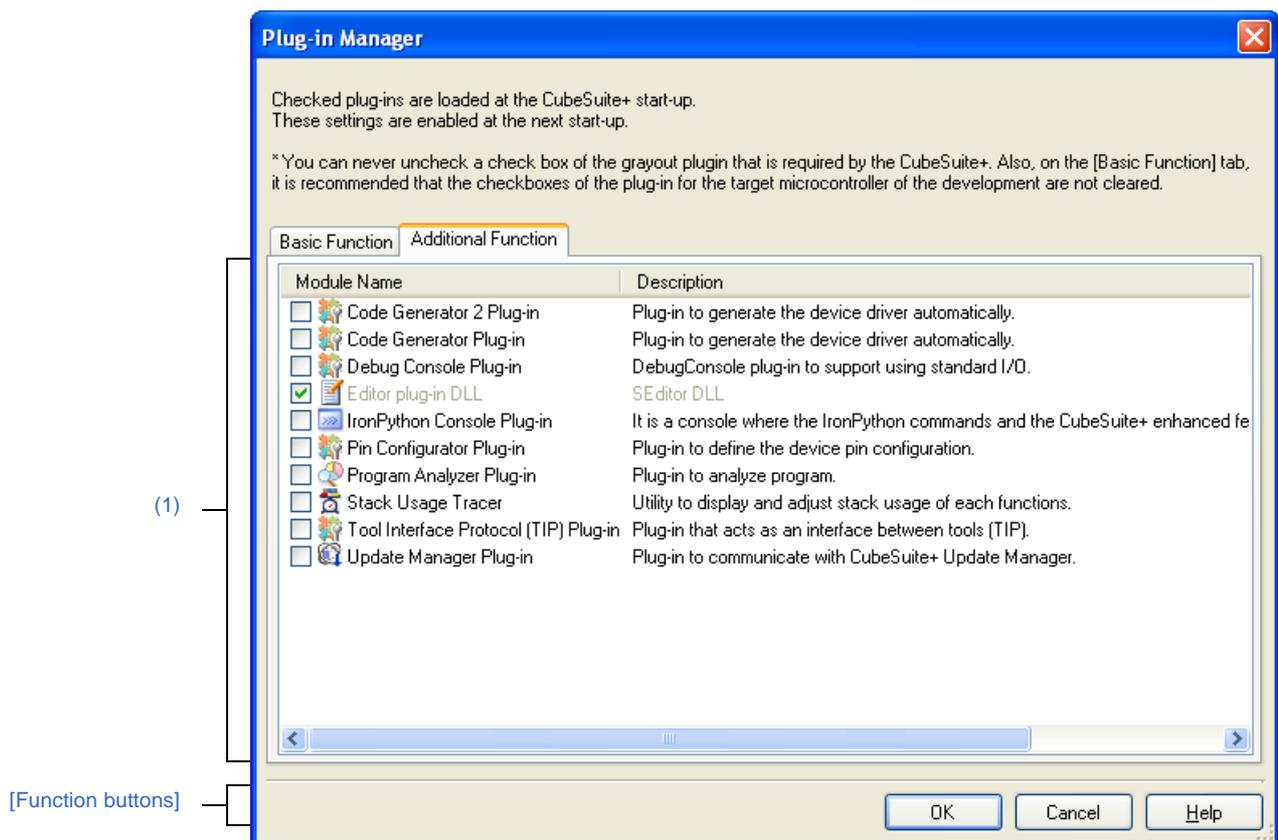


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[How to open]**

- From the [Tool] menu, select [Plug-in Setting...].

**[Description of each area]**

**(1)  Plug-in list area**
This area shows a list of the plug-ins other than build tool and debug tool plug-ins installed in this product (plug-ins present in the Plugins folder of this product), in the following format.

| Module Name | Shows the plug-in module names.<br><br>The plug-ins read in when this product is started are selected with check boxes. |
|---|---|
| Description | Shows a simple description of the plug-in. |

**Remarks 1.** The Editor panel plug-in cannot be removed from the list of those read in, and so the display is grayed out.

      **2.** When this product is started from the command line, plug-ins below the Plugins folder specified with the /np option do not appear in this area. When the /npall option is specified, plug-ins below the Plugins folder are not read in, and thus nothing appears in this area.

See "2.12   Manipulate CubeSuite+ on the Command Line" for details about the /np and /npall options.

**[Function buttons]**

| Button | Function |
|---|---|
| OK | Applies all setting and closes this dialog box. |
| Cancel | Cancels the designated condition and closes the 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-69.   User Setting Dialog Box**
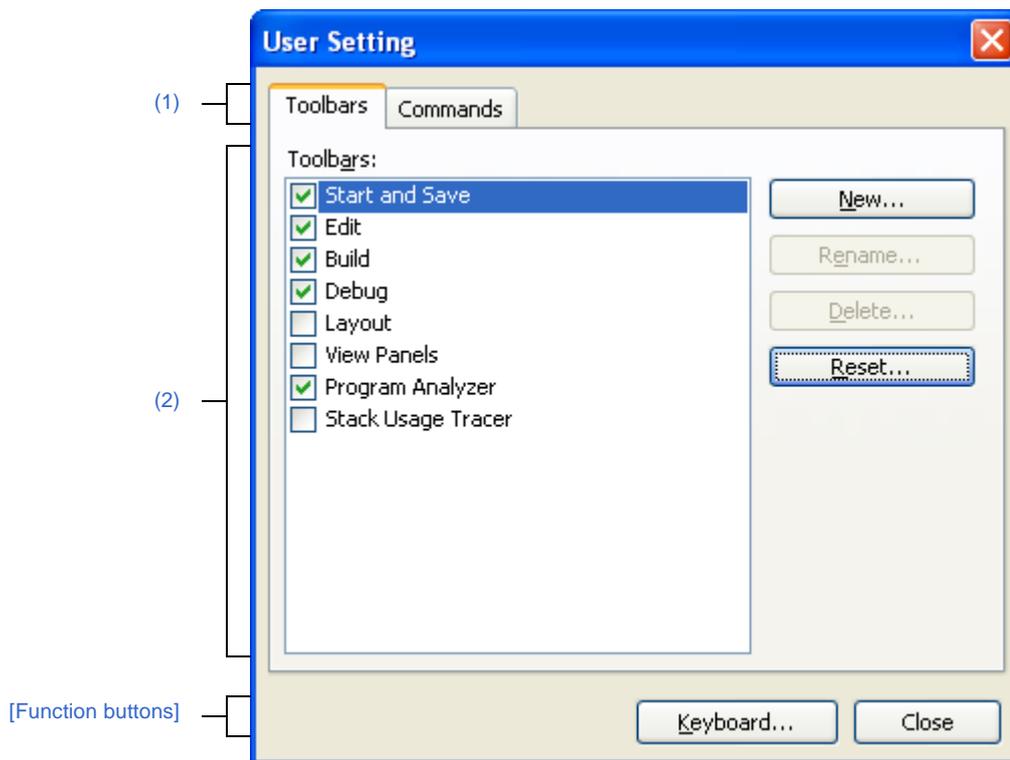


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[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-70.   User Setting Dialog Box: [Toolbars] Tab**
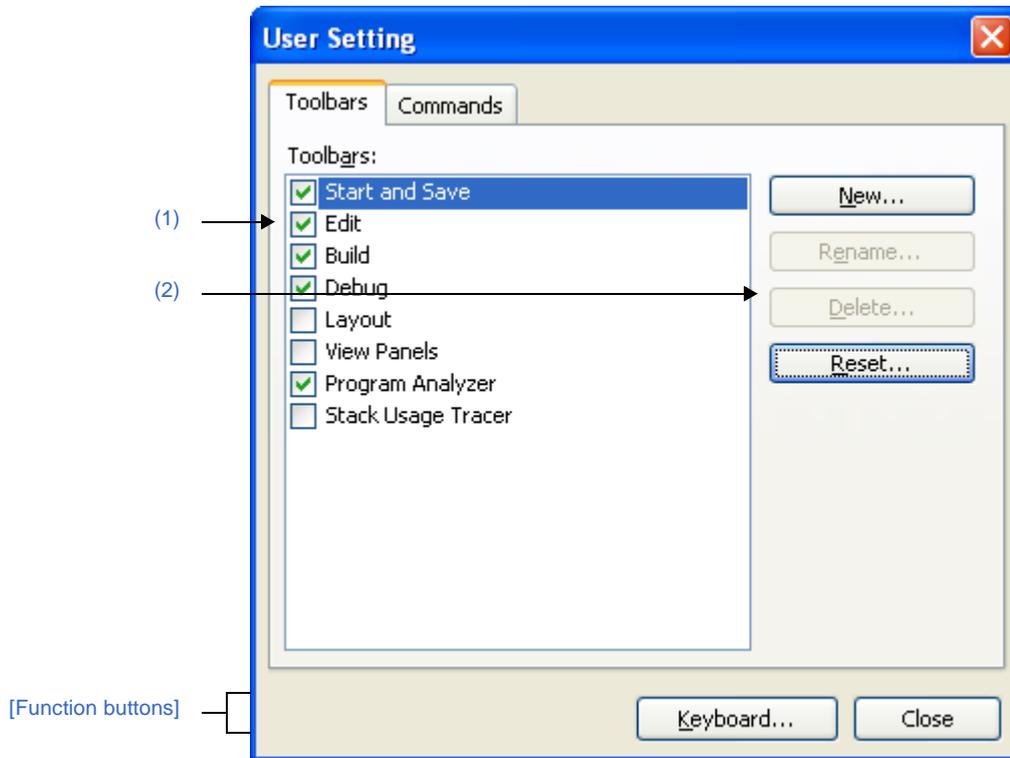


The following items are explained here.

    - [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... | Opens the New Toolbar dialog box to make a new toolbar.<br><br>The name of the new toolbar will be added to the list displayed in the [Toolbars] area with a check mark placed in front.<br><br>Note that there are no buttons on the newly created toolbar.  The register buttons on it, use the [Commands] tab. |
| Rename... | Opens the Rename Toolbar dialog box to change the name of the currently selected toolbar.<br><br>This function is disabled when a toolbar other than those created by the user is selected. |
| Delete... | Deletes the selected toolbar.<br><br>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.<br><br>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-71.   User Setting Dialog Box: [Commands] Tab**
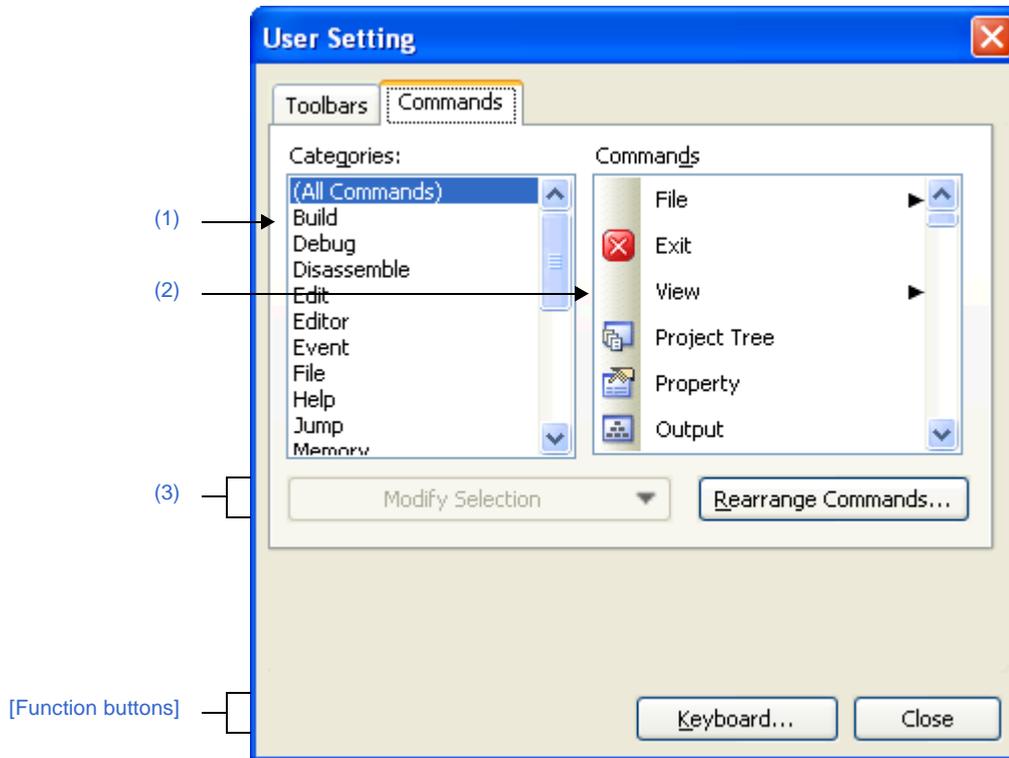


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[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 tool bar 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-72.   New Toolbar Dialog Box**
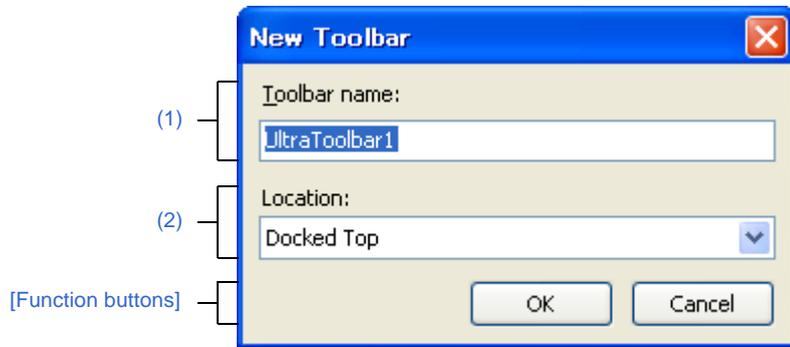


(1)

(2)

[Function buttons]

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.<br>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-73.   Rename Toolbar Dialog Box**

(1)

[Function buttons]

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-74.   Customize Keyboard Dialog Box**
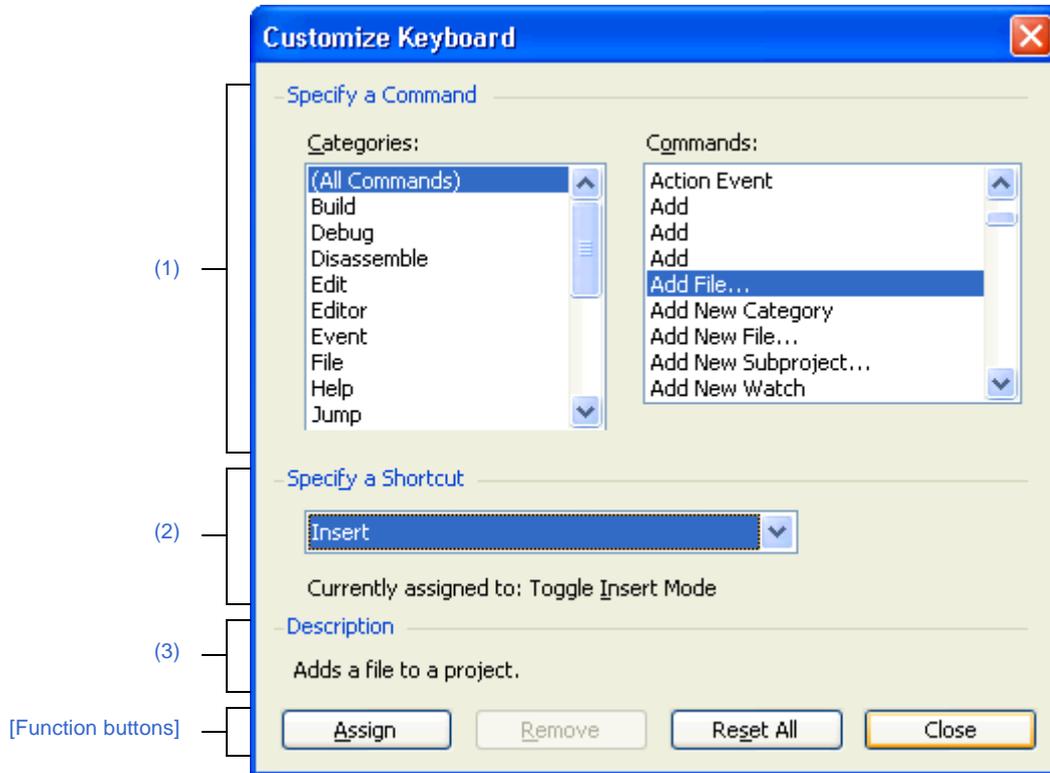


The following items are explained here.

- [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 | |

[Current assigned to] 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).

**(3) [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.<br><br>Note, however, that this button will be disabled if the key selected in the [Specify a Short-cut] 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).<br><br>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-75. Rearrange Commands Dialog Box**
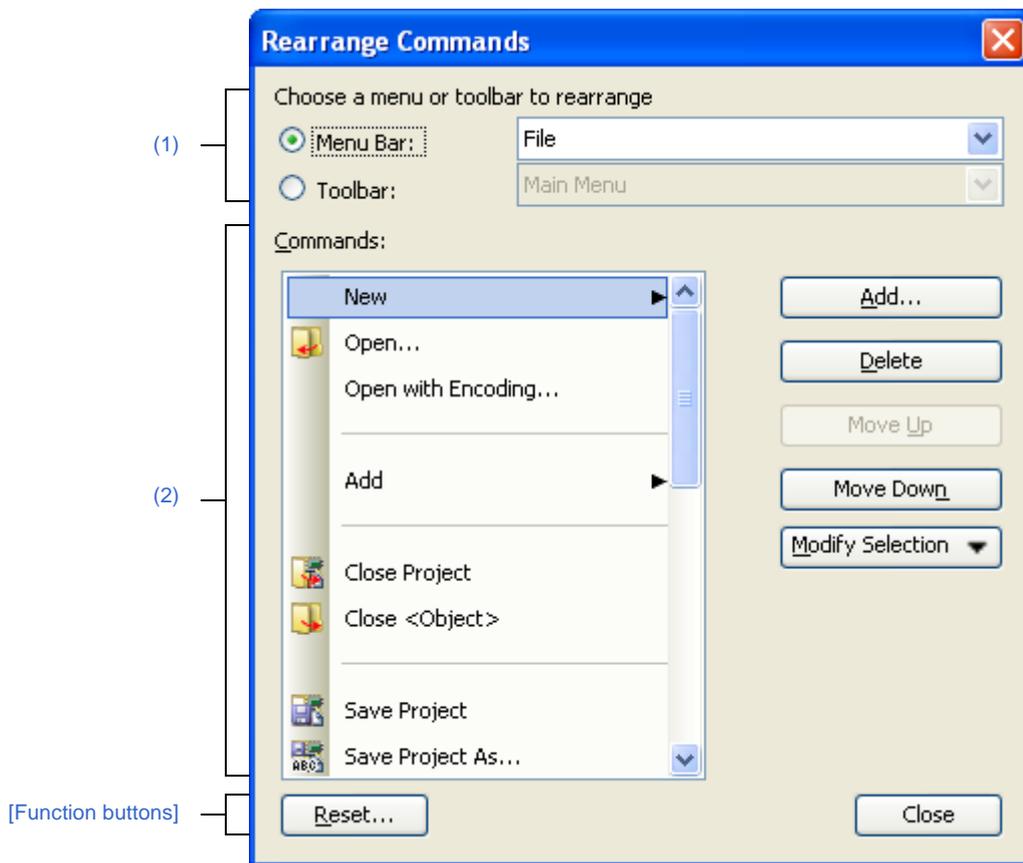


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[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-76.   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-77.   Version Information Dialog Box**
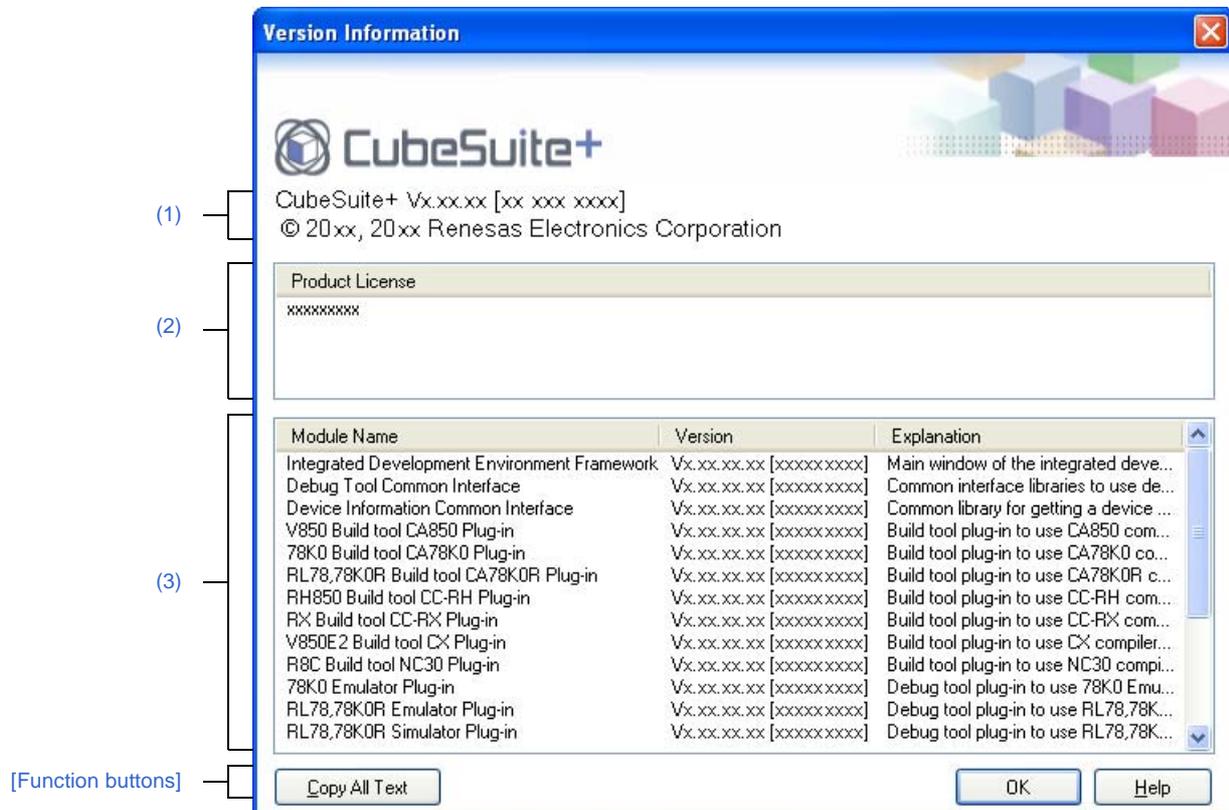


The following items are explained here.

- [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. |

**Detail Version Information dialog box**

This dialog box is used to displays detail version information of this product and the present project information.

**Figure A-78.   Detail Version Information Dialog Box**



(1)
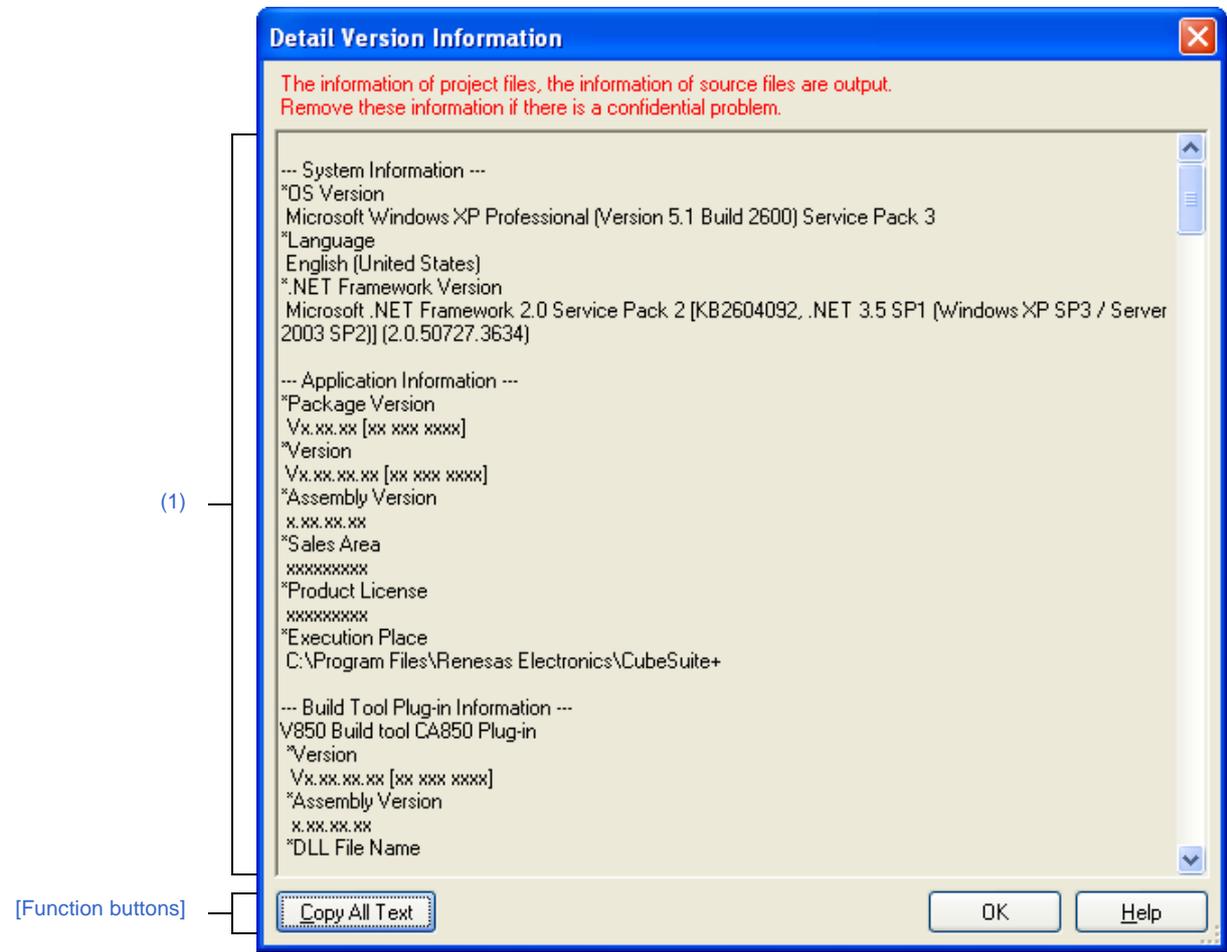
[Function buttons]

The following items are explained here.
- [How to open]
- [Description of each area]
- [[Edit] menu (Detail Version Information dialog box-dedicated items)]
- [Context menu]
- [Function buttons]

**[How to open]**

- From the [Help] menu, select [Detail Verison Information...].
- Click the [Detail Verison Information...] button in the Message dialog box.

**[Description of each area]**

**(1) Information text area**

Show the detail version information of this product and the present project information.

The necessary information is as follows.

| Information | | Contents |
|---|---|---|
| System Information | OS Version | Windows information |
| | Language | |
| | .NET Framework Version | |
| Application Information | Package Version | Information of this product |
| | Version | |
| | Assembly Version | |
| | Sales Area[Note 1] | |
| | Product License | |
| | Additional Information[Note 1] | |
| | Execution Place | |
| | Memory Usage | |
| Plug-in Information[Note 2] | Module name | Each plug-in information for this product in use |
| | Version[Note 3] | |
| | Assembly Version[Note 3] | |
| | DLL file name[Note 3] | |
| | URL[Note 1] | |
| Present Project Information[Note 4] | Main Project Information (Subproject Information) | This shows the absolute path of the project file (the absolute path including the *.mtpj file name for main projects, and the absolute path including the *.mtsp file name for subprojects). |
| | Microcontroller Information | The following item to the specified microcontroller information<br> - Device name<br> - Device file name: List of version |
| | Build Tool Information | The following out of the set build tool plug-in information[Note 5]<br> - Build tool name<br> - Version of the build tool plug-in<br> - 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)<br> - Connected debug tool name<br> - Version of the debug tool plug-in<br> - Version of the debug tool control program |
| The Error Information Which Occurred[Note 6] | | Detailed information on the occurred error information |

**Notes 1.** This is not shown when the information does not exist.

**2.** The module name of a plug-in not read in has the suffix "(-)" added to the end.

The module name of a plug-in read in but not functioning has the suffix "(!)" added to the end.

3.  For a plug-in not read in, this is shown as "-".
4.  This is only shown when projects are opened (when there is subproject, line them up).
5.  These are not shown when the project type is a debug-dedicated project.
6.  This is only shown when this dialog box is opened by the [Detail Version Information...] button in Message dialog box.

### [[Edit] menu (Detail Version Information dialog box-dedicated items)]

| | |
|---|---|
| Copy | Copies the selected characters in the detail version information area to the clip board. |
| Select All | Selects all the characters in the detail version information area. |

### [Context menu]

| | |
|---|---|
| Copy | Copies the selected characters in the detail version information area to the clip board. |
| Select All | Selects all the characters in the detail version information area. |

### [Function buttons]

| Button | Function |
|---|---|
| Copy All Text | Copies all the characters displayed in the detail version information 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-79.   One Point Advice Dialog Box**
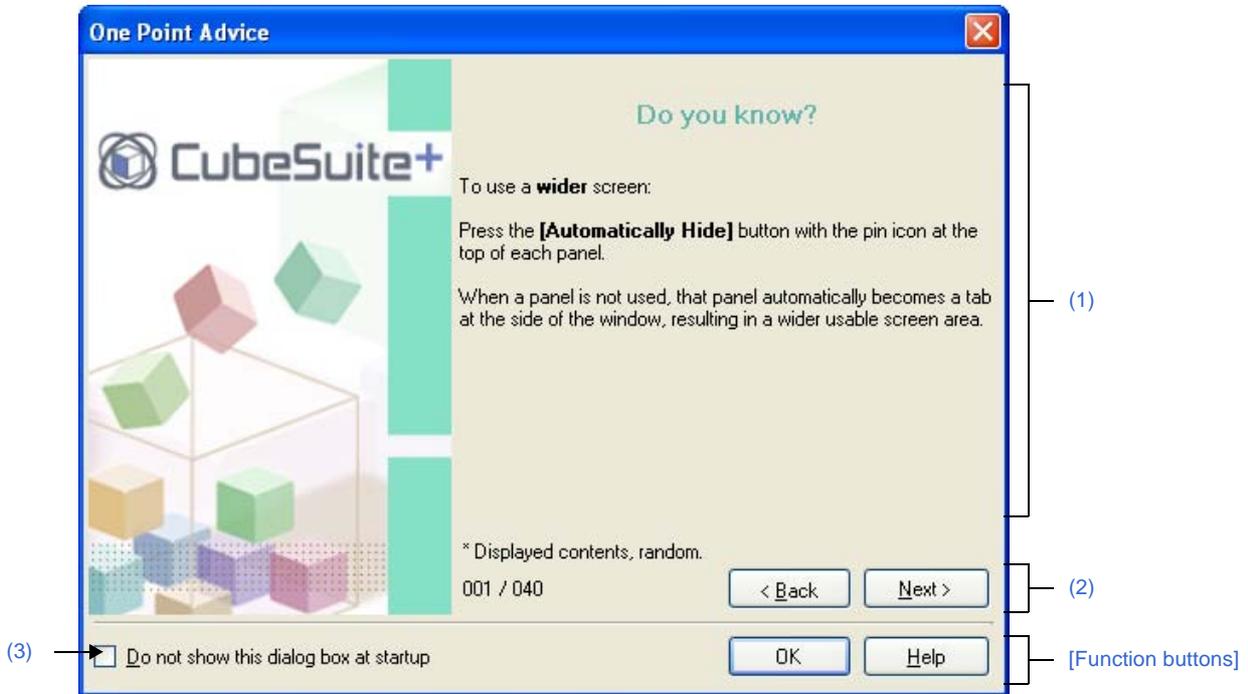


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[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.<br>The button is disabled when the first page is shown. |
| Next | One next page is shown.<br>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.

| ☑ | Does not open this dialog box automatically at startup. |
|---|---|
| ☐ | 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-80.   Other Windows... Dialog Box**
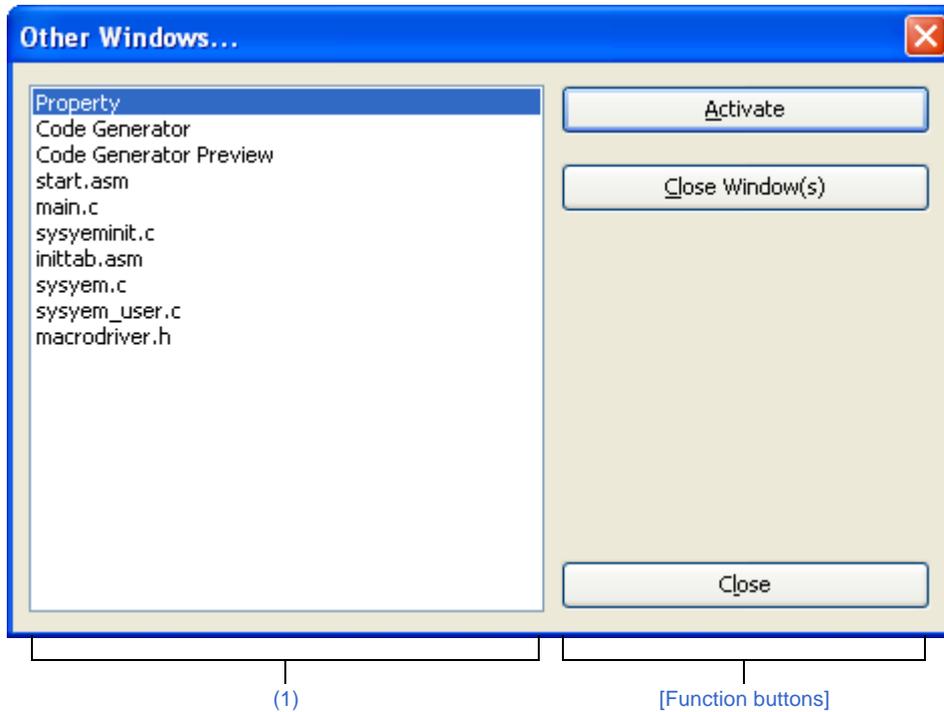


The following items are explained here.

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

## [How to open]

- From the [Window] menu, select [Other Windows...].

      **Remark**    [Other Windows...] only appears when ten or more divide panels are open.

## [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. |

---

| Button | Function |
|--------|----------|
| 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-81.   Open Project Dialog Box**
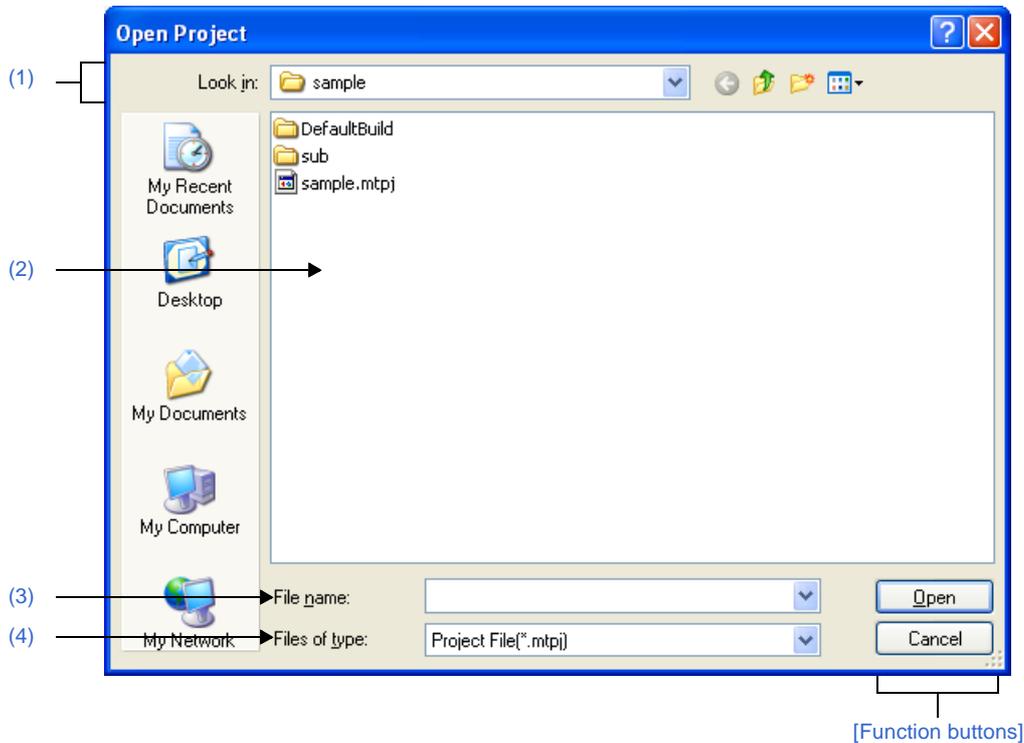


The following items are explained here.

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

**[How to open]**

- From the [Project] menu, select [Open Project...].
- In the Start panel, click the [GO] button in the [Open Existing Project] area.
- 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 type of the project file you want to open.

**(a) When the dialog box is opened from [Project] menu.**

| | |
|---|---|
| Project File (*.mtpj) | Project file |
| Project File for e2 studio (*.rcpc) | Project file for e$^2$ studio |
| Project File for CubeSuite (*.cspj) | Project file for CubeSuite |
| Workspace File for HEW (*.hws) | Workspace file for HEW |
| Project File for HEW (*.hwp) | Project file for HEW |
| Workspace File for PM+ (*.prw) | Workspace file for PM+ |
| Project File for PM+ (*.prj) | Project file for PM+ |

**(b) When the dialog box is opened from the [Open Existing Project] area on the Start panel**

| | |
|---|---|
| Project File(*.mtpj) | Project file |

**(c) When the dialog box is opened from the [Open Existing CubeSuite/High-performance Embedded Workshop/PM+ Project] area on the Start panel**

| | |
|---|---|
| Project File for e2 studio (*.rcpc) | Project file for e$^2$ studio |
| Project File for CubeSuite (*.cspj) | Project file for CubeSuite |
| Workspace File for HEW (*.hws) | Workspace file for HEW |
| Project File for HEW (*.hwp) | Project file for HEW |
| Workspace File for PM+ (*.prw) | Workspace file for PM+ |
| Project File for PM+ (*.prj) | Project file for PM+ |

**(d) When the dialog box is opened from the Create Project dialog box**

| | |
|---|---|
| Project File(*.mtpj) | Project file |
| Subproject File(*.mtsp) | Subproject file |

**[Function buttons]**

| Button | Function |
|---|---|
| Open | - When the dialog box is opened from the [Project] menu (CubeSuite+ project) or the [Open Existing Project] area on the Start panel<br><br>Opens the specified project file.<br><br>- When the dialog box is opened from the [Project] menu (CubeSuite/High-performance Embedded Workshop/PM+ project) or the [Open Existing CubeSuite/High-performance Embedded Workshop/PM+ Project] area on the Start panel<br><br>Opens the Project Convert Setting dialog box.<br><br>- When the dialog box is opened from the Create Project dialog box<br><br>Specifies the selected project file to [Project to be passed] in the caller dialog box. |
| Cancel | Closes this dialog box. |

---

**Open File dialog box**

This dialog box is used to open a file.

**Figure A-82.   Open File Dialog Box**
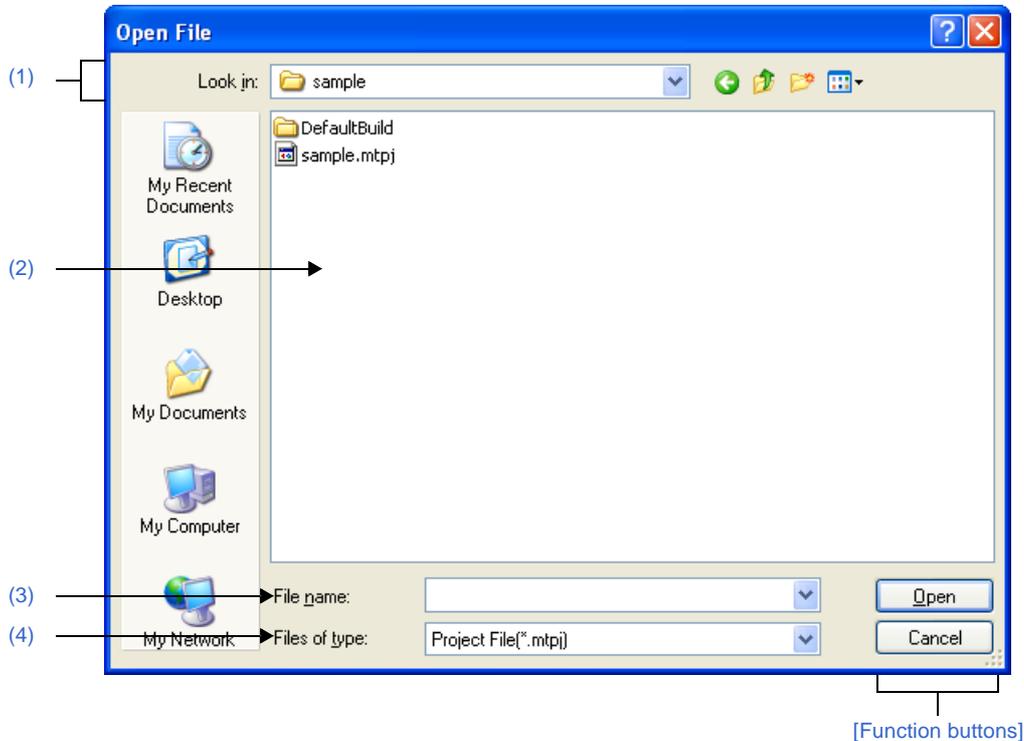


[Function buttons]

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

**[How to open]**

- From the [File] menu, select [Open File...] or [Open with Encoding...].

**[Description of each area]**

**(1)  [Look in] area**
Select the folder that the file you want to open exists.
When you first open this dialog box, the folder is set to "C:\Documents and Settings \*user-name*\My Documents".
The second and subsequent times, this defaults to the last folder that was selected.

**(2)  List of files area**
File list that matches to the selections in [Look in] and [Files of type] is shown.

**(3)  [File name] area**
Specify the file name that you want to open.

---

**(4)  [Files of type] area**

Select the type of the file you want to open.

**(a)  When no project is opened**

| | |
|---|---|
| All files (*.*) | All formats |
| Project File (*.mtpj) | Project file |
| Project File for e2 studio (*.rcpc) | Project file for e$^2$ studio |
| Project File for CubeSuite (*.cspj) | Project file for CubeSuite |
| Workspace File for HEW (*.hws) | Workspace file for HEW |
| Project File for HEW (*.hwp) | Project file for HEW |
| Workspace File for PM+ (*.prw) | Workspace file for PM+ |
| Project File for PM+ (*.prj) | Project file for PM+ |
| Text file (*.txt) | Text format |

**(b)  When the target project type is other than a debug-dedicated project**

| | |
|---|---|
| All files (*.*) | All formats |
| Project File (*.mtpj) | Project file |
| Project File for e2 studio (*.rcpc) | Project file for e$^2$ studio |
| Project File for CubeSuite (*.cspj) | Project file for CubeSuite |
| Workspace File for HEW (*.hws) | Workspace file for HEW |
| Project File for HEW (*.hwp) | Project file for HEW |
| Workspace File for PM+ (*.prw) | Workspace file for PM+ |
| Project File for PM+ (*.prj) | Project file for PM+ |
| C source file (*.c) | C language source file |
| C++ source file (*.cpp; *.cc; *.cp) [CC-RX] | C++ language source file |
| Header file (*.h; *.hpp; *.inc) [CC-RX] | Header file |
| Header file (*.h; *.inc) [CC-RH][CA850][CX][NC30 (Localised support)][CA78K0R][CA78K0] | Header file |
| Assembly source file (*.asm; *.s; *.fsy) [CC-RH] | Assembly source file |
| Assembler source file (*.src; *.s) [CC-RX] | Assembler source file |
| Assemble file (*.s) [CA850] | Assembler source file |
| Assemble file (*.asm; *.s) [CX] | Assembler source file |
| Assemble file (*.a30) [NC30 (Localised support)] | Assembler source file |
| Assemble file (*.asm) [CA78K0R][CA78K0] | Assembler source file |
| Link directive file (*.dir; *.dr) [CA850][CX] | Link directive file |
| Link directive file (*.dr; *.dir) [CA78K0R][CA78K0] | Link directive file |
| Section file (*.sf) [CA850] | Section file |
| Symbol information file (*.sfg) [CX] | Symbol information file |
| Variable and function information file (*.vfi) [CA78K0R] | Variable and function information file |

| | |
|---|---|
| Variable information file (*.vfi) [CA78K0] | Variable and function information file |
| Function information file (*.fin)^Note [CA78K0] | Function information file |
| Link Map file (*.map; *.lbp) [CC-RH] | Link Map file |
| Map file (*.map; *.lbp) [CC-RX][NC30 (Localised support)] | Map file |
| Map file (*.map) [CA850][CX][CA78K0R][CA78K0] | Map file |
| Symbol table file (*.sym) [CA78K0R][CA78K0] | Symbol table file |
| Assemble list file (*.lst) [CC-RX][NC30 (Localised support)] | Assemble list file |
| Stack information file (*.sni) [CC-RH] | Stack information file |
| Intel HEX file (*.hex) [CC-RH] | Intel HEX file |
| Motorola S-record file (*.mot) [CC-RH] | Motorola S-record file |
| Hex file (.hex) [CC-RX][CA850][CX][NC30 (Localised support)] | Hex file |
| Hex file (*.hex; *.hxb; *.hxf) [CA78K0R][CA78K0] | Hex file |
| S record file (*.mot) [CC-RX][NC30 (Localised support)] | S record file |
| Jump table file (*.jmp) [CC-RX] | Jump table file |
| Symbol address file (*.fsy) [CC-RX] | Symbol address file |
| Text file (*.txt) | Text format |

**Note**   This file type is only shown for microcontrollers with a memory bank.

**(c)  When the target project type is a debug-dedicated project**

| | |
|---|---|
| All files (*.*) | All formats |
| Project File (*.mtpj) | Project file |
| Project File for e2 studio (*.rcpc) | Project file for e$^2$ studio |
| Project File for CubeSuite (*.cspj) | Project file for CubeSuite |
| Workspace File for HEW (*.hws) | Workspace file for HEW |
| Project File for HEW (*.hwp) | Project file for HEW |
| Workspace File for PM+ (*.prw) | Workspace file for PM+ |
| Project File for PM+ (*.prj) | Project file for PM+ |
| C source file (*.c) | C language source file |
| Header file (*.h; *.inc) | Header file |
| Assemble file (*.asm; *.s) | Assembler source file |
| Text file (*.txt) | Text format |

**[Function buttons]**

| Button | Function |
|--------|----------|
| Open | - When this dialog box is opened by [Open File...] from the [File] menu<br>  Opens the specified file.<br>- When this dialog box is opened by [Open with Encoding...] from the [File] menu<br>  Opens the Encoding 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-83.   Add Existing Subproject Dialog Box**
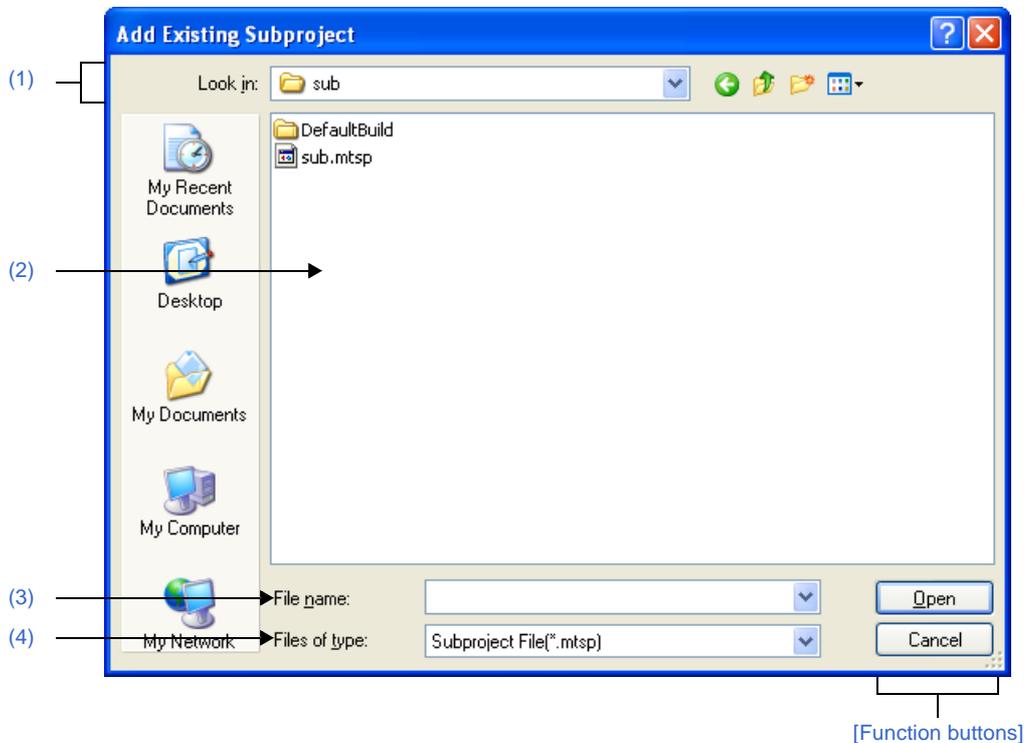


[Function buttons]

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(*.mtsp) | Subproject file |

**[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-84.　Browse For Folder Dialog Box**
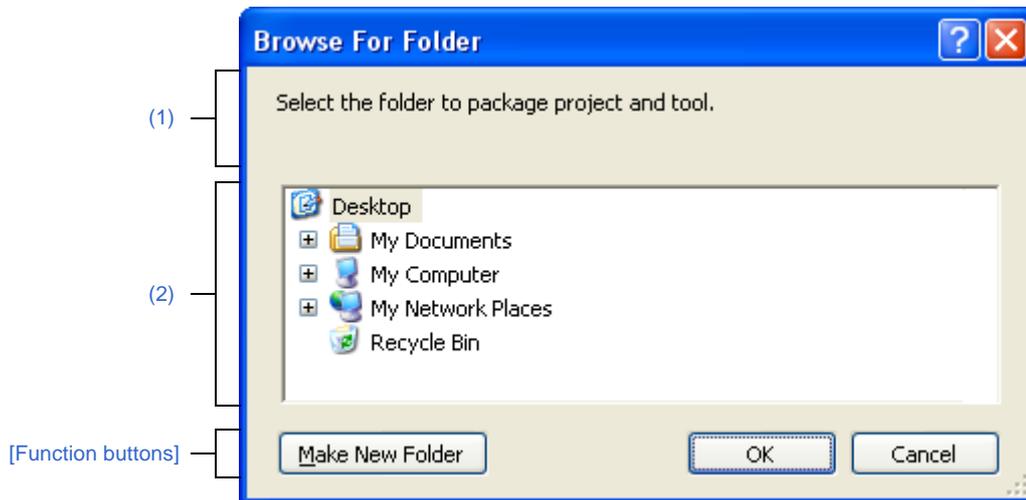


(1)

(2)

[Function buttons]

The following items are explained here.

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

**[How to open]**

- Click the [Copy] button in the History page in the CubeSuite+ Update Manager window.
- From the [Project] menu, select [Save Project and Development Tools as Package].
- On the Project Tree panel, select the Project node and then [Save Project and Development Tools as Package] from the context menu.
- On the Start panel, click  the [Go] button in the [Open Sample Project] area.
- In the Create Project dialog box, click the [Browse...] button 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 [Find in Files] tab or [Replace in Files] tab of the Find and Replace dialog box, click the [...] button.
- In the Pack Settings dialog box, click  the [Browse...] button in the [Place] area.

**[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)  Find 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.<br>The default folder name is "New folder". |
| OK | The designated folder path is set to the path that this dialog box is called from.<br>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-85.   Save Project As Dialog Box**
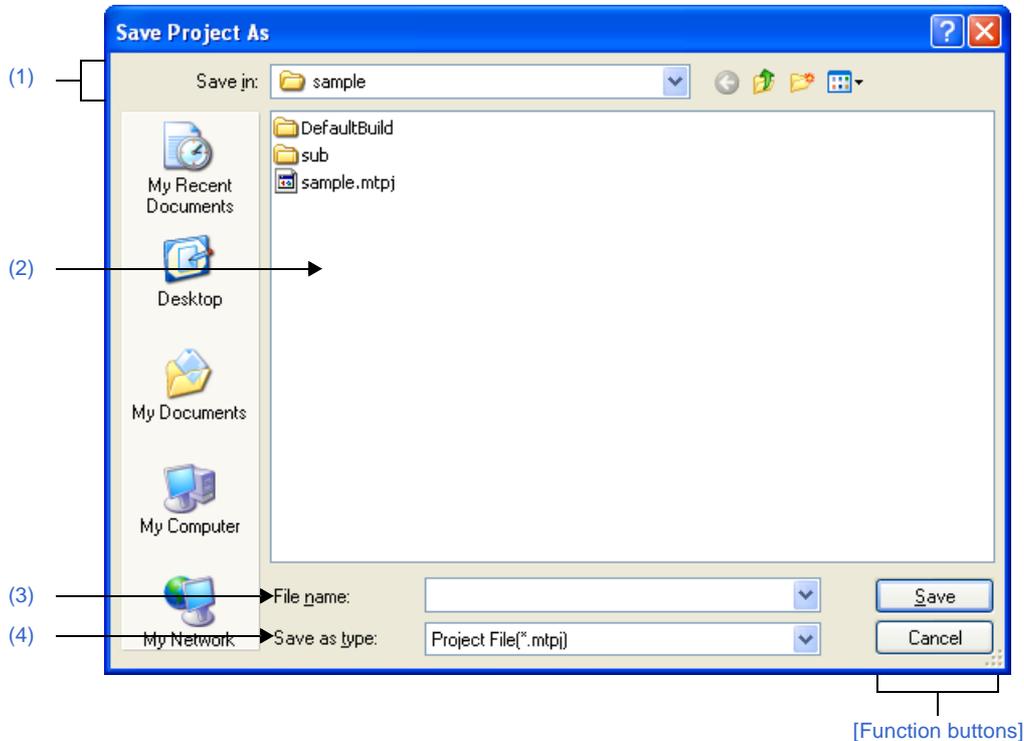


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[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 [Save as type] is shown.

**(3)  [File name] area**

Specify the project file name to save.

---

**(4) [Save as type] area**

The following file type is displayed.

| | |
|---|---|
| Project File(*.mtpj) | Project file |

**[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-86.   Save As Dialog Box**
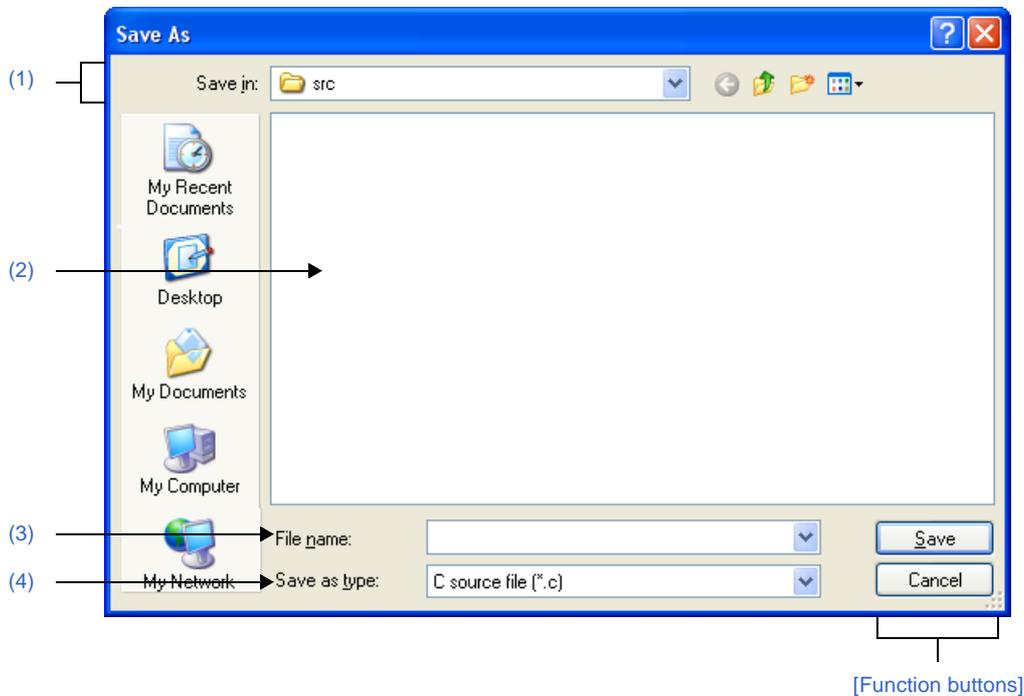


[Function buttons]

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

**[How to open]**

- Focus the Editor panel and from the [File] menu, select [Save *file name* As...].
- Focus the Output panel and from the [File] menu, select [Save Output-*tab name* As...].
- Focus the Error List panel and from the [File] menu, select [Save Error List As...].
- Focus the Python Console panel and from the [File] menu, select [Save Python Console 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 are displayed depend on the file type of the currently editing file.

Remark    The following strings are displayed only for the files registered in the project tree.

| | |
|---|---|
| Preprocessor-expanded output file (*.i) [NC30 (Localised support)] | Preprocessor-expanded output file |
| C source file (*.c) | C language source file |
| C++ source file (*.cpp; *.cc; *.cp) [CC-RX] | C++ language source file |
| Header file (*.h; *.hpp; *.inc) [CC-RX] | Header file |
| Header file (*.h; *.inc) [CA850][CX][NC30 (Localised support)][CA78K0R][CA78K0] | Header file |
| Assembly source file (*.asm; *.s; *.fsy) [CC-RH] | Assembly source file |
| Assembler source file (*.src; *.s) [CC-RX] | Assembler source file |
| Assemble file (*.s) [CA850][CX] | Assembler source file |
| Assemble file (*.asm) [CX][CA78K0R][CA78K0] | Assembler source file |
| Assemble file (*.a30) [NC30 (Localised support)] | Assembler source file |
| Assemble file (*.asm; *.s)Note 1 | Assembler source file |
| Link directive file (*.dir; *.dr) [CA850][CX] | Link directive file |
| Link directive file (*.dr; *.dir) [CA78K0R][CA78K0] | Link directive file |
| Link order specification file (*.mtls) | Link order specification file |
| Section file (*.sf) [CA850] | Section file |
| Symbol information file (*.sfg) [CX] | Symbol information file |
| Variable and function information file (*.vfi) [CA78K0R] | Variable and function information file |
| Variable information file (*.vfi) [CA78K0] | Variable and function information file |
| Function information file (*.fin)Note 2 [CA78K0] | Function information file |
| Link map file (*.map; *.lbp) [CC-RH] | Link map file |
| Map file (*.map) [CA850][CX][NC30 (Localised support)][CA78K0R][CA78K0] | Map file |
| Map file (*.map; *.lbp) [CC-RX] | Map file |
| Symbol table file (*.sym) [CA78K0R][CA78K0] | Symbol table file |
| Intel HEX file (*.hex) [CC-RH] | Intel HEX file |
| Hex file (.hex) [CC-RX][CA850][CX][NC30 (Localised support)] | Hex file |
| Hex file (*.hex; *.hxb; *.hxf) [CA78K0R][CA78K0] | Hex file |
| Motorola S-record file (*.mot) [CC-RH] | Motorola S-record file |
| S record file (*.mot) [CC-RX][NC30 (Localised support)] | S record file |
| Assemble list file (*.prn) [CC-RH] | Assemble list file |

| Assemble list file (*.lst) [CC-RX][NC30 (Localised support)] | Assemble list file |
|---|---|
| Stack information file (*.sni) [CC-RH] | Stack information file |
| Jump table file (*.jmp) [CC-RX] | Jump table file |
| Symbol address file (*.fsy) [CC-RX] | Symbol address file |
| Cross reference file (*.cref) [CC-RX][NC30 (Localised support)] | Cross reference file |
| Link subcommand file (*.clnk) [CC-RX][NC30 (Localised support)] | Link subcommand file |
| Text file (*.txt) | Text format |

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 Output panel**

The following file types (file type) are displayed.

The file can only be saved in text format.

| Text file (*.txt) | Text format |
|---|---|

**[Function buttons]**

| Button | Function |
|---|---|
| Save | Saves the file as the designated file name. |
| Cancel | Closes this dialog box. |

---

| Open Option Setting File dialog box |

This dialog box is used to select an option setting file to import to the [General - Font and Color] category of the Option dialog box.

**Figure A-87.   Open Option Setting File Dialog Box**
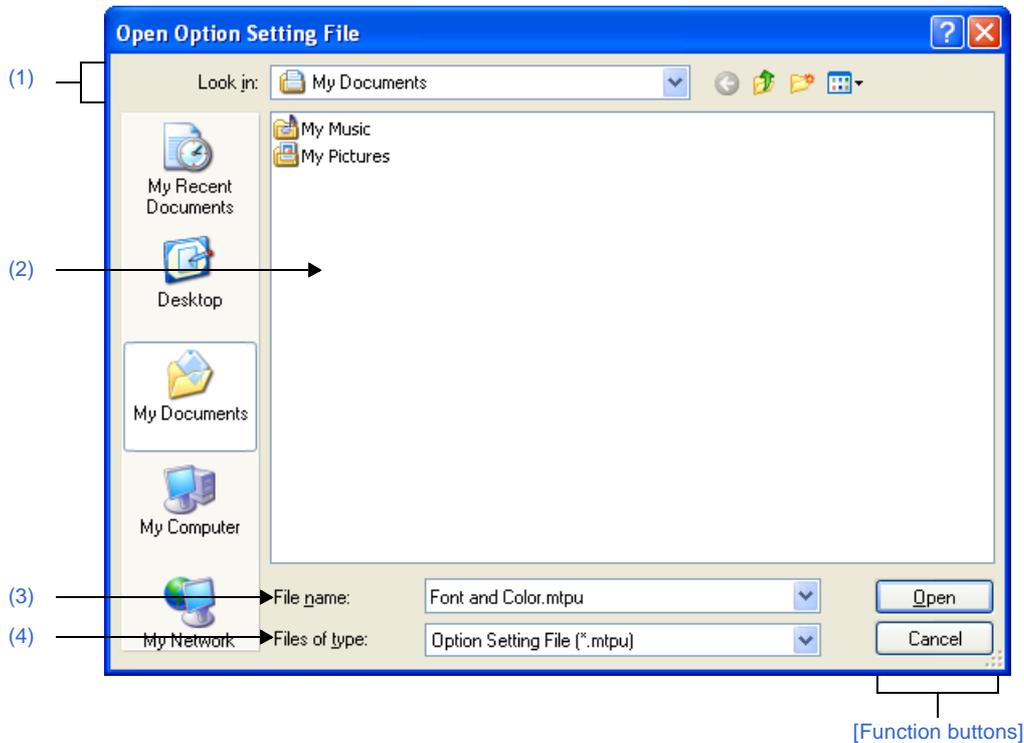


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[How to open]**

- In the [General - External Tools] category of the Option dialog box, click the [Import...] button.

**[Description of each area]**

**(1)  [Look in] area**
Select the folder that the option setting file exists.
When you first select a folder, the folder is set to "C:\Documents and Settings \*user-name*\My Documents".  The second and subsequent times, this defaults to the last folder that was selected.

**(2)  List of files area**
This area displays a list of files matching the conditions selected in [Look in] area and [Files of type] area.

**(3)  [File name] area**
Specify the name of an option setting file.

---

**(4) [Files of type] area**

The following file type is displayed.

| | |
|---|---|
| Option Setting File (*.mtpu) | Option setting file |

**[Function buttons]**

| Button | Function |
|---|---|
| Open | Imports the specified file to the [General - Font and Color] category of the Option dialog box. |
| Cancel | Ignores the setting and closes this dialog box. |

---

**Save Option Setting File dialog box**

---

This dialog box is used to save the setting of the [General - Font and Color] category of the Option dialog box to an option setting file.

**Figure A-88.   Save Option Setting File Dialog Box**
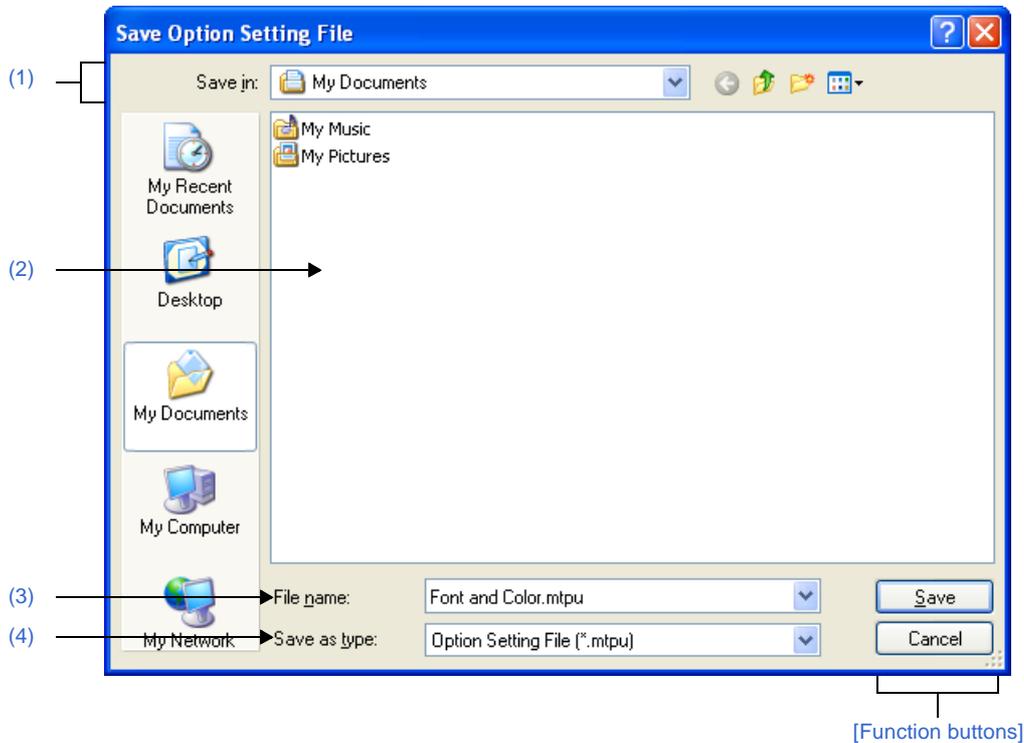


[Function buttons]

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

**[How to open]**

- In the [General - Font and Color] category of the Option dialog box, click the [Export...] button.

**[Description of each area]**

**(1)  [Save in] area**
Select a folder to save an option setting file.
When you first select a folder, 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 [Save in] and [Save as type] is shown.

**(3)  [File name] area**
Specify the name of an option setting file.

---

**(4) [Save as type] area**

The following file type is displayed.

| | |
|---|---|
| Option Setting File (*.mtpu) | Option setting file |

## [Function buttons]

| Button | Function |
|---|---|
| Save | Saves an option setting 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-89.  Select Program Dialog Box**
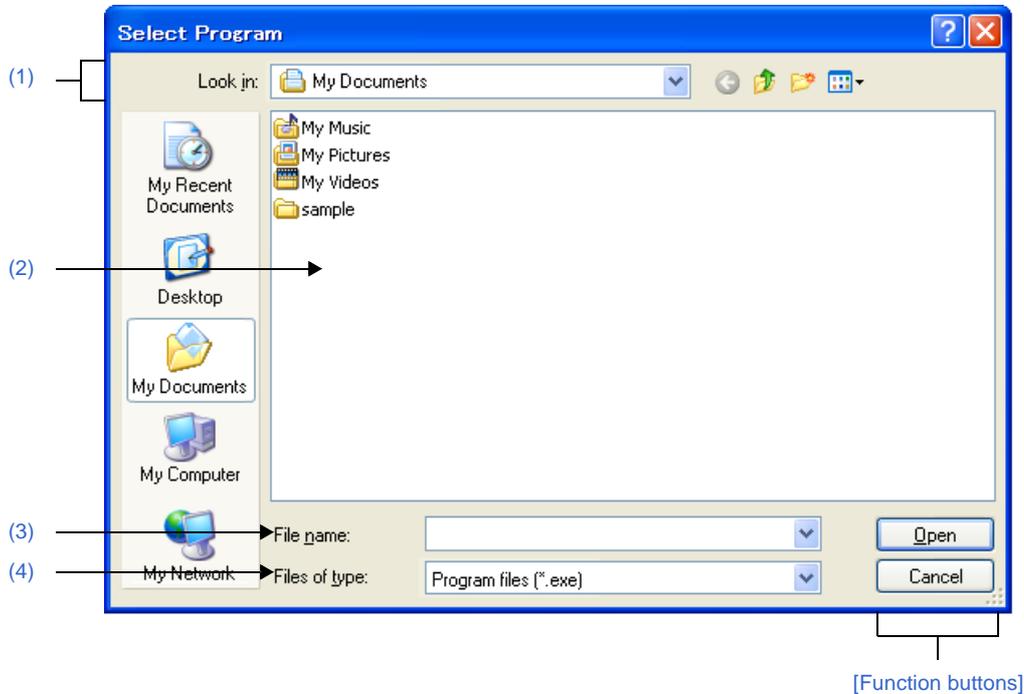


[Function buttons]

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] area**

Select the location (folder) of the executable for the external tool to register from the drop-down list.


**(2)  List of files area**

This area displays a list of files matching the conditions selected in [Look in] area and [Files of type] area.


**(3)  [File name] area**

Specify the name of the executable file for the external tool to register.

---

**(4) [Files of type] area**

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-90. Select External Text Editor Dialog Box**
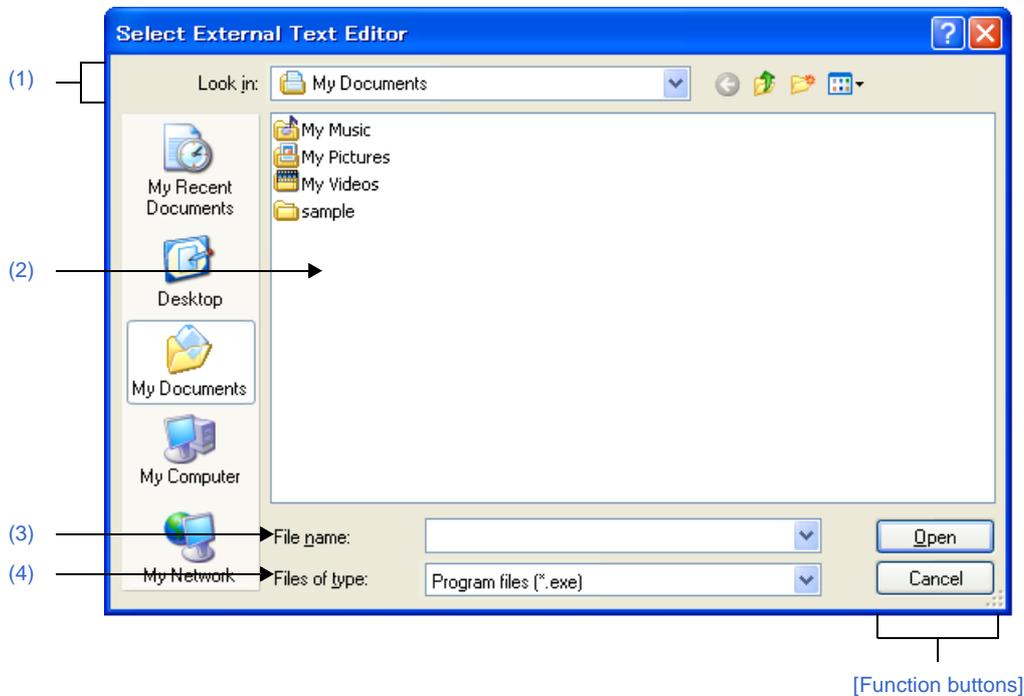


[Function buttons]

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] area**

Select the location (folder) of the executable file for the external text editor to register from the drop-down list.

**(2) List of files area**

This area displays a list of files matching the conditions selected in [Look in] area and [Files of type] area.

**(3) [File name] area**

Specify the name of the executable file for the external text editor.

---

**(4)  [Files of type] area**

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-91.   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.

Use a  print statement to display the result of IronPython functioins.

**[[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. |
| Clear | Clears all output results. |
| Python Initialize | Initializes Python. |
| Select Script File... | Opens the Select Script File dialog box to execute the selected Python script file. |

Select Script File dialog box

This dialog box is used to select the script file for the Python.

**Figure A-92.  Select Script File Dialog Box**
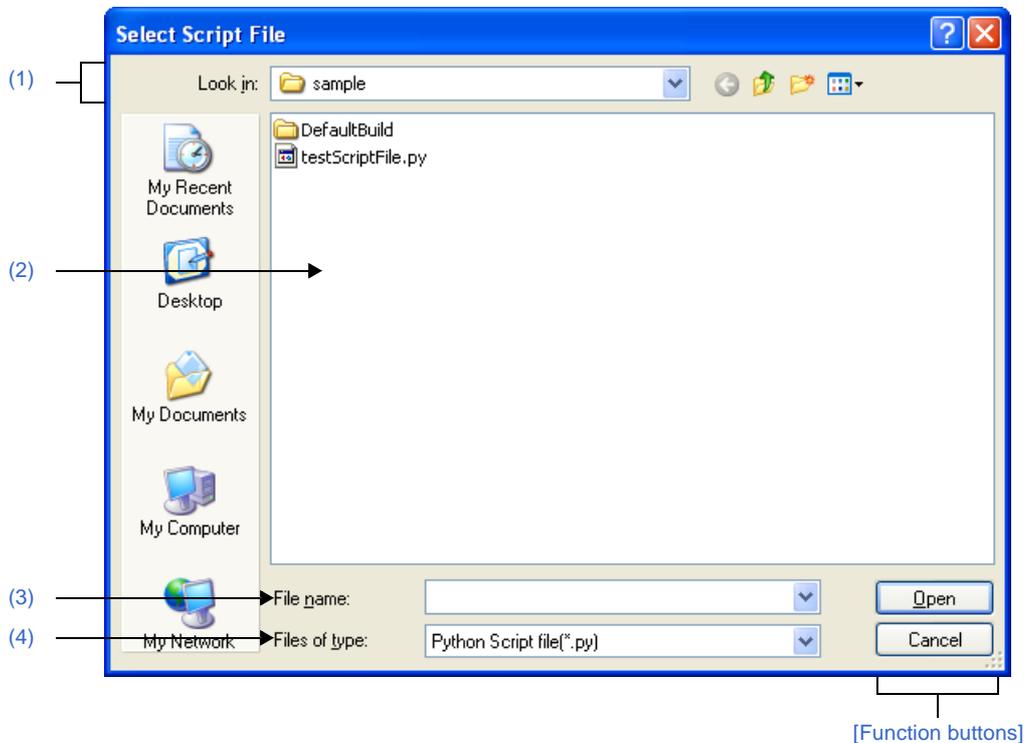


The following items are explained here.
- [How to open]
- [Description of each area]
- [Function buttons]

**[How to open]**

- On the Python Console panel, select [Select Script File...] from the context menu.

**[Description of each area]**

**(1)  [Look in] area**
Select the folder that the script file exists.
When you first select a folder, the folder is set to "C:\Documents and Settings \*user-name*\My Documents".  The second and subsequent times, this defaults to the last folder that was selected.

**(2)  List of files area**
This area displays a list of files matching the conditions selected in [Look in] area and [Files of type] area.

**(3)  [File name] area**
Specify the name of the script file.

**(4)  [Files of type] area**

The following file type is displayed.

| | |
|---|---|
| Python Script File (*.py) | Python script file |

**[Function buttons]**

| Button | Function |
|---|---|
| Open | Executes the specified script file. |
| Cancel | Ignores the setting and closes this dialog box. |

| CubeSuite+ Uninstaller window |
|---|

This window is used to specify one or more installed CubeSuite+ products to uninstall at once.

**Figure A-93.   CubeSuite+ Uninstaller Window**
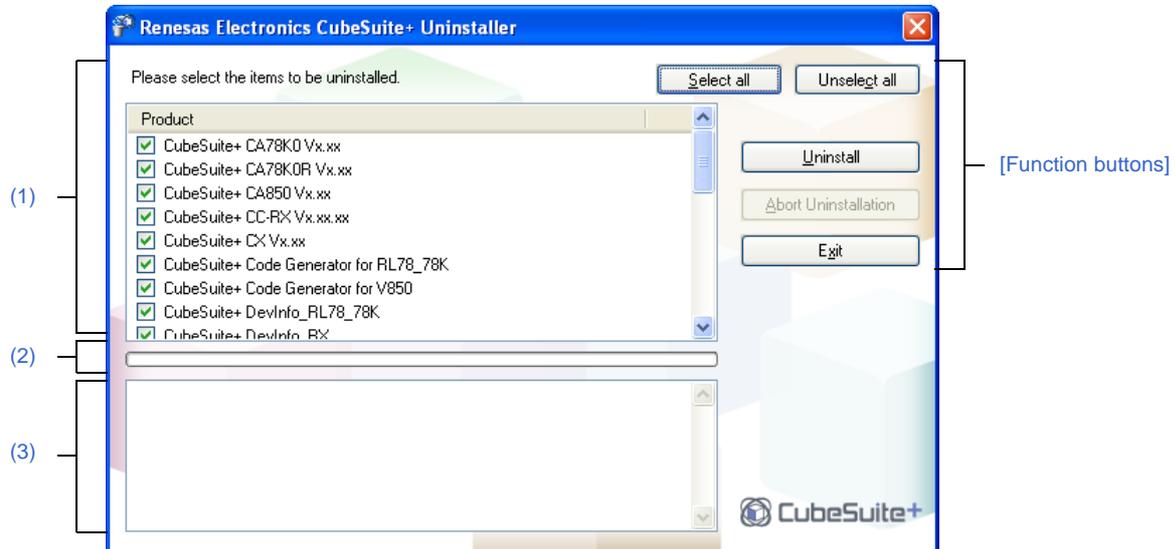


The following items are explained here.

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


**[How to open]**

  - From the Windows [Start] menu, select [Programs] >> [Renesas Electronics CubeSuite+] >> [Uninstaller].

  **Remark**    In Windows 8 or Windows 8.1, double-click on [Uninstaller] on the start screen.


**[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-94.   Project Tree Panel**
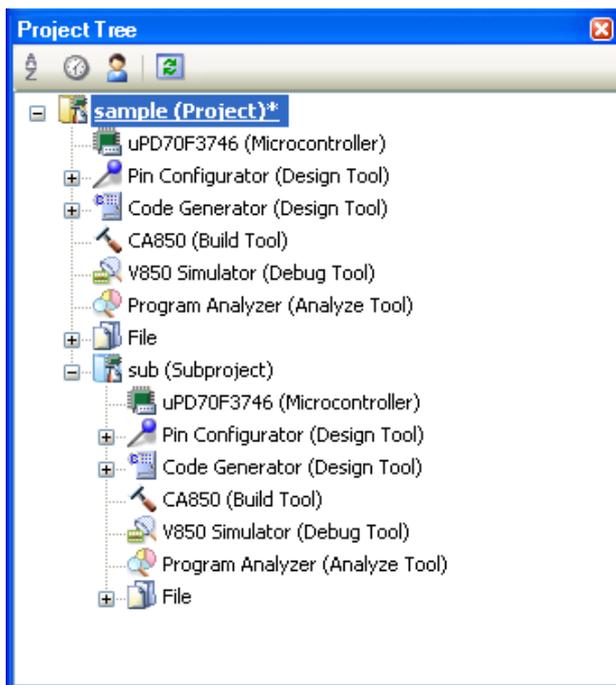


**(2) List control**

This displays a widget such as | Unused ▾ | 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-95.   Sample Context Menu**
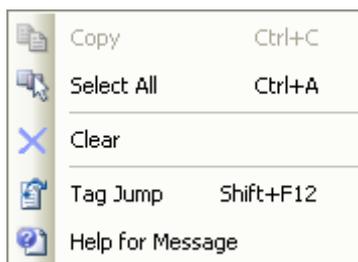
## APPENDIX C   HOW THE MANUALS ARE ORGANIZED

This section describes the manuals of the CubeSuite+ integrated development environment for the microcontrollers (RH850, RX, V850, R8C (Localised support), RL78, 78K0R, and 78K0).

The manuals are organized as follows, in accordance with the software development phases.

**Figure C-1.   List of Manuals**
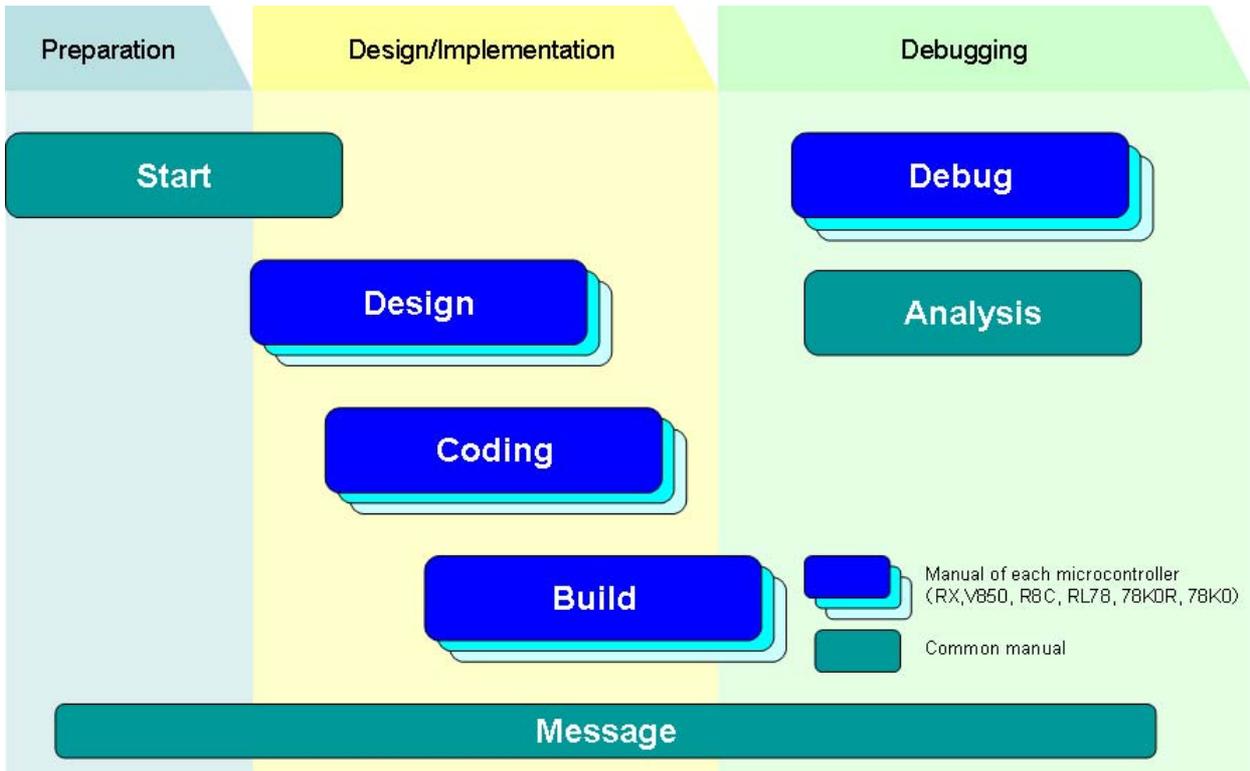


**Remarks 1.**     The manual that you can refer will differ depending on which products are installed.

**2.**     The "CubeSuite+ Integrated Development Environment User's Manual: 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) 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<br>(As for capital letter/small letter of the alphabet, it is pretermission.) |

### D.2    Displaying Icons at Locations of Input Errors

In some of the panels and dialog boxes provided by CubeSuite+, the 🔴 icon will appear at a point where incorect characters are entered or a required item is missing.

**Remark**    Placing the cursor over the 🔴 icon will pop up the information that indicates the characters to be entered (tips to fix the error).

**Figure D-1.   Sample Icons Shown at Locations of Input Errors**
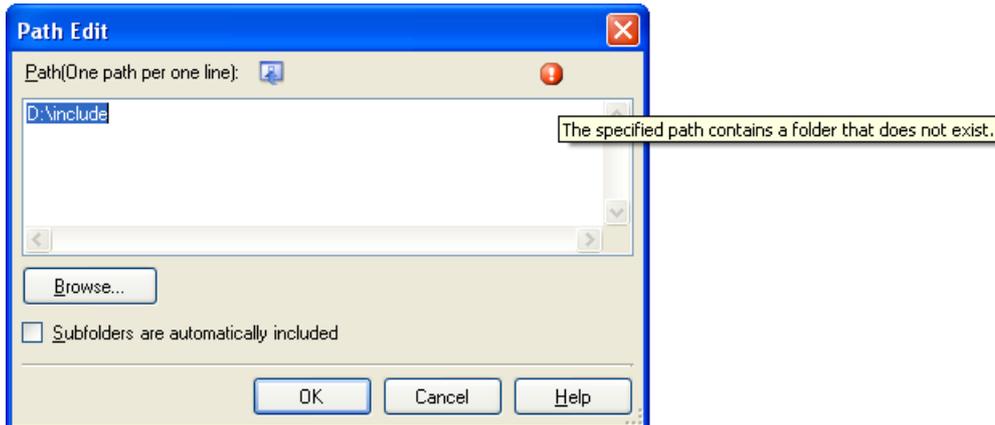
# APPENDIX  E   REGULAR EXPRESSIONS SYNTAX

This section provides detailed explanations of the regular expressions used for the Find and Replace dialog box. The regular expressions supported in CubeSuite+ are based on the Microsoft .NET regular expressions syntax.

**Remark**     Whitespace characters are ignored in all regular expression strings.  Whitespace characters can be specified in the search string by using the \s character.

## E.1   Character Escapes

The following table describes the escape characters and sequences that can be used in regular expressions:

**Table E-1.   List of Character Escapes**

| Escaped Character | Description |
|---|---|
| (Ordinary characters) | Characters other than . $ ^ { [ ( | ) * + ? \ match themselves. |
| \a | Matches a bell (alarm) \u0007. |
| \t | Matches a tab \u0009. |
| \r | Matches a carriage return \u000D. |
| \v | Matches a vertical tab \u000B. |
| \f | Matches a form feed \u000C. |
| \n | Matches a new line \u000A. |
| \e | Matches an escape \u001B. |
| \040 | Matches an ASCII character as octal (exactly three digits). The character \040 represents a space. |
| \x20 | Matches an ASCII character using hexadecimal representation (exactly two digits). |
| \u0020 | Matches a Unicode character using hexadecimal representation (exactly four digits). |
| \ | When followed by a character that is not recognized as an escaped character, matches that character. For example, \* is the same as \x2A. |

## E.2   Character Classes

The following table describes character matching syntax:

**Table E-2.  List of Character Classes**

| Character Class | Description |
|---|---|
| . | Matches any character except \n. When within a character class, the . will be treated as a period character. |
| [aeiou] | Matches any single character in the specified set of characters. |
| [^aeiou] | Matches any single character not in the specified set of characters. |
| [0-9a-fA-F] | Use of a hyphen (-) allows specification of contiguous character ranges. |
| \p{name} | Matches any character in the Unicode general category specified by name (for example, Ll, Nd, Z). See the "E.3  Supported Unicode General Categories", for details on the Unicode general category. |
| \w | Matches any word character, which includes letters, digits, and underscores. |

| Character Class | Description |
|---|---|
| \W | Matches any non-word character. |
| \s | Matches any whitespace character. |
| \S | Matches any non-whitespace character. |
| \d | Matches any decimal digit. |
| \D | Matches any non-digit. |
| [.\w\s] | Escaped built-in character classes such as \w and \s may be used in a character class. This example matches any period, word or whitespace character. |

## E.3    Supported Unicode General Categories

The following table describes the supported Unicode general categories. These categories can be used with the \p and \P character classes. See the "E.2   Character Classes", for details on the character classes.

**Table E-3.   List of Supported Unicode General Categories**

| Unicode General Categories | Description |
|---|---|
| Lu | Letter, Uppercase |
| Ll | Letter, Lowercase |
| Lt | Letter, Titlecase |
| Lm | Letter, Modifier |
| Lo | Letter, Other |
| Mn | Mark, Nonspacing |
| Mc | Mark, Spacing Combining |
| Me | Mark, Enclosing |
| Nd | Number, Decimal Digit |
| Nl | Number, Letter |
| No | Number, Other |
| Pc | Punctuation, Connector |
| Pd | Punctuation, Dash |
| Ps | Punctuation, Open |
| Pe | Punctuation, Close |
| Pi | Punctuation, Initial quote |
| Pf | Punctuation, Final quote |
| Po | Punctuation, Other |
| Sm | Symbol, Math |
| Sc | Symbol, Currency |
| Sk | Symbol, Modifier |
| So | Symbol, Other |
| Zs | Separator, Space |
| Zl | Separator, Line |

| Unicode General Categories | Description |
| --- | --- |
| Zp | Separator, Paragraph |
| Cc | Other, Control |
| Cf | Other, Format |
| Cs | Other, Surrogate |
| Co | Other, Private Use |
| Cn | Other, Not Assigned |

Additional special categories are supported that represent a set of Unicode character categories, as shown in the following table:

**Table E-4.  List of Set of Unicode Character Categories**

| Category | Description |
| --- | --- |
| C | (All control characters) Cc, Cf, Cs, Co, and Cn. |
| L | (All letters) Lu, Ll, Lt, Lm, and Lo. |
| M | (All diacritic marks) Mm, Mc, and Me. |
| N | (All numbers) Nd, Nl, and No. |
| P | (All punctuation) Pc, Pd, Ps, Pe, Pi, Pf and Po. |
| S | (All symbols) Sm, Sc, Sk, and So. |
| Z | (All separators) Zs, Zl, and Zp. |

### E.4   Quantifiers

Quantifiers add optional quantity data to a regular expression. A quantifier expression applies to the character, group, or character class that immediately precedes it.

The following table describes the metacharacters that affect matching quantity:

**Table E-5.   List of Quantifiers**

| Quantifier | Description |
| --- | --- |
| * | Specifies zero or more matches; for example, \w* or (abc)*. Same as {0,}. |
| + | Specifies one or more matches; for example, \w+ or (abc)+. Same as {1,}. |
| ? | Specifies zero or one matches; for example, \w? or (abc)?. Same as {0,1}. |
| {n} | Specifies exactly n matches; for example, (pizza){2}. |
| {n,} | Specifies at least n matches; for example, (abc){2,}. |
| {n,m} | Specifies at least n, but no more than m, matches. |

### E.5   Atomic Zero-Width Assertions

The following table describes the atomic zero-width assertions. The metacharacters described in the following table do not cause the engine to advance through the string or consume characters. They simply cause a match to succeed or fail depending on the current position in the string.

**Table E-6.  List of Atomic Zero-Width Assertions**

| Assertion | Description |
|---|---|
| ^ | Specifies that the match must occur at the beginning of the document or the beginning of the line.<br>For example, ^#region returns only those occurrences of the character string #region that occur at the beginning of a line. |
| $ | Specifies that the match must occur at the end of the string, before \n at the end of the string, or at the end of the line. |
| \A | Specifies that the match must occur at the beginning of the document. |
| \z | Specifies that the match must occur at the end of the document. |
| \b | Specifies that the match must occur on a boundary between \w (alphanumeric) and \W (nonalphanumeric) characters. |
| \B | Specifies that the match must not occur on a \b boundary. |

## E.6    Grouping Constructs

The following table describes the grouping constructs. Grouping constructs allow you to capture groups of sub-expressions and to increase the efficiency of regular expressions with non-capturing look ahead and look behind modifiers.

**Table E-7.   List of Grouping Constructs**

| Grouping Constructs | Description |
|---|---|
| ( ) | Captures the matched substring if used in a find and replace operation. |
| (?=) | Zero-width positive look ahead assertion. Continues match only if the sub-expression matches at this position on the right. For example, (_?=\w) matches an underscore followed by a word character without matching the word character. |
| (?!) | Zero-width negative look ahead assertion. Continues match only if the sub-expression matches at this position on the right. For example, \b(?!un)\w+\b matches words that do not begin with un. |
| (?<=) | Zero-width positive look behind assertion. Continues match only if the sub-expression matches the position on the left. For example, (?<=19)99 matches instances of 99 that follow 19. |
| (?<!) | Zero width negative look behind assertion. Continues match only if the sub-expression does not match this position on the left. |

## E.7    Substitutions

Substitutions are allowed only within find/replace replacement patterns.
Character escapes and substitutions are the only special constructs recognized in a replacement pattern.
 The following table shows how to define named and numbered replacement patterns:

**Table E-8.  List of Substitutions**

| Character | Description |
|---|---|
| $1 | Substitutes the last substring matched by group number 1 (decimal). The second group is number 2 ($2), and so on.<br>For example, the replacement pattern a*$1b inserts the string a* followed by the substring matched by the first capturing group, if any, followed by the string b. |
| $0 | Substitutes a copy of the entire match itself. |
| $& | Substitutes a copy of the entire match itself. |

| Character | Description |
|---|---|
| $$ | Substitutes a single $ literal. |

**Remarks 1.** The * character is not recognized as a metacharacter within a replacement pattern.

   **2.** $ patterns are not recognized within regular expression matching patterns. Within regular expressions, $ designates the end of the string.

## E.8 Other Constructs

The following table describes other regular expression constructs:

**Table E-9. List of Other Constructs**

| Constructs | Description |
|---|---|
| " " | Encapsulates a fixed string of characters. |
| { } | Provides a call to a lexical macro. The use of a WordMacro (which is similar to \w) would appear as {Word-Macro}. |
| (?# ) | Inline comment inserted within a regular expression. The comment terminates at the first closing parenthesis character. |
| \| | Provides an alternation construct that matches any one of the terms separated by the \| (vertical bar) character. For example, cat\|dog\|tiger. The left most successful match wins. |

## APPENDIX  F   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+ (CC-RH/CC-RX/CA850/CX/NC30 (Localised support)/CA78K0R/CA78K0).

### F.1     Overview

CubeSuite+ automatically determines the load module file or hex file output by the build tool provided by CubeSuite+ (CC-RH/CC-RX/CA850/CX/NC30 (Localised support)/CA78K0R/CA78K0) 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 F-1.   Operational Sequence for Using a Debug-dedicated Project**
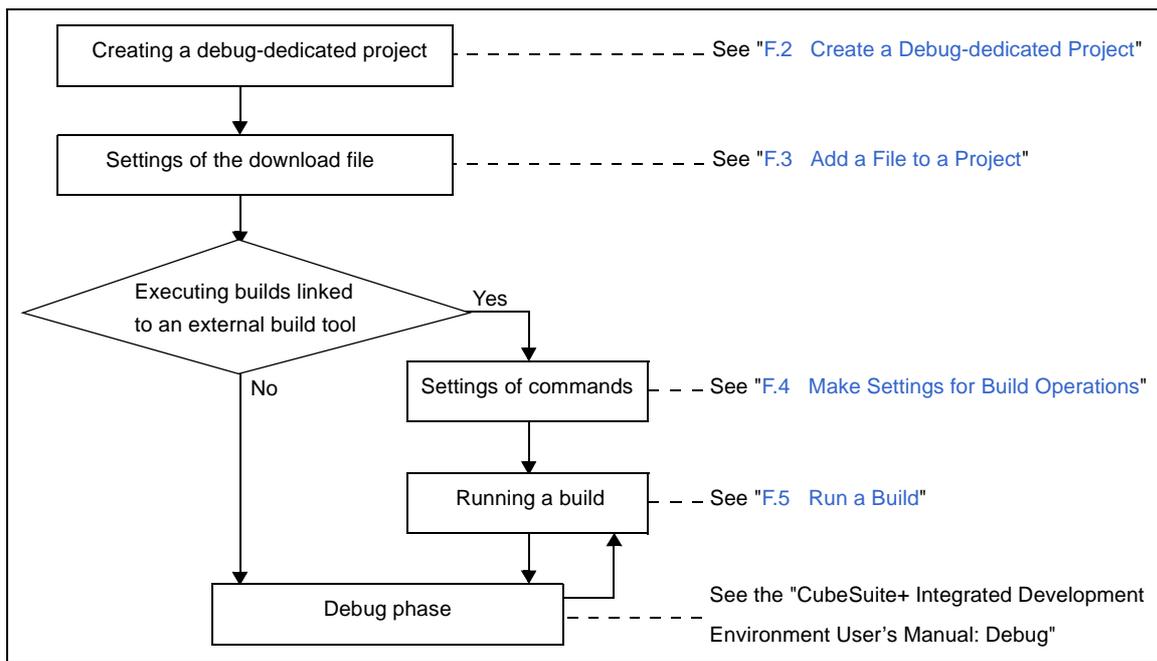


**Caution     The design tool (code generator) and the analyze tool are not available for a debug-dedicated project.**

### F.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 F-2.   Create Project Dialog Box (Creating a Debug-dedicated Project)**
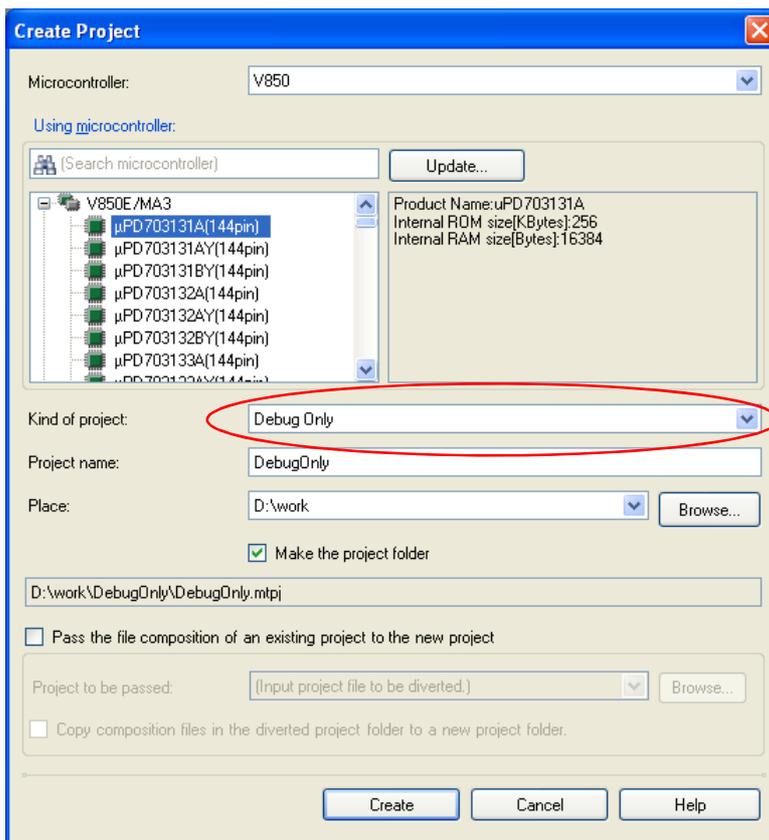


When you click the [Create] button after configuring all settings[Note], the project file of the debug-dedicated project is created in the location specified in the [Place] item and the structure of the created debug-dedicated project is displayed as a tree in the Project Tree panel.

**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 F-3.   Project Tree Panel (After Creating a Debug-dedicated Project)**
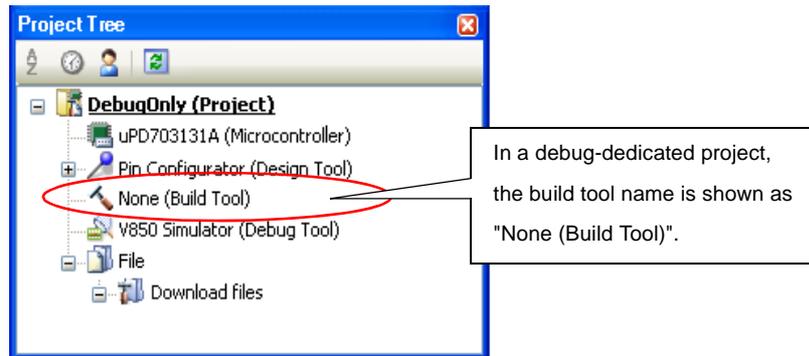


F.3    Add a File to a Project

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.

### F.3.1    Add a download file

Add a load module file to the debug-dedicated project as the download file to debug.

**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+ Integrated Development Environment User's Manual: Debug", for details on the downloading.

**2.**    Multiple load module files can be added to the debug-dedicated project. [V850]

A load module file can be added by the following methods.
- Adding an existing file
- Creating and adding an empty file

**(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 F-4.   Project Tree Panel (Adding a Load Module File)**
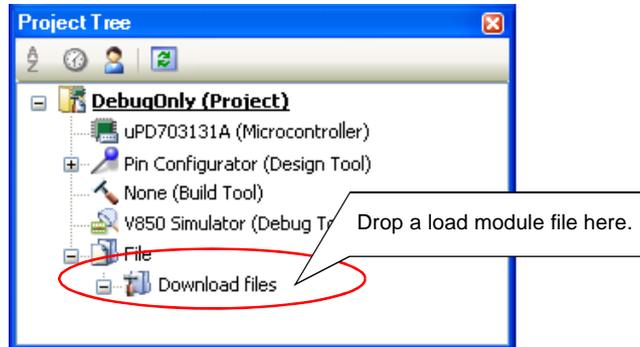


**(2) Creating and adding an empty file**

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 F-5.   Add File Dialog Box (Adding a Load Module File)**
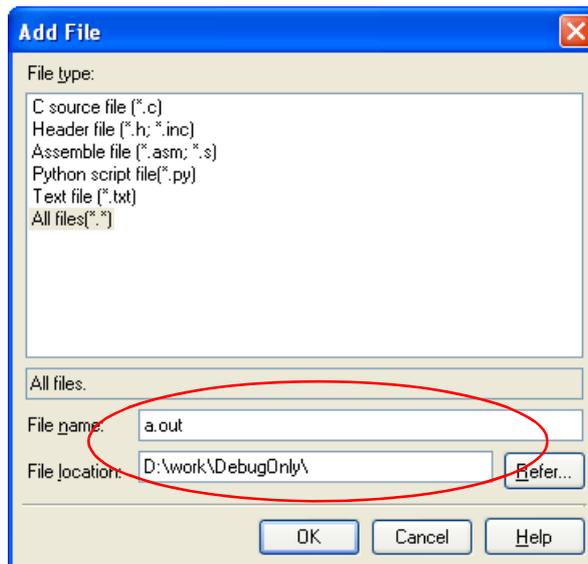


In the dialog box, specify the file name to be created newly and the location in which it is created, and then click the [OK] button.

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

**Figure F-6.   Project Tree Panel (After Adding Load Module File "a.out")**
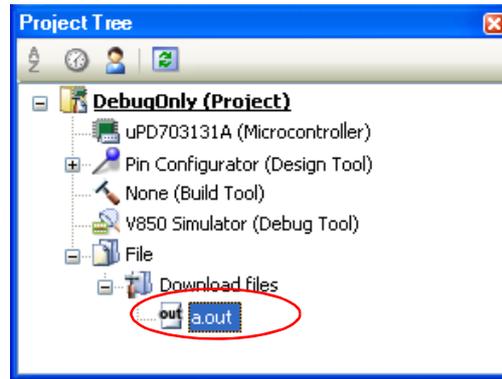


### F.3.2     Add C source files and other files

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.

**Caution     To perform source level debugging, a load module file with the symbol information must be added as a download file to debug (see "F.3.1   Add a download file").**

When you add load module files to the Download file node of the project tree, and download to the debug tool, based on source file information obtained from the download file, source files are automatically added to the project tree.

**Cautions 1.     When a load module file has been added in the [Download File Settings] tab on the Property panel of the debug tool, source files are not added to the project tree.**

**2.     Whether or not to automatically add source files to the project tree is determined by a setting in the [General - Build/Debug] category in the Option dialog box.**

When there are files which cannot be added by the above method, use the following methods to add them.

- 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 F-7.   Project Tree Panel (Adding a File)**
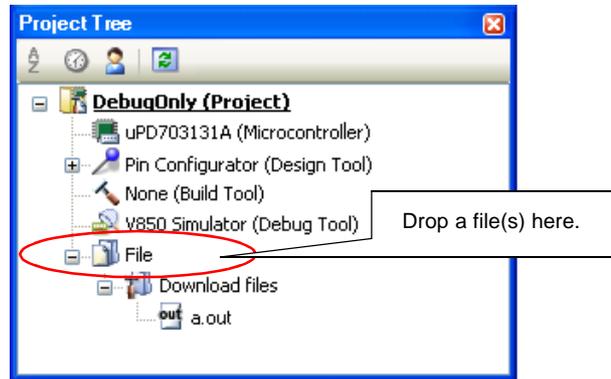


**(b)  Add a folder**

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 F-8.   Add Folder and File Dialog Box**
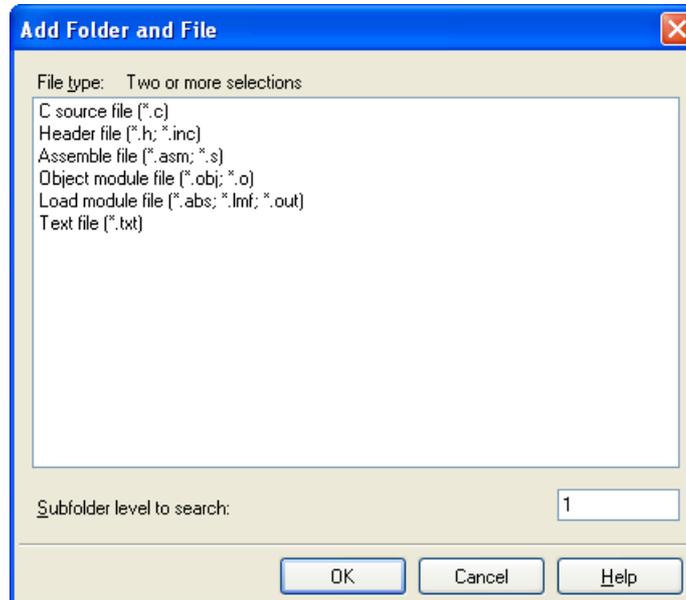


In the dialog box, select the types of the files to be added and specify the number of levels of the subfolder to be added to the debug-dedicated project.  At this time, you can select multiple file types by left clicking while holding down the [Ctrl] or [Shift] key.  If nothing is selected, it is assumed 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 F-9.   Add File Dialog Box (Adding a File)**
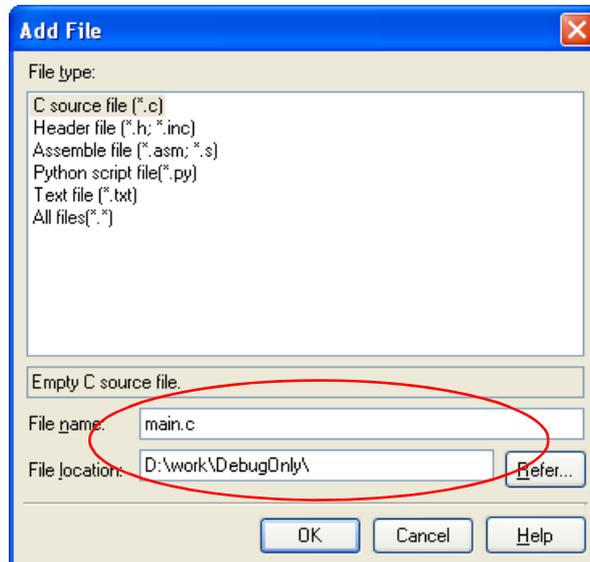


In the dialog box, specify the file name to be created newly and the location in which it is created, and then click the [OK] button.

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 F-10.   Project Tree Panel (After Adding File "main.c")**
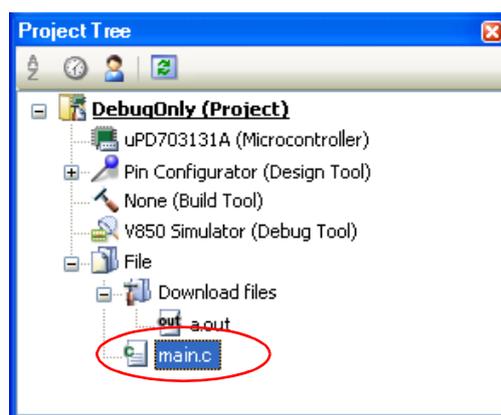
**Figure F-11.   Project Tree Panel (After Adding Folder "src")**
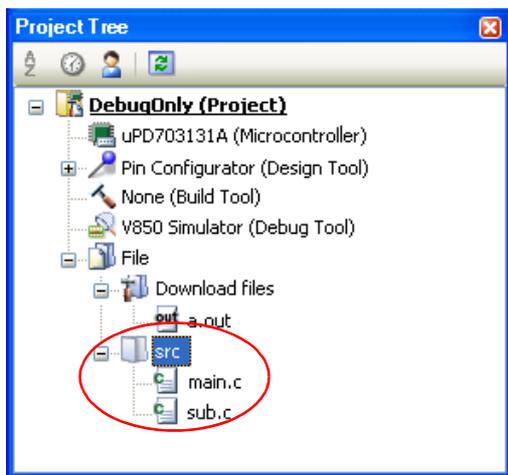


By double-clicking the added file name on the project tree, you can open the Editor panel and edit the contents of the file directly.

**Remarks 1.**   By dragging a file and dropping it onto the Editor panel, you can open the file in the Editor panel.
**2.**   When the environment is set to use an external text editor on the Option dialog box, the file is opened with the external text editor that has been set.
Other files are opened with the applications associated by the host OS.

**F.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 F-12.   [Remove from Project] Item**
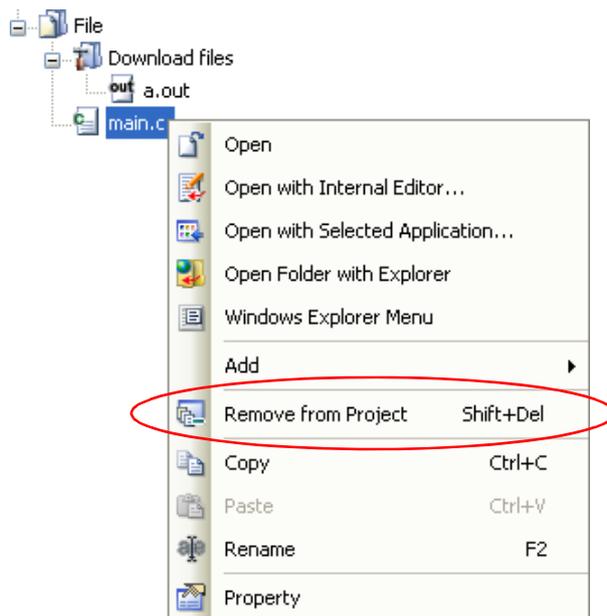
## F.4     Make Settings for Build Operations

Configure CubeSuite+ to execute builds linked to an external build tool (e.g. a compiler/assembler other than the build tool provided by CubeSuite+).

### F.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 "F.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 F-13.   Property Panel: [Build Options] Tab (Build Category)**
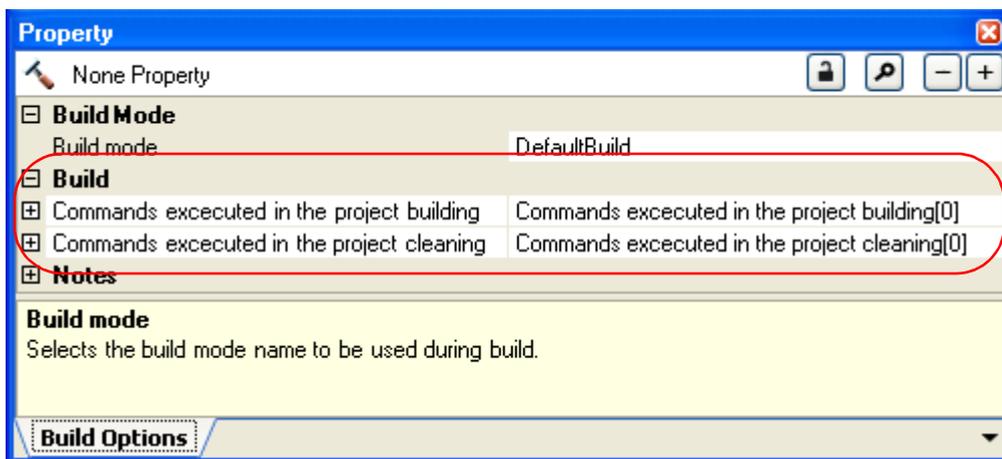


Below are how to set the commands.

- 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 "F.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 F-14.   [Commands executed in the project building] Property**



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

**Figure F-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 1023 characters per line, up to 64 lines.

**Remarks 1.**     This property supports placeholders.

If a line is double clicked in [Placeholder], the placeholder will be reflected in [Text].

**2.**     When "#!python" is described in the first  line, the contents from the second line to the last line are regarded as the script of the Python console, and then executed when running a build.

The placeholders can be described in the scriput.

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

**Figure F-16.   [Commands executed in the project building] Property (After Setting Commands)**



**(2)  Setting the command when running a clean**

Set the command to be executed when running a clean (see "F.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 F-17.   [Commands executed in the project cleaning] Property**
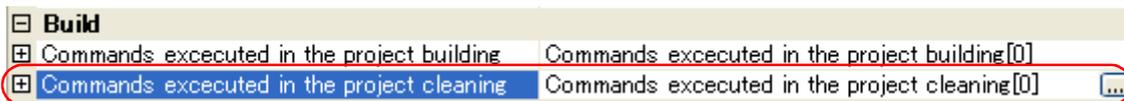


If you click the [...] button, the Text Edit dialog box will open.

**Figure F-18.   Text Edit Dialog Box**
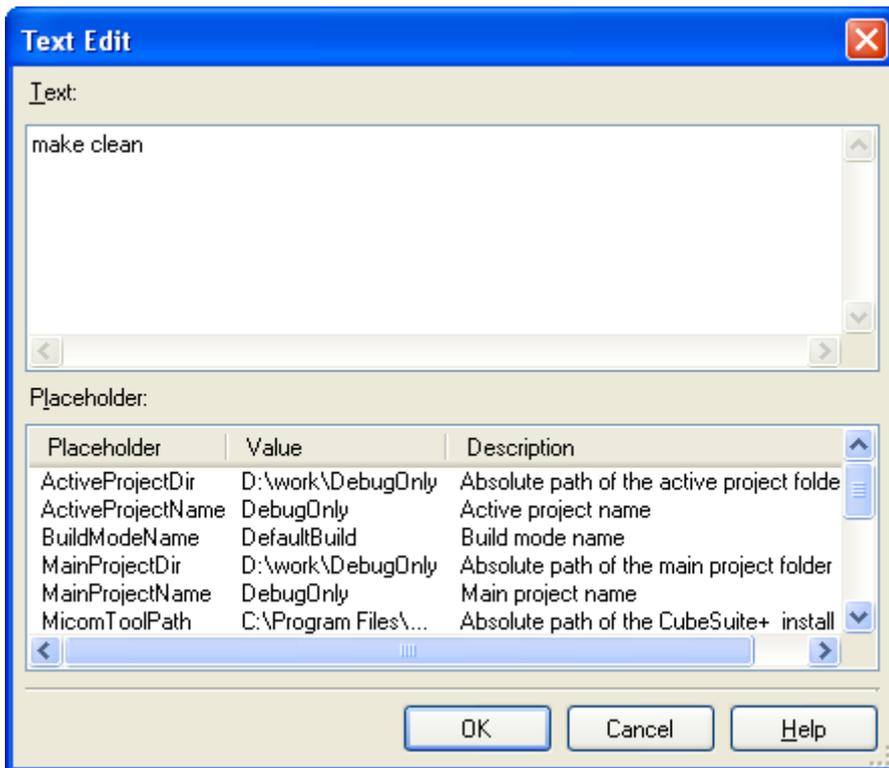


In [Text], enter the command to be executed with one item per line.
The commands can be specified up to 1023 characters per line, up to 64 lines.

**Remarks 1.**      This property supports placeholders.
                    If a line is double clicked in [Placeholder], the placeholder will be reflected in [Text].

**2.**      When "#!python" is described in the first line, the contents from the second line to the last line are
            regarded as the script of the Python console, and then executed when running a clean.
            The placeholders can be described in the script.

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

**Figure F-19.   [Commands executed in the project cleaning] Property (After Setting Commands)**

To change the specified commands, you can use the [...] button or enter the command directly in the text box of the subproperty.

**F.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 "F.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 F-20.　Build Mode Settings Dialog Box**
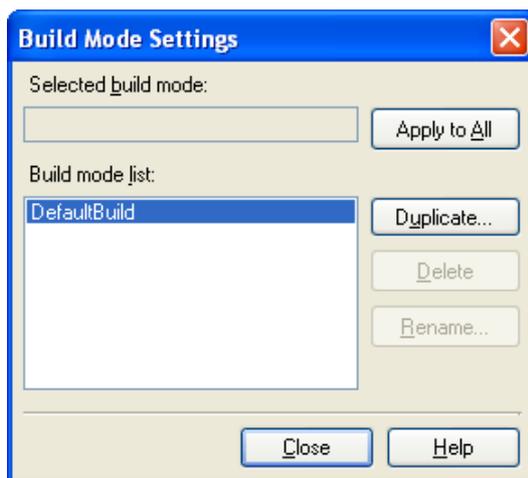


Select the build mode to be duplicated from [Build mode list] and click the [Duplicate...] button. The Character String Input dialog box will open.

**Figure F-21.　Character String Input Dialog Box**
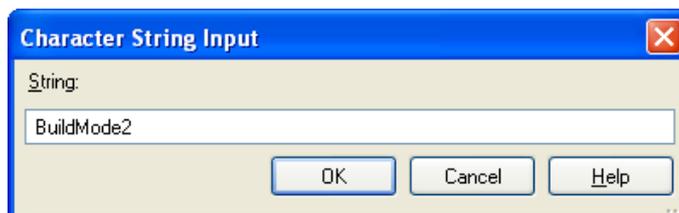


In the dialog box, enter the name of the build mode to be created and then click the [OK] button. The build mode with that name will be duplicated.

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 F-22. Build Mode Settings Dialog Box (After Adding Build Mode)**
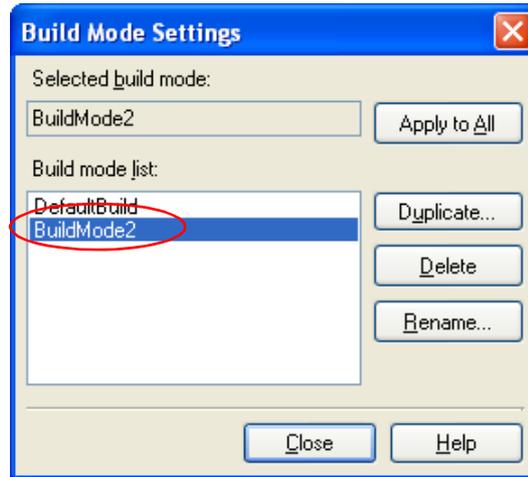


**Remarks 1.** Creating a build mode is regarded a project change.
When closing the project, you will be asked to confirm whether or not to save the build mode.

**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 F-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 F-24.   Build Mode Settings Dialog Box**
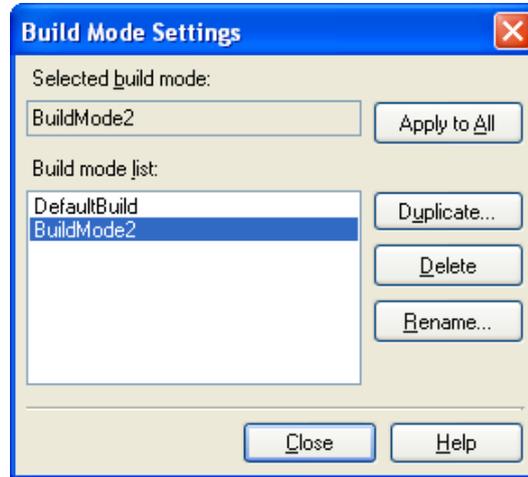


Select the build mode to be changed from [Build mode list].  The selected build mode will be displayed in [Selected build mode].

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 sub-project 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).**

**F.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".

**Caution**　　**The active project is the project subjected to debugging.**
**It is not possible to set the active project while connected to the debug tool, and therefore you must disconnect the debug tool for active project setting.**

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 F-25.   [Set *selected project* as Active Project] Item**
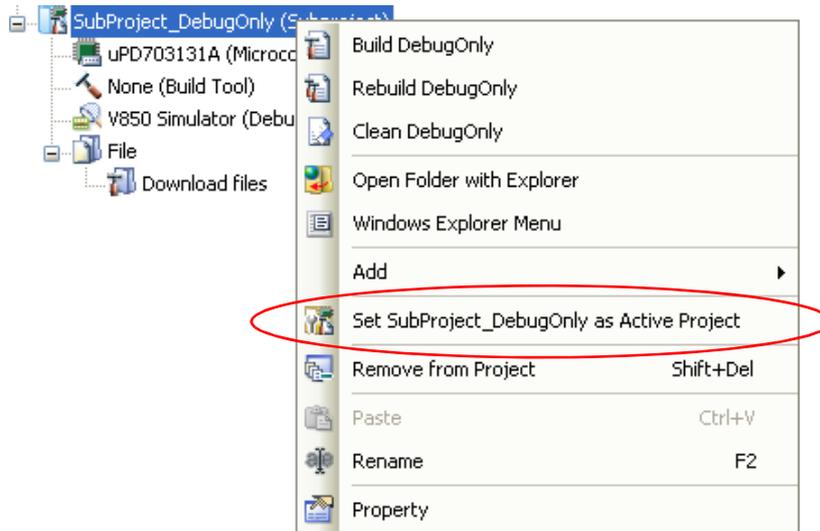


When a project is set as the active project, that Project node is underlined as follows.

**Figure F-26.   Active Project**
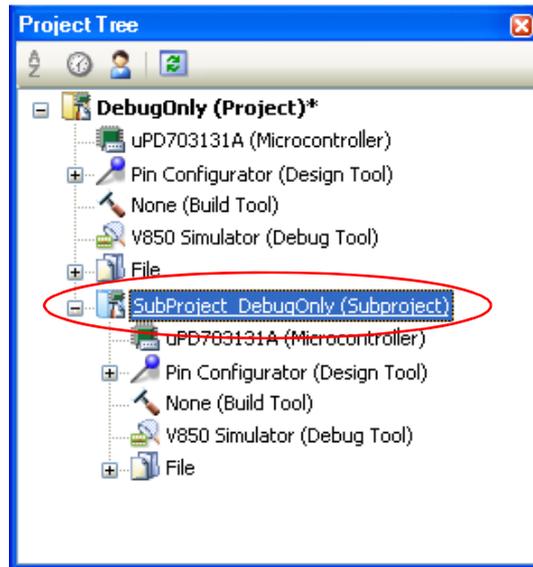


**Remarks 1.**　　Immediately after creating a project newly, the main project is the active project.

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

**Caution**　　**When you run a build using the toolbar button, the build applies to the whole project.**
　　　　　　　**To build only the active project, run the build from the context menu of the active project.**

## F.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 F-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.<br>"Build" can run a build of only updated filesNote.<br>-> See "F.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.<br>"Rebuild" can run a build of all build target filesNote.<br>-> See "F.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.<br>"Clean" can delete all the intermediate files and generated files output by running a buildNote.<br>-> See "F.5.3 Run a clean". |
| Rapid build | Runs a build in parallel with the change of the build setting.<br>-> See "F.5.4 Run a rapid build". |
| Batch build | Runs builds in batch with the build modes that the project has.<br>-> See "F.5.5 Run a batch build". |

**Note** It is assumed that a proper command has been set in the Property panel (see "F.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 F-27. Build Execution Results (Build, Rebuild, or Batch Build)**
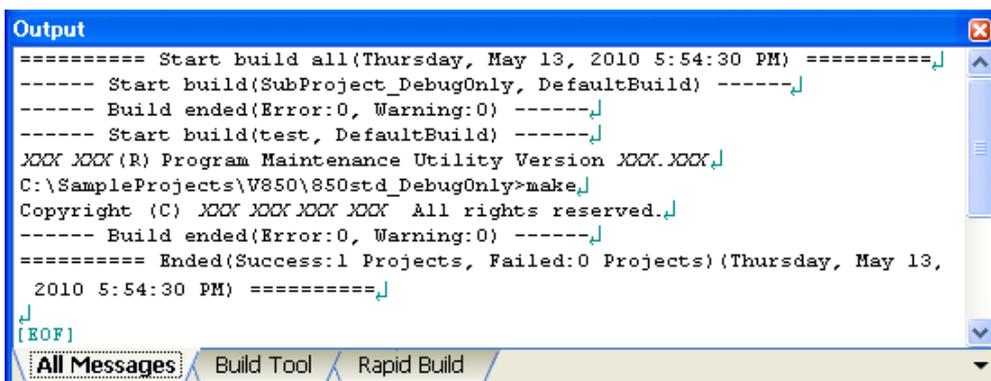
**Figure F-28.   Build Execution Results (Rapid Build)**
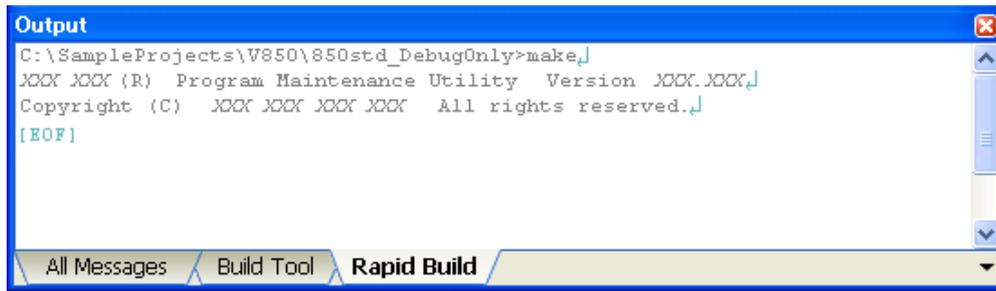
```
Output                                                              ☒
C:\SampleProjects\V850\850std_DebugOnly>make↵
XXX XXX (R)  Program Maintenance Utility   Version  XXX.XXX↵
Copyright (C)  XXX XXX XXX XXX   All rights reserved.↵
[EOF]




 All Messages  \  Build Tool  \  Rapid Build  /                     ▼
```

>　　　**Remark**　　The text in the [Rapid Build] tab becomes dimmed.

### F.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 "F.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+ Integrated
>　　　　　　　　　　　Development Environment User's Manual: 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.

>　　　**Remark**　　When a dependent project is set for the active project, a build of the dependent project is also run.

**Figure F-29.   [Build *active project*] Item**
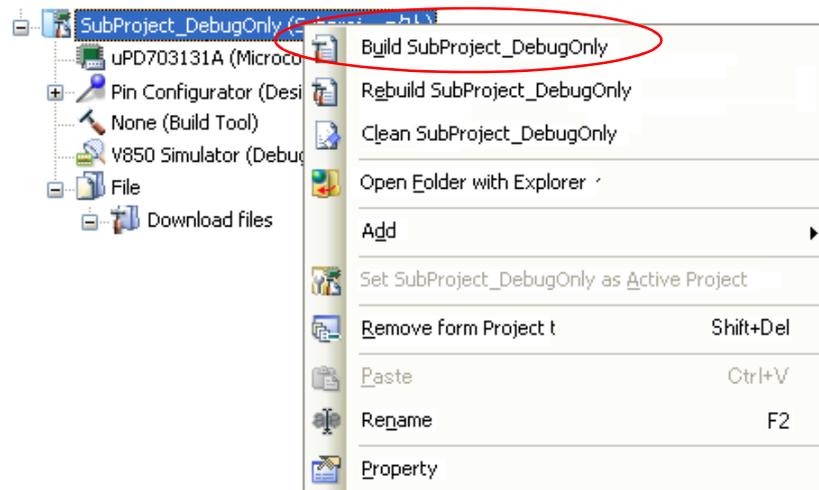


### F.5.2   Run a rebuild

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 "F.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 [icon] 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+ Integrated Development Environment User's Manual: 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.

**Remark**    When a dependent project is set for the active project, a rebuild of the dependent project is also run.

**Figure F-30.   [Rebuild *active project*] Item**
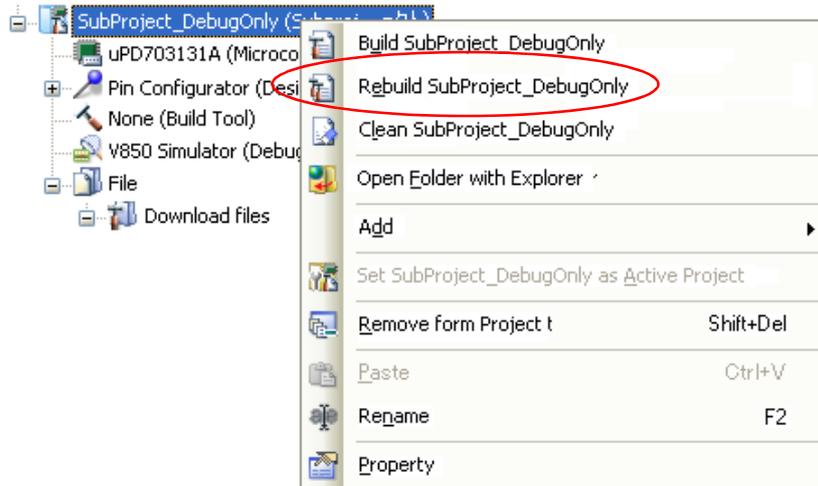


### F.5.3    Run a clean

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 "F.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 F-31.   [Clean Project] Item**
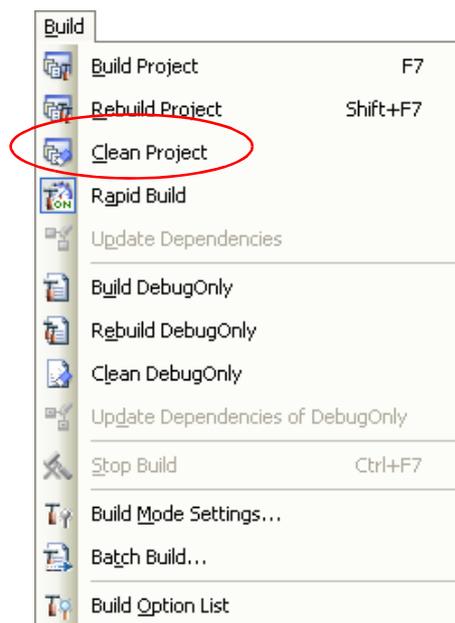
**Remarks 1.**    Cleans are run in the order of subproject, main project.
Subprojects are cleaned in the order that they are displayed on the project tree.  You can change the display order of the subprojects by dragging the subproject to be moved and dropping it on the desired location.

**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+ Integrated Development Environment User's Manual: 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 F-32.   [Clean *active project*] Item**
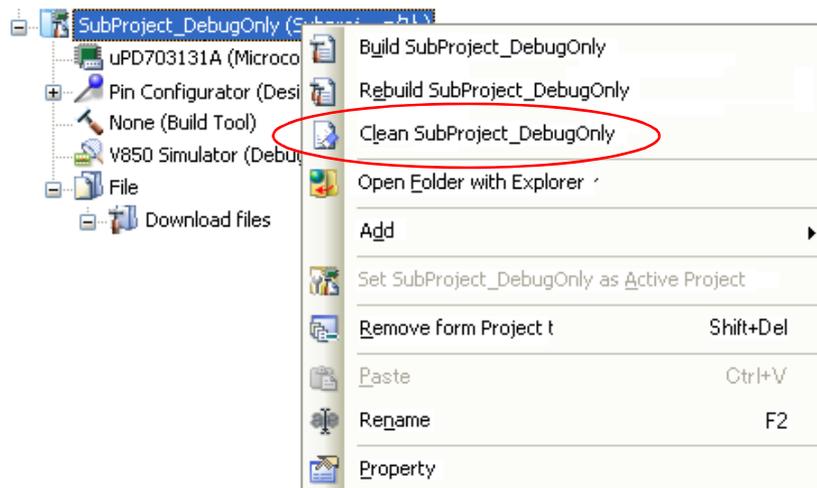


**F.5.4     Run a rapid build**

CubeSuite+ can automatically start a build (see "F.5.1   Run a build") when one of the following events occurs.
- When any one of the following files that are added to the debug-dedicated project is updated:
  (C source file, assembler source file, header file, jump table file, and symbol address file)
- When the C source file, assembler source file, header file, jump table file, or symbol address file is added to or removed from the debug-dedicated project
- When the property of the build tool in the Property panel of the debug-dedicated project is changed

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

**Figure F-33.   [Rapid Build] Item**

[When a rapid build Is valid]                              [When a rapid build Is invalid]
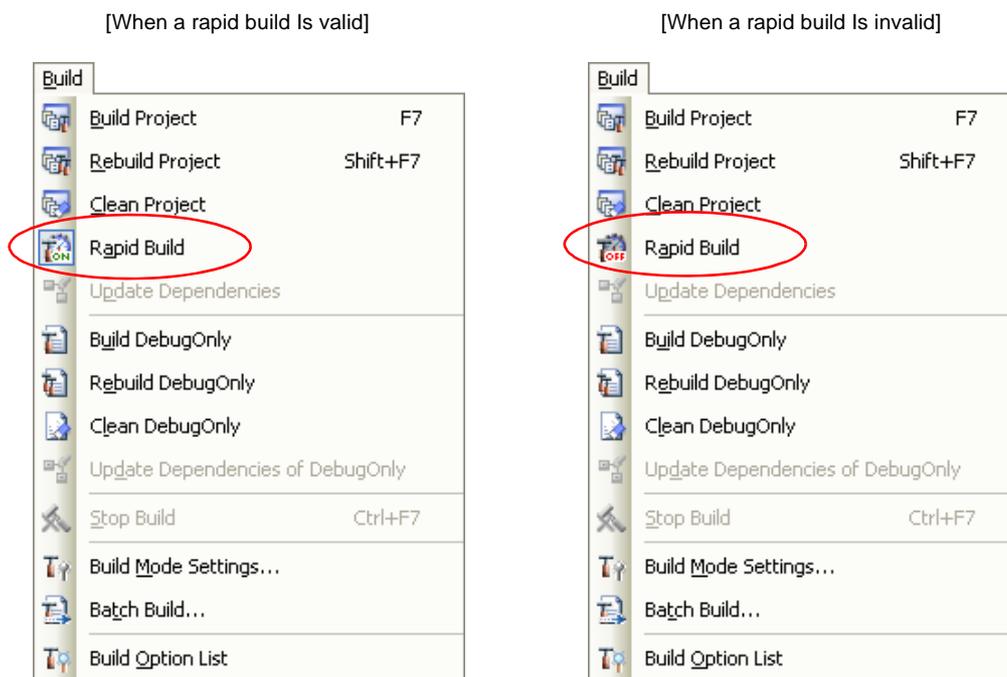


**Caution**    **This function is valid only when editing source files with the Editor panel.**

**Remarks 1.**    After editing source files, it is recommend to save frequently by pressing the [Ctrl] + [S] key.
       **2.**    Enable/Disable setting of the rapid build applies to the entire project (main project and subprojects).
       **3.**    If you disable a rapid build while it is running, it will be stopped at that time.

**F.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 F-34.   Batch Build Dialog Box**
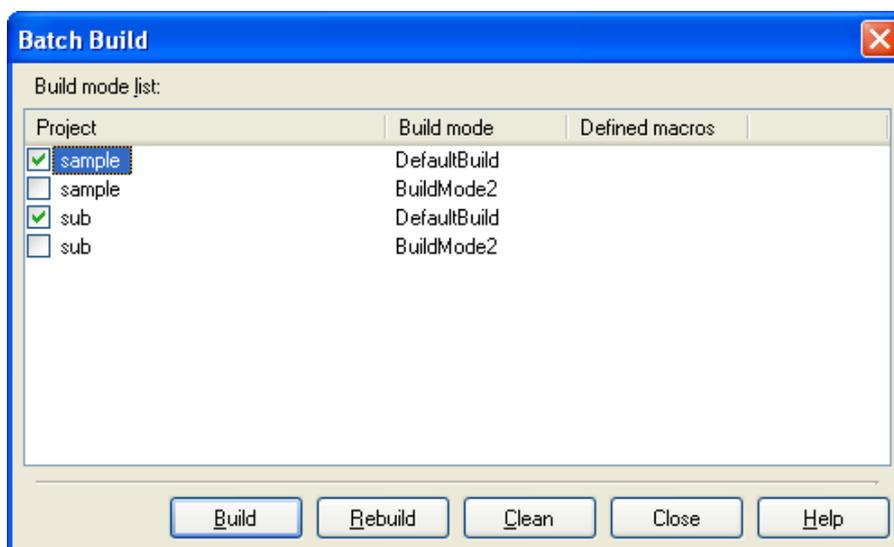
In the dialog box, the list of the combinations of the names of the main project and subprojects in the currently opened project and their build modes and macro definitions ([Defined macros] is invalid when the target project is a debug-dedicated project) is displayed.

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:       "F.5.1   Run a build"
- Rebuild:     "F.5.2   Run a rebuild"
- Clean:       "F.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+ Integrated Development Environment User's Manual: Build").

### F.5.6   Stop running a build

To stop running a build, rebuild, or batch build, click the [icon] button on the toolbar.

**Remark**   See the sections below for a build, rebuild, and batch build.
- Build:         "F.5.1   Run a build"
- Rebuild:       "F.5.2   Run a rebuild"
- Batch build:   "F.5.5   Run a batch build"

### F.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 F-35.   Save As Dialog Box**
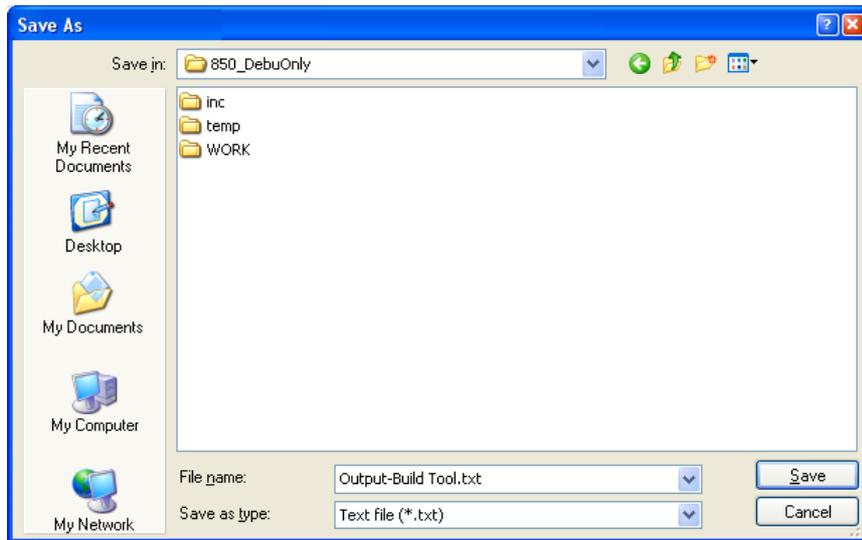


In the dialog box, specify a text file name to be saved and the location in which the file is saved, and then click the

[Save] button.

## APPENDIX　G　Python CONSOLE/Python FUNCTIONS

This section describes the Python Console and Python functions provided by CubeSuite+.

### G.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 "G.3　CubeSuite+ Python Function/Class/Property/Event" and "2.11　Execute Python Fuctions").
- When you start CubeSuite+ from the command line, you can specify and execute a script file (see "2.12　Manipulate CubeSuite+ on the Command Line").
- When loading a project file, you can run a script you have prepared in advance (see "G.2　Related File").

### G.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.

## G.3　CubeSuite+ Python Function/Class/Property/Event

This section describes CubeSuite+ Python functions, classes, and properties.
Below is a list of CubeSuite+ Python functions, classes, and properties.

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("main", 1, True)".

- 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"
```

- You should be careful when you describe a path for a folder or file as parameters.
  IronPython recognizes the backslash character (\) as a control character. For example, if a folder or file name starts with a "t", then the sequence "\t" will be recognized as a tab character. Do the following to avoid this.

**Examples 1.**　In a quoted string (""), prepend the letter "r" to make IronPython recognize the string as a path.

```
r"C:\test\test.py"
```

**2.**　Use a forward slash (/) instead of a backslash (\).

```
"C:/test/test.py"
```

A slash (/) is used in this document.

**G.3.1    CubeSuite+ Python function (for basic operation)**

Below is a list of CubeSuite+ Python functions (for basic operation).

**Table G-1.   CubeSuite+ Python Function (For Basic Operation)**

| Function Name | Function Description |
|---|---|
| ClearConsole | This function clears the string displayed on the Python console. |
| 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. |

**ClearConsole**

This function clears the string displayed on the Python console.

**[Specification format]**

```
ClearConsole()
```

**[Argument(s)]**

None

**[Return value]**

If the string was cleared successfully: True

If there was an error when clearing the string: False

**[Detailed description]**

- This function clears the string displayed on the Python console.

**[Example of use]**

```
>>>ClearConsole()
True
>>>
```

**CubeSuiteExit**

This function exits from CubeSuite+.

**[Specification format]**

```
CubeSuiteExit()
```

**[Argument(s)]**

None

**[Return value]**

None

**[Detailed description]**

- This function exits from CubeSuite+.

**Caution**     **The editing file will not be saved, even if the project file has been modified.**
                       **Use Save function to save the editing file.**

**[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.

**[Specification format]**

```
Hook(scriptFile)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| scriptFile | 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 in the script file.
  There is no problem even if functions other than a hook or callback function are declared.
  The hook or the callback function is registered when the script file is ended.
- If Hook functions are declared, they 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 |
|---|---|
| BeforeBuild | Before build |
| 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
- If Callback functions are declared, they are called after CubeSuite+ events occur.
- 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 changeed |
| 30 | Before build |
| 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
```

**Cautions 1.     Do not use the following functions in the callback function.**

**debugger.Reset function**

**debugger.Run function**

**debugger.Breakpoint function**

**2.     It is not possible to call debugger.XRunBreak.Set with different conditions in the callback function.**

**Do not make a specification like the following.**

```
def pythonConsoleCallback(Id):
  if Id = 63:
    debugger.XRunBreak.Delete()
    debugger.XRunBreak.Set(1, TimeType.Ms, True)
```

**[Example of use]**

```
>>>Hook("E:/TestFile/TestScript/testScriptFile2.py")
```

**Save**

This function saves all editing files and projects.

**[Specification format]**

```
Save()
```

**[Argument(s)]**

None

**[Return value]**

If all editing files and projects were saved successfully: True

If there was an error when saving all editing files and projects: False

**[Detailed description]**

- This function saves all editing files and projects.

**[Example of use]**

```
>>>Save()
True
>>>
```

---

**Source**

This function runs a script file.

**[Specification format]**

```
Source(scriptFile)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *scriptFile* | Specify the script file to run. |

**[Return value]**

None

**[Detailed description]**

- This function runs the script file specified by *scriptFile*.
- This function operates the same as "execfile" of IronPython.

**[Example of use]**

```
>>>Source("../../testScriptFile2.py")
>>>Source("E:/TestFile/TestScript/testScriptFile.py")
>>>
```

**G.3.2     CubeSuite+ Python function (common)**

Below is a list of CubeSuite+ Python functions (common).

**Table G-2.   CubeSuite+ Python Function (Common)**

| Function Name | Function Description |
| --- | --- |
| common.GetOutputPanel | This function displays the contents of the Output panel. |
| common.OutputPanel | This function displays the string on the Output panel. |

---

**common.GetOutputPanel**

This function displays the contents of the Output panel.

**[Specification format]**

```
common.GetOutputPanel()
```

**[Argument(s)]**

None

**[Return value]**

String displayed on the Output panel

**[Detailed description]**

　- This function displays the string displayed on the Output panel.

**[Example of use]**

```
>>> common.OutputPanel("------ Start ----- ")
True
>>> com = common.GetOutputPanel()
------ Start -----
>>> print com
------ Start -----
```

---

---

**common.OutputPanel**

This function displays the string on the Output panel.

**[Specification format]**

```
common.OutputPanel(output, messageType = MessageType.Information)
```

**[Argument(s)]**

| Argument | Description |
|----------|-------------|
| output | Specify the string displayed on the Output panel. |
| messageType | Specify the type of messages to be colored in the Output panel. The colors are in accord with the settings for the [General - Font and Color] category in the Option dialog box. |

| Type | Description |
|------|-------------|
| MessageType.Error | Error |
| MessageType.Information | Standard (default). |
| MessageType.Warning | Warning |

**[Return value]**

If the string was displayed on the Output panel successfully: True

If there was an error when displaying the string on the Output panel: False

**[Detailed description]**

- This function displays the string specified by *output* on the Output panel.

**[Example of use]**

```
>>>common.OutputPanel("An error occured.", MessageType.Error)
True
>>>
```

**G.3.3    CubeSuite+ Python function (for project)**

Below is a list of CubeSuite+ Python functions (for a project).

**Table G-3.    CubeSuite+ Python Function (For Project)**

| Function Name | Function Description |
|---|---|
| project.Change | This function changes the active project. |
| project.Close | This function closes a project. |
| project.Create | This function creates a new project. |
| project.File.Add | This function adds a file to the active project. |
| project.File.Exists | This function confirms whether the file exists in the active project. |
| project.File.Information | This function displays the list of the files registered in the active project. |
| project.File.Remove | This function removes a file from the active project. |
| project.GetDeviceNameList | This function displays the list of the device names of the microcontroller. |
| project.GetFunctionList | This function displays the list of the functions of the active project. |
| project.GetVariableList | This function displays the list of the variables of the active project. |
| project.Information | This function displays the list of project files. |
| project.Open | This function opens a project. |

**project.Change**

This function changes the active project.

**[Specification format]**

```
project.Change(projectName)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *projectName* | Specify the full path of the project or subproject to be changed. |

**[Return value]**

If the active project was changed successfully: True

If there was an error when changing the active project: False

**[Detailed description]**

- This function changes the project specified in *projectName* to the active project.
- The project file specified in *projectName* must be included the currently opened project.

**[Example of use]**

```
>>>project.Close("C:/project/sample/sub1/subproject.mtpj")
True
>>>
```

---

**project.Close**

This function closes a project.

## [Specification format]

```
project.Close(save = False)
```

## [Argument(s)]

| Argument | Description |
|---|---|
| *save* | Specify whether to save all files being edited and a project.<br>　True: Save all editing files and a project.<br>　False: Do not save all editing files and a project (default). |

## [Return value]

If the project was closed successfully: True
If there was an error when closing the project: False

## [Detailed description]

- This function closes a currently opened project.
- If *save* is set to "True", then all files being edited and a project are saved.

## [Example of use]

```
>>>project.Close()
True
>>>
```

---

**project.Create**

This function creates a new project.

**[Specification format]**

```
project.Create(fileName, micomType, deviceName, projectKind = ProjectKind.Auto, compiler =
Compiler.Auto, subProject = False, registerNaming = RegisterNaming.Structured)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *fileName* | Specify the full path of a new projrct file. |
| | If no file extension is specified, the filename is automatically supplemented. |
| | If the project to be created is a main project (subProject = False) or a subproject (sub-Project = True), the name is supplemented by ".mtpj" or ".mtsp", respectively. |
| | When the extension is other than that specified, it is replaced by the actual extension. |
| *micomType* | Specify the microcontroller type of a new projrct. |
| | The types that can be specified are shown below. |
| | <table><tr><td>Type</td><td>Description</td></tr><tr><td>MicomType.RH850</td><td>Project  for RH850</td></tr><tr><td>MicomType.RX</td><td>Project  for RX</td></tr><tr><td>MicomType.V850</td><td>Project  for V850</td></tr><tr><td>MicomType.RL78</td><td>Project  for RL78</td></tr><tr><td>MicomType.K0R</td><td>Project  for 78K0R</td></tr><tr><td>MicomType.K0</td><td>Project  for 78K0</td></tr></table> |
| *deviceName* | Specify the device name of the microcontroller of a new projrct by a string. |

| Argument | Description |
|---|---|
| *projectKind* | Specify the type of a new projrct. |

The types that can be specified are shown below.

The following is automatically specified if the microcontroller type is RH850 and "Project-Kind.Auto" is specified or *projectKind* is not specified.

   When the microcontroller is single core: ProjectKind.Application

   When the microcontroller is multi-core and main project: ProjectKind.MulticoreBoot-Loader

   When the microcontroller is multi-core and subproject: ProjectKind.MulticoreApplica-tion

| Type | Description |
|---|---|
| ProjectKind.Application | Project  for application |
| ProjectKind.Library | Project  for library |
| ProjectKind.DebugOnly | Debug-dedicated project |
| ProjectKind.Empty | Project  for empty application |
| ProjectKind.CppApplication | Project  for C++ application |
| ProjectKind.RI600V4 | Project  for RI600V4 |
| ProjectKind.RI600PX | Project  for RI600PX |
| ProjectKind.RI850V4 | Project  for RI850V4 |
| ProjectKind.RI850MP | Project  for RI850MP |
| ProjectKind.RI78V4 | Project  for RI78V4 |
| ProjectKind.MulticoreBoot-Loader | Project for boot loader for multi-core |
| ProjectKind.MulticoreApplication | Project  for application for multi-core |
| ProjectKind.Auto | The type of a projrct is selected in accord with the specification for *micomType*, *deviceName*, and *sub-Project* (default). |

| Argument | Description |
|---|---|
| *compiler* | Specify the compiler to be used. |
| | If the compiler is not specified, it is selected automatically depending on the microcontroller type. |

| Type | Description |
|---|---|
| Compiler.Auto | The compiler to be used is selected in accord with the specification for *micomType* (default). |
| Compiler.CC_RH | CC-RH<br><br>If this argument is not specified when *micomType* is set to "MicomType.RH850", CC-RH is selected automatically. |
| Compiler.CC_RX | CC-RX<br><br>If this argument is not specified when *micomType* is set to "MicomType.RX", CC-RX is selected automatically. |
| Compiler.CA850 | CA850<br><br>If this argument is not specified when *micomType* is set to "MicomType.V850" and *deviceName* is set to "V850E" or "V850ES", CA850 is selected automatically. |
| Compiler.CX | CX<br><br>If this argument is not specified when *micomType* is set to "MicomType.V850" and *deviceName* is set to "V850E2", CX is selected automatically. |
| Compiler.CA78K0R | CA78K0R<br><br>If this argument is not specified when *micomType* is set to "MicomType.K0R" or "MicomType.RL78", CA78K0R is selected automatically. |
| Compiler.CA78K0 | CA78K0<br><br>If this argument is not specified when *micomType* is set to "MicomType.K0", CA78K0 is selected automatically. |

| Argument | Description |
|---|---|
| *subProject* | Specify whether to create a main project or a subproject. |
| | False: Create a main projet (default). |
| | True: Create a subprojet. |
| *registerNaming* | When the microcontroller of the project to be created is RH850, the IOR display type is specified. |
| | This argument is ignored if it is specified when the microcontroller is not RH850.The types that can be specified are shown below. |

| Type | Description |
|---|---|
| RegisterNaming.Combined | The combined naming is selected as the IOR display type. |
| RegisterNaming.Structured | The structured naming is selected as the IOR display type (default). |

**[Return value]**

If a new project was created successfully: True

If there was an error when creating a new project: False

**[Detailed description]**

- This function creates a new project file specified by *fileName*.

  Specify the microcontroller of the project by *micomType* and *deviceName*.

  Specify the kind of the project by *projectKind*.

- If *subProject* is set to "True", then a subproject is created.

**[Example of use]**

```
>>>project.Create("c:/project/test.mtpj", MicomType.RX, "R5F52105AxFN", ProjectKind.Application)
True
>>>
```

**project.File.Add**

This function adds a file to the active project.

**[Specification format]**

```
project.File.Add(fileName, category="")
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *fileName* | Specify the full path of the file to be added to the active project. <br> When specifying multiple files, specify in the format ["file1", "file2"]. |
| *category* | Specify the category that the file is added (default: not specified). <br> When specifying multiple levels, specify in the format ["one", "two"]. |

**[Return value]**

If a file was added to the active project successfully: True

If there was an error when a file was added to the active project: False

If there was an error when any files were added to the active project when multiple files were specified for *fileName*:
False

**[Detailed description]**

- This function adds the file specified in *fileName* to the active project.
- If *category* is specified, the file is added below that category.
  If the specified category does not exist, it is created newly.

**[Example of use]**

```
>>>project.File.Add("C:/project/sample/src/test.c", "test")
True
>>>project.File.Add(["C:/project/sample/src/test1.c", "C:/project/sample/src/test2.c"],
["test", "src"])
True
```

---

**project.File.Exists**

This function confirms whether the file exists in the active project.

**[Specification format]**

```
project.File.Exists(fileName)
```

**[Argument(s)]**

| Argument | Description |
|----------|-------------|
| *fileName* | Specify the full path of the file whose existence in the active project is to be checked. |

**[Return value]**

If the specified file existed in the active project: True

If the specified file did not exist in the active project: False

**[Detailed description]**

- This function confirms whether the file specified in *fileName* exists in the active project.

**[Example of use]**

```
>>>project.File.Exists("C:/project/sample/src/test.c")
True
>>>
```

**project.File.Information**

This function displays the list of the files registered in the active project.

**[Specification format]**

```
project.File.Information()
```

**[Argument(s)]**

None

**[Return value]**

List of the files registered in the active project (in a full path)

**[Detailed description]**

- This function displays the list of the full path of the files registered in the active project.

**[Example of use]**

```
>>>project.File.Information()
C:\prj\src\file1.c
C:\prj\src\file2.c
C:\prj\src\file3.c
>>>
```

**project.File.Remove**

This function removes a file from the active project.

**[Specification format]**

```
project.File.Remove(fileName)
```

**[Argument(s)]**

| Argument | Description |
|----------|-------------|
| *fileName* | Specify the full path of the file to be removed from the active project. <br> When specifying multiple files, specify in the format ["file1", "file2"]. |

**[Return value]**

If a file was removed from the active project successfully: True

If there was an error when a file was removed from the active project: False

**[Detailed description]**

- This function removes the file specified in *fileName* from the active project.
- The file is not deleted.

**[Example of use]**

```
>>>project.File.Remove("C:/project/sample/src/test.c")
True
>>>project.File.Remove(["C:/project/sample/src/test1.c", "C:/project/sample/src/test2.c"])
True
```

**project.GetDeviceNameList**

This function displays the list of the device names of the microcontroller.

**[Specification format]**

```
project.GetDeviceNameList(micomType, nickName="")
```

**[Argument(s)]**

| Argument | Description | |
|---|---|---|
| *micomType* | Specify the microcontroller type of a new projrct. The types that can be specified are shown below. | |
| | Type | Description |
| | MicomType.RH850 | Project for RH850 |
| | MicomType.RX | Project for RX |
| | MicomType.V850 | Project for V850 |
| | MicomType.RL78 | Project for RL78 |
| | MicomType.K0R | Project for 78K0R |
| | MicomType.K0 | Project for 78K0 |
| *nickName* | Specify the nickname of the microcontroller by a string (default: not specified). Specify a character string displayed in the first layer of the [Using microcontroller] list in the Create Project dialog box that is used to create a new project. | |

**[Return value]**

List of device names

**[Detailed description]**

- This function displays the list of the device names of the microcontroller specified by *micomType*.
- When *nickName* is specified, only the names of the devices specified by nickName are displayed.

**[Example of use]**

```
>>>project.GetDeviceNameList(MicomType.RL78)
R5F10BAF
R5F10AGF
R5F10BAG
R5F10BGG
........
>>>devlist = project.GetDeviceNameList(MicomType.RL78, "RL78/F13 (ROM:128KB)")
R5F10BAG
R5F10BGG
........
>>>
```

**project.GetFunctionList**

This function displays the list of the functions of the active project.

**[Specification format]**

```
project.GetFunctionList(fileName="")
```

**[Argument(s)]**

| Argument | Description |
|----------|-------------|
| *fileName* | Specify the full path of the file that the list of the functions are displayed (default: not specified). |

**[Return value]**

List of function information (see the FunctionInfo property for detail)

**[Detailed description]**

- This function displays the list of the functions of the active project shown by the following format.

```
function-name return-value-type start-address end-address file-name
```

- When *fileName* is specified, only the functions included in the specified file are displayed.
- When *fileName* is not specified, then all the functions will be displayed.

**[Example of use]**

```
>>>project.GetFunctionList()
func1 int 0x00200 0x00224 C:\project\src\test1.c
func2 int 0x00225 0x002ff C:\project\src\test2.c
>>>project.GetFunctionList("C:/project/src/test1.c")
func1 int 0x00200 0x00224 C:\project\src\test1.c
>>>
```

**project.GetVariableList**

This function displays the list of the variables of the active project.

**[Specification format]**

```
project.GetVariableList(fileName="")
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *fileName* | Specify the full path of the file that the list of the variables are displayed (default: not specified). |

**[Return value]**

List of variable information (see the VariableInfo property for detail)

**[Detailed description]**

- This function displays the list of the variables of the active project shown by the following format.

```
variable-name attribute type address size file-name
```

- When *fileName* is specified, only the variables included in the specified file are displayed.
- When *fileName* is not specified, then all the variables will be displayed.

**[Example of use]**

```
>>>project.GetVariableList()
var1 volatile int 0x000014e4 4 C:\project\src\test1.c
var2 static int 0x000014e8 4 C:\project\src\test2.c
>>>project.GetVariableList("C:/project/src/test1.c")
var1 volatile int 0x000014e4 4 C:\project\src\test1.c
>>>
```

**project.Information**

This function displays the list of project files.

**[Specification format]**

```
project.Information()
```

**[Argument(s)]**

None

**[Return value]**

List of project file names

**[Detailed description]**

- This function displays the list of project files of the main project and subprojects included in the loaded project.

**[Example of use]**

```
>>>project.Information()
C:\project\sample\test.mtpj
C:\project\sample\sub1\sub1project.mtsp
C:\project\sample\sub2\sub2project.mtsp
>>>
```

**project.Open**

This function opens a project.

**[Specification format]**

```
project.Open(fileName, save = False)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| fileName | Specify a project file. |
| save | If another project was opened, specify whether to save any files being edited and the project when you close it.<br>  True: Save all editing files and a project.<br>  False: Do not save all editing files and a project (default). |

**[Return value]**

If the project was closed successfully: True

If there was an error when closing the project: False

**[Detailed description]**

- This function opens a project specified by *fileName*.
- If other project is opened, that project is closed.
  If *save* is set to "True", then all files being edited and a project are saved.
- If other project is not opened, the setting of *save* is ignored.

**[Example of use]**

```
>>>project.Open(r"C:/test/test.mtpj")
True
>>>
```

**G.3.4     CubeSuite+ Python function (for build tool)**

Below is a list of CubeSuite+ Python functions (for the build tool).

**Table G-4.   CubeSuite+ Python Function (For build Tool)**

| Function Name | Function Description |
|---|---|
| build.All | This function runs a build. |
| build.ChangeBuildMode | This function changes the build mode. |
| build.Clean | This function runs a clean. |
| build.File | This function runs a build of a specified file. |
| build.Update | This function updates the dependencies for the build tool. |

**build.All**

This function runs a build.

**[Specification format]**

```
build.All(rebuild = False, waitBuild = True)
```

**[Argument(s)]**

| Argument | Description |
|----------|-------------|
| rebuild | Specify whether to run a rebuild of a project.<br><br>True: Run a rebuild of a project.<br><br>False: Run a build of a project (default). |
| waitBuild | Specify whether to wait until completing a build.<br><br>True: Wait until completing a build (default).<br><br>False: Return a prompt without waiting to complete a build. |

**[Return value]**

- When *waitBuild* is set to "True"

   If a build was completed successfully: True

   If a build failed or was canceled: False


- When *waitBuild* is set to "False"

   If a build successfully started execution: True

   If a build failed to start execution: False

**[Detailed description]**

- This function runs a build of a project.

   If a subproject is added to the project, a build of the subproject is run.
- If *rebuild* is set to "True", then a rebuild of a project is run.
- If *waitBuild* is set to "False", then a prompt is returned without waiting to complete a build.
- Regardless of whether a build is successful, the build.BuildCompleted event is issued when a build completes.

**[Example of use]**

```
>>>build.All()
True
>>>
```

**build.ChangeBuildMode**

This function changes the build mode.

**[Specification format]**

```
build.ChangeBuildMode(buildmode)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *buildmode* | Specify the build mode to be changed to with a string. |

**[Return value]**

If the build mode was changed successfully: True

If there was an error when changing the build mode: False

**[Detailed description]**

- This function changes the build modes of the main project and subprojects to the build mode specified in *buildmode.*
- If *buildmode* does not exist in the project, a new build mode is created based on "DefaultBuild", and then the build mode is changed to that.

**[Example of use]**

```
>>>build.ChangeBuildMode("test_release")
True
>>>
```

---

**build.Clean**

This function runs a clean.

**[Specification format]**

```
build.Clean(all = False)
```

**[Argument(s)]**

| Argument | Description |
|----------|-------------|
| *all* | Specify whether to clean a project including subprojects.<br><br>True: Clean all project including subprojects.<br><br>False: Clean an active project (default). |

**[Return value]**

If a clean was completed successfully: True

If there was an error when running a clean: False

**[Detailed description]**

- This function runs a clean of a project (removes the files generated by a build).
- If *all* is set to "True", then a clean of the subproject is run.

**[Example of use]**

```
>>>build.Clean()
True
>>>
```

---

**build.File**

---

This function runs a build of a specified file.

**[Specification format]**

```
build.File(fileName, rebuild = False, waitBuild = True)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *fileName* | Specify a file to run a build. |
| *rebuild* | Specify whether to run a rebuild of a specified file.<br>    True: Run a rebuild of a specified file.<br>    False: Run a build of a specified file (default). |
| *waitBuild* | Specify whether to wait until completing a build.<br>    True: Wait until completing a build (default).<br>    False: Return a prompt without waiting to complete a build. |

**[Return value]**

- When *waitBuild* is set to "True"

  If a build was completed successfully: True

  If there was an error when running a build: False


- When *waitBuild* is set to "False"

  If a build successfully started execution: True

  If a build failed to start execution: False

**[Detailed description]**

- This function runs a build of a file specified by *fileName*.
- If *rebuild* is set to "True", then a rebuild of a specified file is run.
- If *waitBuild* is set to "False", then a prompt is returned without waiting to complete a build.
- The build.BuildCompleted event is issued when a build completes.

**[Example of use]**

```
>>>build.File("C:/test/test.c")
True
>>>
```

---

**build.Update**

This function updates the dependencies for the build tool.

**[Specification format]**

```
build.Update()
```

**[Argument(s)]**

None

**[Return value]**

None

**[Detailed description]**

- This function updates the dependencies of the files during build.

**[Example of use]**

```
>>>build.Update()
>>>
```

### G.3.5    CubeSuite+ Python function (for debug tool)

Below is a list of CubeSuite+ Python functions (for the debug tool).

**Table G-5.   CubeSuite+ Python Function (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. |
| 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<br>debugger.Ie.SetValue | This function sets or refers to 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<br>debugger.Jump.Address | This function displays each panel. |

| Function Name | Function Description |
|---|---|
| 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 refers to the memory. |
| debugger.Memory.ReadRange | This function refers to the specified number of locations in memory. |
| debugger.Memory.Write | This function writes to the memory. |
| debugger.Memory.WriteRange | This function writes multiple data to the memory. |
| debugger.Next | This function performs procedure step execution. |
| debugger.Register.GetValue | This function refers to 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.Timer.Clear | This function clears the result measured by a conditional timer. |
| debugger.Timer.Delete | This function deletes a conditional timer. |
| debugger.Timer.Disable | This function disables a conditional timer. |
| debugger.Timer.Enable | This function enables a conditional timer. |
| debugger.Timer.Get | This function references the result measured by a conditional timer. |
| debugger.Timer.Information | This function displays conditional timer information. |
| debugger.Timer.Set | This function sets the conditional timer. |
| debugger.Trace.Clear | This function clears the trace memory. |
| debugger.Trace.Delete | This function deletes a conditional trace. |
| debugger.Trace.Disable | This function disables a conditional trace. |
| debugger.Trace.Enable | This function enables a conditional trace. |
| debugger.Trace.Get | This function dumps the trace data. |
| debugger.Trace.Information | This function displays conditional trace information. |
| debugger.Trace.Set | This function sets conditional trace information. |
| 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. |

| Function Name | Function Description |
|---|---|
| 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 refers to 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. |

---

**debugger.Address**

This function evaluates an address expression.

**[Specification format]**

```
debugger.Address(expression)
```

**[Argument(s)]**

| Argument | Description |
|----------|-------------|
| *expression* | Specify an address expression. |

**[Return value]**

Converted address (numerical value)

**[Detailed description]**

- This function converts the address expression specified by *expression* into the address.

    **Caution**    **If a script is specified to execute in the CubeSuite+.exe startup options, then the symbol conversion function will not be available until the debugging tool is connected. In other words, this function cannot be used, so execute it after connection.**

**[Example of use]**

```
>>>debugger.Address("main")
0x4088
>>>debugger.Address("main + 1")
0x4089
>>>
```

---

---

**debugger.Assemble.Disassemble**

This function performs disassembly.

**[Specification format]**

```
debugger.Assemble.Disassemble(address, number = 1, code = True)
```

**[Argument(s)]**

| Argument | Description |
|----------|-------------|
| address | Specify the address at which to start disassembly. |
| number | Specify the number of lines to display (default: 1). |
| code | Specify whether to display instruction codes.<br>    True: Display instruction codes (default).<br>    False: Do not display instruction codes. |

**[Return value]**

List of result of disassembly (see the DisassembleInfo property for detail)

**[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.
- If "." is specified in *address*, then it is interpreted as the address following the last address disassembled.

**[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 |
|----------|-------------|
| address | Specify the address at which to start assembly. |
| code | Specify the string to assemble. |

**[Return value]**

If line assembly was performed successfully: True

If there was an error when performing line assembly: False

**[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")
True
>>>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 |
|---|---|
| *breakNumber* | Specify the break event number to delete. |

**[Return value]**

If a break point was deleted successfully: True

If there was an error when deleting a break point: False

**[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)
True
>>>debugger.Breakpoint.Disable(1)
True
>>>debugger.Breakpoint.Delete(1)
True
>>>
```

---

**debugger.Breakpoint.Disable**

This function disables a break point setting.

**[Specification format]**

```
debugger.Breakpoint.Disable(breakNumber = "")
```

**[Argument(s)]**

| Argument | Description |
|----------|-------------|
| *breakNumber* | Specify the break event number to disable. |

**[Return value]**

If a break point setting was disabled successfully: True

If there was an error when disabling a break point setting: False

**[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)
True
>>>debugger.Breakpoint.Disable(1)
True
>>>debugger.Breakpoint.Delete(1)
True
>>>
```

---

---

**debugger.Breakpoint.Enable**

This function enables a break point setting.

**[Specification format]**

```
debugger.Breakpoint.Enable(breakNumber = "")
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *breakNumber* | Specify the break event number to enable. |

**[Return value]**

If a break point setting was enabled successfully: True

If there was an error when enabling a break point setting: False

**[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)
True
>>>debugger.Breakpoint.Disable(1)
True
>>>debugger.Breakpoint.Delete(1)
True
>>>
```

---

| debugger.Breakpoint.Information |
| --- |

This function displays break point information.

**[Specification format]**

```
debugger.Breakpoint.Information()
```

**[Argument(s)]**

None

**[Return value]**

List of break point information (see the BreakpointInfo property for detail)

**[Detailed description]**

- This function displays the break point settings in the following format.
  *break-name* is "PythonBreak*xxxx*" (*xxxx*: 4-digit number).

```
break-event-number break-name state address-location
```

**[Example of use]**

```
>>>debugger.Breakpoint.Information()
   1 PythonBreak0001 Enable  0x000002dc
   2 Break0001 Enable  test1.c#_sub1
   3 PythonBreak0002 Enable  0x000002ec
   4 Break0002 Enable  test1.c#_sub1+10
>>>
```

---

### debugger.Breakpoint.Set

This function configures a break point.

**[Specification format]**

```
debugger.Breakpoint.Set(BreakCondition)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *BreakCondition* | Specify a break condition.<br>See the BreakCondition property for details about creating break conditions. |

**[Return value]**

Set break event number (numerical value)

**[Detailed description]**

- This function sets a break point according to the specifications in *BreakCondition*.

**[Example of use]**

```
>>>Condition = BreakCondition()
>>>Condition.Address = "main"
>>>breakNumber = debugger.Breakpoint.Set(Condition)
1
>>>print breakNumber
1
>>>debugger.Breakpoint.Information()
   1 PythonBreak0001 Enable  0x000002dc
```

**debugger.Connect**

This function connects to the debug tool.

**[Specification format]**

```
debugger.Connect()
```

**[Argument(s)]**

None

**[Return value]**

If the debug tool was connected successfully: True

If there was an error when connecting to the debug tool: False

**[Detailed description]**

- This function connects to the debug tool.

**[Example of use]**

```
>>>debugger.Connect()
True
>>>
```

---

**debugger.DebugTool.Change**

This function changes the debug tool.

**[Specification format]**

```
debugger.DebugTool.Change(debugTool)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *debugTool* | Specify the debug tool to change.<br>The debug tools that can be specified are shown below.<table><tr><th>Type</th><th>Description</th></tr><tr><td>DebugTool.Simulator</td><td>Simulator</td></tr><tr><td>DebugTool.Minicube</td><td>MINICUBE</td></tr><tr><td>DebugTool.Minicube2</td><td>MINICUBE2 (Serial connect)</td></tr><tr><td>DebugTool.Minicube2Jtag</td><td>MINICUBE2 (JTAG connect)</td></tr><tr><td>DebugTool.Iecube</td><td>IECUBE</td></tr><tr><td>DebugTool.Iecube2</td><td>IECUBE2</td></tr><tr><td>DebugTool.E1Jtag</td><td>E1 (JTAG connect)</td></tr><tr><td>DebugTool.E1Serial</td><td>E1 (Serial connect)</td></tr><tr><td>DebugTool.E20Jtag</td><td>E20 (JTAG connect)</td></tr><tr><td>DebugTool.E20Serial</td><td>E20 (Serial connect)</td></tr><tr><td>DebugTool.EZ_Emulator</td><td>EZ Emulator</td></tr></table> |

**[Return value]**

If the debug tool was changed successfully: True
If there was an error when changing the debug tool: False

**[Detailed description]**

- This function changes the debug tool to the one 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.

  **Caution**　　**It is possible to specify non-selectable emulators. Only specify emulators that can be selected in CubeSuite+' s debugging tool.**

---

**[Example of use]**

```
>>>debugger.DebugTool.Change(DebugTool.Simulator)
True
>>>
```

**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 |
|------|-------------|
| Simulator | Simulator |
| Minicube | MINICUBE |
| Minicube2 | MINICUBE2 (Serial connect) |
| Minicube2Jtag | MINICUBE2 (JTAG connect) |
| Iecube | IECUBE |
| Iecube2 | IECUBE2 |
| E1Jtag | E1 (JTAG connect) |
| E1Serial | E1 (Serial connect) |
| E20Jtag | E20 (JTAG connect) |
| E20Serial | E20 (Serial connect) |
| EZ_Emulator | EZ Emulator |

**[Detailed description]**

- This function displays information about the debug tool.

**[Example of use]**

```
>>>debugType = debugger.DebugTool.GetType()
Minicube2
>>>if debugType != DebugTool.Simulator:
... debugger.DebugTool.Change(DebugTool.Simulator)
...
>>>
```

**debugger.Disconnect**

This function disconnects from the debug tool.

**[Specification format]**

```
debugger.Disconnect()
```

**[Argument(s)]**

None

**[Return value]**

If the debug tool was disconnected successfully: True

If there was an error when disconnecting from the debug tool: False

**[Detailed description]**

- This function disconnects from the debug tool.

**[Example of use]**

```
>>>debugger.Disconnect()
True
>>>
```

---

**debugger.Download.Binary**

This function downloads a binary file.

**[Specification format]**

```
debugger.Download.Binary(fileName, address, append = False, flashErase = False)
```

**[Argument(s)]**

| Argument | Description |
|----------|-------------|
| fileName | Specify a download file. |
| address | Specify a download start address. |
| append | Specify whether to make an additional download.<br>　True: Perform additional download.<br>　False: Perform overwrite download (default). |
| flashErase | Specify whether to initialize a flash memory before download.<br>　True: Initialize a flash memory before download.<br>　False: Do not initialize a flash memory before download (default). |

**Caution**　**If two or more parameters are specified, then three parameters must be specified.**
**It is not possible to specify only *fileName* and *address*.**

**[Return value]**

If a binary file was downloaded successfully: True
If there was an error when downloading a binary file: False

**[Detailed description]**

- This function downloads data in binary format.

**[Example of use]**

```
>>>debugger.Download.Binary("C:/test/testModule.bin",  0x1000, False)
True
>>>debugger.Download.Binary("C:/test/testModule2.bin", 0x2000, True)
False
>>>
```

---

**debugger.Download.Binary64Kb**

This function downloads a binary file in within-64 KB format.

**[Specification format]**

```
debugger.Download.Binary64Kb(fileName, address, append = False, flashErase = False)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| fileName | Specify a download file. |
| address | Specify a download start address. |
| append | Specify whether to make an additional download.<br>    True: Perform additional download.<br>    False: Perform overwrite download (default). |
| flashErase | Specify whether to initialize a flash memory before download.<br>    True: Initialize a flash memory before download.<br>    False: Do not initialize a flash memory before download (default). |

**Caution**     **If two or more parameters are specified, then three parameters must be specified.**
                **It is not possible to specify only *fileName* and *address*.**

**[Return value]**

If a binary file was downloaded successfully: True
If there was an error when downloading a binary file: False

**[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)
True
>>>debugger.Download.Binary64Kb("C:/test/testModule2.bin", 0x2000, True)
False
>>>
```

---

**debugger.Download.BinaryBank**

This function downloads a binary file in memory bank format.

**[Specification format]**

```
debugger.Download.BinaryBank(fileName, address, append = False, flashErase = False)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| fileName | Specify a download file. |
| address | Specify a download start address. |
| append | Specify whether to make an additional download. <br> True: Perform additional download. <br> False: Perform overwrite download (default). |
| flashErase | Specify whether to initialize a flash memory before download. <br> True: Initialize a flash memory before download. <br> False: Do not initialize a flash memory before download (default). |

**Caution**    **If two or more parameters are specified, then three parameters must be specified.**
                **It is not possible to specify only *fileName* and *address*.**

**[Return value]**

If a binary file was downloaded successfully: True
If there was an error when downloading a binary file: False

**[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)
True
>>>debugger.Download.BinaryBank("C:/test/testModule2.bin", 0x2000, True)
False
>>>
```

---

---

**debugger.Download.Coverage**

This function downloads coverage data. [IECUBE][IECUBE2][Simulator]

**[Specification format]**

```
debugger.Download.Coverage(fileName)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *fileName* | Specify a coverage data file. |

**[Return value]**

If a binary file was downloaded successfully: True

If there was an error when downloading a binary file: False

**[Detailed description]**

- This function downloads coverage data.

**[Example of use]**

```
>>>debugger.Download.Coverage("C:/test/testModule.csrcv")
True
>>>
```

---

**debugger.Download.Hex**

This function downloads a hex file.

**[Specification format]**

```
debugger.Download.Hex(fileName, offset = 0, append = False, flashErase = False)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| fileName | Specify a download file. |
| offset | Specify an offset (default: 0). |
| append | Specify whether to make an additional download.<br>　True: Perform additional download.<br>　False: Perform overwrite download (default). |
| flashErase | Specify whether to initialize a flash memory before download.<br>　True: Initialize a flash memory before download.<br>　False: Do not initialize a flash memory before download (default). |

**Caution**　**If two or more parameters are specified, then three parameters must be specified.**
**It is not possible to specify only *fileName* and *offset*.**

**[Return value]**

If a binary file was downloaded successfully: True
If there was an error when downloading a binary file: False

**[Detailed description]**

- This function downloads data in hex format.

**[Example of use]**

```
>>>debugger.Download.Hex("C:/test/testModule.hex")
True
>>>
```

---

**debugger.Download.Hex64Kb**

This function downloads a hex file in within-64 KB format.

**[Specification format]**

```
debugger.Download.Hex64Kb(fileName, offset = 0, append = False, flashErase = False)
```

**[Argument(s)]**

| Argument | Description |
|----------|-------------|
| fileName | Specify a download file. |
| offset | Specify an offset (default: 0). |
| append | Specify whether to make an additional download.<br>True: Perform additional download.<br>False: Perform overwrite download (default). |
| flashErase | Specify whether to initialize a flash memory before download.<br>True: Initialize a flash memory before download.<br>False: Do not initialize a flash memory before download (default). |

**Caution**  **If two or more parameters are specified, then three parameters must be specified.**
**It is not possible to specify only *fileName* and *offset*.**

**[Return value]**

If a binary file was downloaded successfully: True
If there was an error when downloading a binary file: False

**[Detailed description]**

 - When using the memory bank, this function downloads hex files in within-64 KB format.

**[Example of use]**

```
>>>debugger.Download.Hex64Kb("C:/test/testModule.hex")
True
>>>
```

---

**debugger.Download.HexBank**

This function downloads a hex file in memory bank format.

**[Specification format]**

```
debugger.Download.HexBank(fileName, offset = 0, append = False, flashErase = False)
```

**[Argument(s)]**

| Argument | Description |
|----------|-------------|
| fileName | Specify a download file. |
| offset | Specify an offset (default: 0). |
| append | Specify whether to make an additional download.<br>True: Perform additional download.<br>False: Perform overwrite download (default). |
| flashErase | Specify whether to initialize a flash memory before download.<br>True: Initialize a flash memory before download.<br>False: Do not initialize a flash memory before download (default). |

> **Caution** **If two or more parameters are specified, then three parameters must be specified.**
> **It is not possible to specify only *fileName* and *offset*.**

**[Return value]**

If a binary file was downloaded successfully: True
If there was an error when downloading a binary file: False

**[Detailed description]**

- When using the memory bank, this function downloads hex files in memory-bank format.

**[Example of use]**

```
>>>debugger.Download.HexBank("C:/test/testModule.hex")
True
>>>debugger.Download.HexBank("C:/test/testModule2.hex", 0x1000, True)
False
>>>
```

---

**debugger.Download.HexIdTag**

This function downloads a hex file with ID tag.

**[Specification format]**

```
debugger.Download.HexIdTag(fileName, offset = 0, append = False, flashErase = False)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *fileName* | Specify a download file. |
| *offset* | Specify an offset (default: 0). |
| *append* | Specify whether to make an additional download.<br>True: Perform additional download.<br>False: Perform overwrite download (default). |
| *flashErase* | Specify whether to initialize a flash memory before download.<br>True: Initialize a flash memory before download.<br>False: Do not initialize a flash memory before download (default). |

**Caution** **If two or more parameters are specified, then three parameters must be specified.**
**It is not possible to specify only *fileName* and *offset*.**

**[Return value]**

If a binary file was downloaded successfully: True
If there was an error when downloading a binary file: False

**[Detailed description]**

- This function downloads a hex file with ID tag.

**[Example of use]**

```
>>>debugger.Download.HexIdTag("C:/test/testModule.hex")
True
>>>debugger.Download.HexIdTag("C:/test/testModule2.hex", 0x1000, True)
False
>>>
```

**debugger.Download.Information**

This function displays download information.

**[Specification format]**

```
debugger.Download.Information()
```

**[Argument(s)]**

None

**[Return value]**

List of download information (see the DownloadInfo property for detail)

**[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, flashErase = False)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *fileName* | Specify a download file. |
| *downloadOption* | Specify an option.<br>The options that can be specified are shown below.<br><br>| Type | Description |<br>|---|---|<br>| DownloadOption.NoSymbol | Do not load symbol information. |<br>| DownloadOption.SymbolOnly | Only load symbol information. |<br>| DownloadOption.Both | Load both symbol information and object information (default). | |
| *append* | Specify whether to make an additional download.<br>    True: Perform additional download.<br>    False: Perform overwrite download (default). |
| *flashErase* | Specify whether to initialize a flash memory before download.<br>    True: Initialize a flash memory before download.<br>    False: Do not initialize a flash memory before download (default). |

**[Return value]**

If a binary file was downloaded successfully: True
If there was an error when downloading a binary file: False

**[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("C:/test/testModule.lmf")
True
>>>debugger.Download.LoadModule("C:/test/testModule2.lmf", DownloadOption.SymbolOnly, True)
False
>>>
```

---

---

debugger.Erase

---

This function erases the flash memory.

**[Specification format]**

```
debugger.Erase(eraseOption = EraseOption.Code)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *eraseOption* | Specify an option.<br>The options that can be specified are shown below.<table><tr><th>Type</th><th>Description</th></tr><tr><td>EraseOption.Code</td><td>Erase the code flash memory (default).</td></tr><tr><td>EraseOption.Data</td><td>Erase the data flash memory.</td></tr><tr><td>EraseOption.External</td><td>Erase the flash memory in external space.</td></tr></table> |

**Caution** **IECUBE, IECUBE2, and the simulator do not have functionality to delete code flash memory.  For this reason, if you are using IECUBE, IECUBE2, or the simulator, you cannot omit *eraseOption*, or specify "EraseOption.Code".**

**[Return value]**

If the flash memory was erased successfully: True
If there was an error when erasing the flash memory: False

**[Detailed description]**

- This function erases the flash memory, specified by *eraseOption*.

**[Example of use]**

```
>>>debugger.Erase()
True
>>>debugger.Erase(EraseOption.External)
False
>>>
```

**debugger.GetBreakStatus**

This function displays a break condition.

**[Specification format]**

```
debugger.GetBreakStatus()
```

**[Argument(s)]**

None

**[Return value]**

Break-trigger string (See [Detailed description])

**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.

  During execution, this will be "None".

| Break-trigger String | Description | 78K0 | | | RL78,78K0R | | | R8C | V850 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Iecube | Minicube2 **Note 1** | Simulator | Iecube | Minicube2 **Note 1** | Simulator | EZ_Emulator | Iecube | Minicube **Note 2** | Minicube2 **Note 1** | Simulator |
| None | No break | o | o | - | o | o | - | o | o | o | o | - |
| Manual | Forced break | o | o | o | o | o | o | o | o | o | o | o |
| Event | Break due to event | o | o | o | o | o | o | o | o | o | o | o |
| Software | Software break | o | o | - | o | o | - | o | o | o | o | - |
| TraceFull | Break due to trace full | o | - | o | o | - | o | - | o | - | - | o |
| TraceDelay | Break due to trace delay | o | - | - | o | - | - | - | - | - | - | - |
| NonMap | Access to non-mapped area | o | - | o | o | - | o | - | o | - | - | o |
| WriteProtect | Write to write-protected area | o | - | o | o | - | o | - | o | - | - | o |
| ReadProtect | Read from read-protected area | o | - | - | - | - | - | - | - | - | - | - |
| SfrIllegal | Illegal SFR access | o | - | - | - | - | - | - | - | - | - | - |
| SfrReadProtect | Read from non-readable SFR | o | - | - | o | - | - | - | - | - | - | - |

| Break-trigger String | Description | 78K0 | | | RL78,78K0R | | | R8C | V850 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Iecube | Minicube2 Note 1 | Simulator | Iecube | Minicube2 Note 1 | Simulator | EZ_Emulator | Iecube | Minicube Note 2 | Minicube2 Note 1 | Simulator |
| SfrWriteProtect | Write to non-writable SFR | o | - | - | o | - | - | - | - | - | - | - |
| IorIllegal | Illegal access to peripheral I/O register (with address) | - | - | - | - | - | - | - | o | - | - | - |
| StackOverflow | Break due to stack overflow | o | - | - | o | - | - | - | - | - | - | - |
| StackUnderflow | Break due to stack underflow | o | - | - | o | - | - | - | - | - | - | - |
| UninitializeStackPointer | Break due to uninitialized stack pointer | o | - | - | o | - | - | - | - | - | - | - |
| UninitializeMemoryRead | Read uninitialized memory | o | - | - | o | - | - | - | - | - | - | - |
| TimerOver | Execution timeout detected | o | - | - | o | - | - | - | o | - | - | - |
| UnspecifiedIllegal | Illegal operation in user program relating to peripheral chip features | o | - | - | o | - | - | - | - | - | - | - |
| ImsIxsIllegal | Break due to illegal write to IMS/IXS register | o | - | - | - | - | - | - | - | - | - | - |
| BeforeExecution | Pre-execution break | o | - | - | o | - | - | - | - | - | - | - |
| SecurityProtect | Accessed security-protected region | - | - | - | - | - | - | - | - | - | - | - |
| FlashMacroService | Flash macro service active | - | - | - | - | - | - | - | - | o | o | - |
| RetryOver | Number of retries exceeded limit | o | - | - | - | - | - | - | - | - | - | - |
| FlashIllegal | Illegal Flash break | o | - | - | o | - | - | - | - | - | - | - |
| Peripheral | Break from peripheral | o | - | - | o | - | - | - | - | - | - | - |
| WordMissAlignAccess | Word access to odd address | - | - | - | o | - | o | - | - | - | - | - |
| Temporary | Temporary break | o | o | o | o | o | o | o | o | o | o | o |
| Escape | Escape break | - | - | - | - | - | - | - | o | o | o | - |
| Fetch | Fetched from guard area or area where fetches are prohibited | o | - | - | o | - | - | - | - | - | - | - |

| Break-trigger String | Description | 78K0 | | | RL78,78K0R | | | R8C | V850 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Iecube | Minicube2[Note 1] | Simulator | Iecube | Minicube2[Note 1] | Simulator | EZ_Emulator | Iecube | Minicube[Note 2] | Minicube2[Note 1] | Simulator |
| IRamWriteProtect | Wrote to IRAM guard area (with address)[Note 3] | - | - | - | - | - | - | - | o | - | - | - |
| IllegalOpcodeTrap | Break due to illegal instruction exception | - | - | - | - | - | - | - | o | Δ [Note 6] | - | - |
| Step | Step execution break[Note 4] | o | o | o | o | o | o | - | - | - | - | o |
| FetchGuard | Fetch guard break[Note 4] | o | - | - | o | - | - | - | - | - | - | - |
| TraceStop | Trace stop[Note 4] | o | - | - | o | - | - | - | - | - | - | - |
| ExecutionFails | Execution failed[Note 5] | o | o | - | o | o | - | o | o | o | o | - |

**Notes 1.** Applies to all of the following: MINICUBE2, E1Serial, E20Serial, and EZ_Emulator.

    **2.** Applies to all of the following: MINICUBE, E1Jtag, E20Jtag, and MINICUBE2Jtag.

    **3.** Performed a verification check on the IRAM guard area during break, and the value was overwritten (if this affects multiple addresses, only the first address is shown).

    **4.** This is only a break cause during trace.

    **5.** This is only a break cause during a break.

    **6.** Not displayed with V850-MINICUBE on V850E/ME2, etc. (same core) when a post-execution event is used.

| Break-trigger String | Description | RX | | V850E2 | | | | RH850 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | E1Jtag, E1Serial E20Jtag, E20Serial | Simulator | Iecube2 | Minicube[Note 2] | Minicube2[Note 1] | Simulator | Full-spec emulator | E1 / E20 | SIM |
| None | No break | o | - | o | o | o | - | - | - | - |
| Manual | Forced break | o | o | o | o | o | o | o | o | o |
| Event | Break due to event | o | o | o | o | o | o | o | o | o |
| Software | Software break | o | - | o | o | o | - | o | o | - |
| TraceFull | Break due to trace full | o | o | o | - | - | o | o | o | o |
| NonMap | Access to non-mapped area | - | - | - | - | - | o | - | - | o |
| WriteProtect | Write to write-protected area | - | - | - | - | - | o | - | - | o |
| TimerOver | Execution timeout detected | - | - | o | o | - | - | - | - | - |

| Break-trigger String | Description | RX | | V850E2 | | | | RH850 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | E1Jtag, E1Serial E20Jtag, E20Serial | Simulator | Iecube2 | Minicube**Note 2** | Minicube2**Note 1** | Simulator | Full-spec emulator | E1 / E20 | SIM |
| FlashMacroService | Flash macro service active | - | - | o | o | o | - | - | - | - |
| Temporary | Temporary break | o | o | o | o | o | o | o | o | o |
| IllegalOpcodeTrap | Break due to illegal instruction exception | - | - | o | o | - | - | - | - | - |
| Step | Step execution break**Note 3** | o | - | - | - | - | o | o | o | o |
| ExecutionFails | Execution failed**Note 4** | o | - | o | o | o | - | - | - | - |
| WaitInstruction | Break caused by executing WAIT instruction | - | o | - | - | - | - | - | - | - |
| UndefinedInstructionEx-ception | Break caused by undefined instruc-tion exception | - | o | - | - | - | - | - | - | - |
| PrivilegeInstructionEx-ception | Break caused by privileged instruc-tion exception | - | o | - | - | - | - | - | - | - |
| AccessException | Break caused by access exception | - | o | - | - | - | - | - | - | - |
| FloatingPointException | Break caused by floating point exception | - | o | - | - | - | - | - | - | - |
| InterruptException | Break caused by interrupt | - | o | - | - | - | - | - | - | - |
| IntInstructionException | Break caused by INT instruction exception | - | o | - | - | - | - | - | - | - |
| BrkInstructionException | Break caused by BRK instruction exception | - | o | - | - | - | - | - | - | - |
| IOFunctionSimulation-Break | Break caused by peripheral function simulation | - | o | - | - | - | - | - | - | - |
| IllegalMemoryAccess-Break | Break caused by illegal memory access | - | o | - | - | - | - | - | - | - |
| StreamIoError | Break caused by stream I/O error | - | o | - | - | - | - | - | - | - |
| CoverageMemoryAlloca-tionFailure | Failed to allocate coverage memory | - | o | - | - | - | - | - | - | - |
| TraceMemoryAllocation-Failure | Failed to allocate trace memory | - | o | - | - | - | - | - | - | - |
| StepCountOver | Step count over | - | - | - | - | - | - | o | o | o |
| DebuggingInformation-AcquisitionFailure | Failed to acquire debugging infor-mation | - | - | - | - | - | - | o | o | o |

**Notes 1.**     Applies to all of the following: MINICUBE2, E1Serial, and E20Serial.

      **2.**     Applies to all of the following: MINICUBE, E1Jtag, E20Jtag, and MINICUBE2Jtag.

      **3.**     This is only a break cause during trace.

      **4.**     This is only a break cause during a break.

**[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 (string)

| CPU Status | Description |
| --- | --- |
| Hold | In bus hold |
| HoldStopIdle | Bus hold/Software STOP/Hardware STOP/IDLE mode |
| PowOff | Power not supplied to the target |
| Reset | In reset state |
| Standby | In standby mode |
| Stop | In STOP mode |
| StopIdle | Software STOP/Hardware STOP/IDLE mode |
| Wait | In wait state |
| Halt | In HALT mode |
| Sleep | In sleep 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 (string)

| IE Status | Description |
|-----------|-------------|
| Break | Break in effect |
| Coverage | Coverage running |
| Timer | Timer running |
| Tracer | Trace running |
| Step | Step executing |
| Run | User program running |
| RunOrStep | User program running or step executing |

**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. In addition, "None" will be returned before the debugging tool is connected.**

**[Detailed description]**

  - This function displays the current IE status.

**[Example of use]**

```
>>>debugger.GetIeStatus()
Run
>>>
```

**debugger.GetPC**

This function displays the PC value.

**[Specification format]**

```
debugger.GetPC()
```

**[Argument(s)]**

None

**[Return value]**

PC value (numeric 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 |
|---|---|
| *goOption* | Specify an option.<br>The options that can be specified are shown below.<br><br>| Type | Description |<br>|---|---|<br>| GoOption.IgnoreBreak | Execute ignoring breakpoints. |<br>| GoOption.WaitBreak | Wait until program stops. |<br>| 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 or refers to the IE register or DCU register.

**[Specification format]**

```
debugger.Ie.GetValue(ieType, address)
debugger.Ie.SetValue(ieType, address, value)
```

**[Argument(s)]**

| Argument | Description | |
|---|---|---|
| *ieType* | Specify a register.<br>The registers that can be specified are shown below. | |
| | Type | Description |
| | IeType.Reg | IE register [78K0] [RL78] [78K0R] [IECUBE [V850]] [IECUBE2 [V850]] |
| | IeType.Dcu | DCU register [IECUBE [V850]] |
| *address* | Specify the address to reference/set. | |
| *value* | Specify the setting value. | |

**[Return value]**

debugger.Ie.GetValue is the register value (numeric value)

debugger.Ie.SetValue is True if the setting was completed successfully, or False if there was an error when setting the register.

**[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)
True
>>>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"
...
True
OK
>>>
```

**debugger.IsRunning**

This function checks the execution status of the user program.

**[Specification format]**

```
debugger.IsRunning()
```

**[Argument(s)]**

None

**[Return value]**

If the user program is running: True

If the user program is not running: False

**[Detailed description]**

- This function checks the execution status of the user program.

**[Example of use]**

```
>>>if debugger.IsRunning() == True :
...  print "OK"
...
True
OK
>>>
```

---

**debugger.Jump.File**

**debugger.Jump.Address**

---

This function displays each panel.

**[Specification format]**

```
debugger.Jump.File(fileName, lineNumber = 1)
debugger.Jump.Address(jumpType, adddress = 0)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *fileName* | Specify the name of the file to display. |
| *lineNumber* | Specify the line to display (default: 1). |
| *jumpType* | Specify the type of panel to display. The panel types that can be specified are shown below. <table><tr><td>Type</td><td>Description</td></tr><tr><td>JumpType.Source</td><td>Editor panel</td></tr><tr><td>JumpType.Assemble</td><td>Disassemble panel</td></tr><tr><td>JumpType.Memory</td><td>Memory panel</td></tr></table> |
| *address* | 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("C:/test/testJump.c")
>>>debugger.Jump.File("C:/test/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]**

If the memory map was cleared successfully: True

If there was an error when clearing the memory map: False

**[Detailed description]**

- This function clears the mapping settings.

**[Example of use]**

```
>>>debugger.Map.Clear()
True
>>>
```

---

**debugger.Map.Information**

This function displays map information.

**[Specification format]**

```
debugger.Map.Information()
```

**[Argument(s)]**

None

**[Return value]**

List of map information (see the MapInfo class for detail)

**[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 0x03FFEFFF 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 |
|---|---|
| mapType | Specify a memory type. The memory types that can be specified are shown below. |
| | | Type | Description |
| | |---|---|
| | | MapType.EmulationRom | Emulation ROM area |
| | | MapType.EmulationRam | Emulation RAM area |
| | | MapType.Target | Target memory area |
| | | MapType.TargetRom | Target ROM area |
| | | MapType.Stack | Stack area |
| | | MapType.Protect | I/O protect area |
| address1 | Specify a map start address. |
| address2 | Specify a map end address. |
| accessSize | Specify an access size (bit) (default: 8). For V850, specify either 8, 16, or 32. For 78K0R [IECUBE], specify either 8 or 16. |
| cs | Specify the chip select (default: not specified). When mapping emulation memory (alternative ROM/RAM) in the IECUBE [V850E1], 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. *If chip select is specified, then accessSize cannot be omitted.* |

**[Return value]**

If memory mapping was configured successfully: True
If there was an error when configuring memory mapping: False

**[Detailed description]**

- This function configures memory mapping with the memory type specified by *mapType*.

---

**[Example of use]**

```
>>>debugger.Map.Set(MapType.EmulationRom, 0x100000, 0x10ffff)
True
>>>
```

---

**debugger.Memory.Copy**

This function copies the memory.

## [Specification format]

```
debugger.Memory.Copy(address1, address2, address3)
```

## [Argument(s)]

| Argument | Description |
|----------|-------------|
| *address1* | Specify the start address to copy from. |
| *address2* | Specify the end address to copy from. |
| *address3* | Specify the address to copy to. |

## [Return value]

If the memory was copied successfully: True

If there was an error when copying the memory: False

## [Detailed description]

- This function copies the memory from *address1* to *address2* into *address3*.

## [Example of use]

```
>>>debugger.Memory.Copy(0x1000, 0x2000, 0x3000)
True
>>>
```

---

debugger.Memory.Fill

This function fills the memory.

**[Specification format]**

```
debugger.Memory.Fill(address1, address2, value, memoryOption = MemoryOption.Byte)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| address1 | Specify the start address to fill. |
| address2 | Specify the end address to fill to. |
| value | Specify the fill value. |
| memoryOption | 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) [RH850,RX,V850] |
| MemoryOption.Word | Word unit (RL78,78K: 16 bits, RH850,RX,V850: 32 bits) |

**[Return value]**

If the memory was filled successfully: True

If there was an error when filling the memory: False

**[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)
True
>>>debugger.Memory.Fill(0x2000, 0x3000, 0x0A, MemoryOption.Word)
False
>>>
```

---

**debugger.Memory.Read**

This function refers to the memory.

## [Specification format]

```
debugger.Memory.Read(address, memoryOption = MemoryOption.Byte)
```

## [Argument(s)]

| Argument | Description | | |
|----------|-------------|---|---|
| `address` | Specify the address to reference. | | |
| `memoryOption` | Specify the display unit. <br> 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) [RH850,RX,V850] |
| | MemoryOption.Word | | Word unit (RL78,78K: 16 bits, RH850,RX,V850: 32 bits) |

## [Return value]

Referenced memory value (numeric 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.ReadRange**

This function refers to the specified number of locations in memory.

**[Specification format]**

```
debugger.Memory.ReadRange(address, count, memoryOption = MemoryOption.Byte)
```

**[Argument(s)]**

| Argument | Description | | |
|---|---|---|---|
| address | Specify the start address to reference. | | |
| count | Specify the number of locations in memory for reference. | | |
| memoryOption | Specify the display unit.<br>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) [RH850,RX,V850] | |
| | MemoryOption.Word | Word unit (RL78,78K: 16 bits, RH850,RX,V850: 32 bits) | |

**[Return value]**

List of referenced memory value (numeric value)

**[Detailed description]**

- This function displays, in hexadecimal notation, the number of values specified by *count* with the width in memory specified by *memoryOption* in the range from the address specified by *address*.
- In case of failure to acquire a value from memory, "?" is displayed (0x??, 0x????, and 0x???????? in the 8-, 16-, and 32-bit cases, respectively).

**[Example of use]**

```
>>>debugger.Memory.ReadRange(0x100, 3, MemoryOption.Word)
0x00000011 0x0000ff30 0x0000ff40
>>>mem = debugger.Memory.ReadRange(0x1ffffd, 5, MemoryOption.Byte)
0x23 0x43 0x32 0x?? 0x??
>>>print mem
[35, 67, 50, None, None]
```

---

---

**debugger.Memory.Write**

This function writes to the memory.

**[Specification format]**

```
debugger.Memory.Write(address, value, memoryOption = MemoryOption.Byte)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| address | Specify the address to set. |
| value | Specify the value to set. |
| memoryOption | 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) [RH850,RX,V850] |
| | MemoryOption.Word | Word unit (RL78,78K: 16 bits, RH850,RX,V850: 32 bits) |

**[Return value]**

If the memory was written to successfully: True

If there was an error when writing to the memory: False

**[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)
True
>>>debugger.Memory.Read(0x100)
0xFF
>>>debugger.Memory.Write(0x100, 0xFE, MemoryOption.HalfWord)
False
>>>
```

---

### debugger.Memory.WriteRange

This function writes multiple data to the memory.

**[Specification format]**

```
debugger.Memory.WriteRange(address, valuelist, memoryOption = MemoryOption.Byte)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| address | Specify the start address to write. |
| valuelist | Specify the list of the value to set. |
| memoryOption | Specify the unit to set.<br>The units that can be specified are shown below.<table><tr><th>Type</th><th>Description</th></tr><tr><td>MemoryOption.Byte</td><td>Byte unit (8 bits) (default)</td></tr><tr><td>MemoryOption.HalfWord</td><td>Half-word unit (16 bits) [RH850,RX,V850]</td></tr><tr><td>MemoryOption.Word</td><td>Word unit (RL78,78K: 16 bits, RH850,RX,V850: 32 bits)</td></tr></table> |

**[Return value]**

If the memory was written to successfully: True
If there was an error when writing to the memory: False

**[Detailed description]**

- This function writes, in accord with the setting of *memoryOption*, the list of values specified by *valuelist* to the address range starting at the address specified by *address*.

**[Example of use]**

```
>>> mem = [0x10, 0x20, 0x30]
>>>debugger.Memory.WriteRange(0x100, mem, MemoryOption.Byte)
True
>>>debugger.Memory.ReadRange(0x100, 3, MemoryOption.Byte)
0x10 0x20 0x30
>>>debugger.Memory.WriteRange(0x100, mem, MemoryOption.Word)
True
>>>debugger.Memory.ReadRange(0x100, 3, MemoryOption.Word)
0x00000010 0x00000020 0x00000030
```

**debugger.Next**

This function performs procedure step execution.

**[Specification format]**

```
debugger.Next(nextOption = NextOption.Source)
```

**[Argument(s)]**

| Argument | Description | |
|---|---|---|
| *nextOption* | 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.Instruction)
>>>
```

---

**debugger.Register.GetValue**

This function refers register/IO register/SFR.

**[Specification format]**

```
debugger.Register.GetValue(regName)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *regName* | Specify the register name to reference. |

**[Return value]**

Register value (numeric 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("A:RB1")
0x20
>>>debugger.Register.SetValue("pc", 0x200)
True
>>>debugger.Register.GetValue("pc")
0x200
>>>
```

---

**debugger.Register.SetValue**

This function sets the value of a register, IO register, and SFR.

**[Specification format]**

```
debugger.Register.SetValue(regName, value)
```

**[Argument(s)]**

| Argument | Description |
|----------|-------------|
| regName | Specify the register name to set. |
| value | Specify the value to set. |

**[Return value]**

If the value was set successfully: True

If there was an error when setting the value: False

**[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("A:RB1")
0x20
>>>debugger.Register.SetValue("pc", 0x200)
True
>>>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 | |
|----------|-------------|---|
| *runOption* | Specify an option.<br>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 |
|---|---|
| *stepOption* | Specify the execution unit. <br> 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.Instruction)
```

---

**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.Timer.Clear**

This function clears the result measured by a conditional timer.

**[Specification format]**

```
debugger.Timer.Clear()
```

**[Argument(s)]**

None

**[Return value]**

If the result measured by a conditional timer was cleared successfully: True

If there was an error when clearing the result measured by a conditional timer: False

**[Detailed description]**

- This function clears the result measured by a conditional timer.

**[Example of use]**

```
>>>debugger.Timer.Get()
1 Total: 2000 ns, Pass Count: 4 , Average: 500 ns, Max: 800 ns, Min: 300 ns
>>>debugger.Timer.Clear()
True
>>>debugger.Timer.Get()
1 Total: 0 ns, Pass Count: 0 , Average: 0 ns, Max: 0 ns, Min: 0 ns
>>>
```

---

**debugger.Timer.Delete**

This function deletes a conditional timer.

**[Specification format]**

```
debugger.Timer.Delete(timerNumber = "")
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *timerNumber* | Specify the timer event number to delete. |

**[Return value]**

If a timer was deleted successfully: True

If there was an error when deleting a timer: False

**[Detailed description]**

- This function deletes the timer of the timer event numer specified by *timerNumber*.
- If *timerNumber* is not specified, then timers of all timer event numbers will be deleted.

**[Example of use]**

```
>>>debugger.Timer.Delete(1)
True
>>>
```

---

**debugger.Timer.Disable**

This function disables a conditional timer.

**[Specification format]**

```
debugger.Timer.Disable(timerNumber = "")
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| `timerNumber` | Specify the timer event number to disable. |

**[Return value]**

If a timer setting was disabled successfully: True

If there was an error when disabling a timer setting: False

**[Detailed description]**

- This function disables the timer of the timer event specified by *timerNumber*.
- If *timerNumber* is not specified, then timers of all timer event numbers will be disabled.

**[Example of use]**

```
>>>debugger.Timer.Disable(1)
True
>>>
```

---

**debugger.Timer.Enable**

This function enables a conditional timer.

**[Specification format]**

```
debugger.Timer.Enable(timerNumber = "")
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| timerNumber | Specify the timer event number to enable. |

**[Return value]**

If a timer setting was enabled successfully: True

If there was an error when enabling a timer setting: False

**[Detailed description]**

- This function enables the timer of the timer event specified by *timerNumber*.
- If *timerNumber* is not specified, then timers of all timer event numbers will be enabled.

**[Example of use]**

```
>>>debugger.Timer.Enable(1)
True
>>>
```

---

---

**debugger.Timer.Get**

This function references the result measured by a conditional timer.

**[Specification format]**

```
debugger.Timer.Get()
```

**[Argument(s)]**

None

**[Return value]**

List of conditional timer information (see the TimerInfo class for detail)

**[Detailed description]**

- The result measured by a conditional timer is shown by the following format.

```
timer-event-number Total: total-execution-time ns, Pass Count: pass-count , Average:
average-execution-time ns, Max: maximum-execution-time ns, Min: minimum-execution-time ns
```

**[Example of use]**

```
>>>debugger.Timer.Get()
1 Total: 2000 ns, Pass Count: 4 , Average: 500 ns, Max: 800 ns, Min: 300 ns
>>>
```

---

---

**debugger.Timer.Information**

This function displays conditional timer information.

**[Specification format]**

```
debugger.Timer.Information()
```

**[Argument(s)]**

None

**[Return value]**

List of conditional timer event information (see the TimerEventInfo class for detail)

**[Detailed description]**

- This function displays conditional timer information displays in the following format.

```
timer-event-number timer-name state start-address - end-address
```

**[Example of use]**

```
>>>ti = debugger.Timer.Information()
1 PythonTimer0001 Enable main - sub
>>>print ti[0].Number
1
>>>print ti[0].Name
PythonTimer0001
>>>
```

---

### debugger.Timer.Set

This function sets the conditional timer.

**[Specification format]**

```
debugger.Timer.Set(TimerCondition)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *TimerCondition* | Specify a condition of a conditional timer.<br>See the TimerCondition class for creating a conditional timer. |

**[Return value]**

Set timer event number (numerical value)

**[Detailed description]**

- This function sets a conditional timer according to the contents specified with *TimerCondition*.
- The specified conditional timer is registered with the following name.
  *number* is a four-digit decimal.

```
Python Timer number
```

**[Example of use]**

```
>>>tc = TimerCondition()
>>>tc.StartAddress = "main"
>>>tc.EndAddress = "chData"
>>>tc.EndData = 0x20
>>>tc.EndTimerType = TimerType.Write
>>>ts_number = debugger.Timer.Set(tc)
1
>>>print ts_number
1
```

---

**debugger.Trace.Clear**

This function clears the trace memory.

**Remark**    This function provides the same function as debugger.XTrace.Clear.

**[Specification format]**

```
debugger.Trace.Clear()
```

**[Argument(s)]**

None

**[Return value]**

If the trace memory was cleared successfully: True
If there was an error when clearing the trace memory: False

**[Detailed description]**

- This function clears the trace memory.

**[Example of use]**

```
>>>debugger.Trace.Clear()
False
>>>
```

---

**debugger.Trace.Delete**

This function deletes a conditional trace.

**[Specification format]**

```
debugger.Trace.Delete(timerNumber = "")
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *timerNumber* | Specify the trace event number to delete. |

**[Return value]**

If a trace was deleted successfully: True

If there was an error when deleting a trace: False

**[Detailed description]**

- This function deletes the trace of the trace event numer specified by *tracenumber*.
- If *traceNumber* is not specified, then traces of all trace event numbers will be deleted.

**[Example of use]**

```
>>>debugger.Trace.Delete(1)
True
>>>
```

---

| **debugger.Trace.Disable** |
| --- |

This function disables a conditional trace.

**[Specification format]**

```
debugger.Trace.Disable(traceNumber = "")
```

**[Argument(s)]**

| Argument | Description |
| --- | --- |
| *traceNumber* | Specify the trace event number to disable. |

**[Return value]**

If a trace setting was disabled successfully: True

If there was an error when disabling a trace setting: False

**[Detailed description]**

- This function disables the timer of the trace event specified by *traceNumber*.
- If *traceNumber* is not specified, then traces of all trace event numbers will be disabled.

**[Example of use]**

```
>>>debugger.Trace.Disable(1)
True
>>>
```

---

**debugger.Trace.Enable**

This function enables a conditional trace.

**[Specification format]**

```
debugger.Trace.Enable(traceNumber = "")
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| *traceNumber* | Specify the trace event number to enable. |

**[Return value]**

If a trace setting was enabled successfully: True

If there was an error when enabling a trace setting: False

**[Detailed description]**

- This function enables the timer of the trace event specified by *traceNumber*.
- If *traceNumber* is not specified, then traces of all trace event numbers will be enabled.

**[Example of use]**

```
>>>debugger.Trace.enable(1)
True
>>>
```

---

**debugger.Trace.Get**

This function dumps the trace data.

> **Remark**     This function provides the same function as debugger.XTrace.Dump.

## [Specification format]

```
debugger.Trace.Get(frameCount, fileName = "", append = False)
```

## [Argument(s)]

| Argument | Description |
|---|---|
| *frameCount* | Specify the number of dumps. |
| *fileName* | Specify the name of the file to dump to (default: not specified). |
| *append* | Specify whether to append trace data to the file. <br> True: Append trace data to the file. <br> False: Do not append trace data to the file (default). |

## [Return value]

List of trace information (see the TraceInfo property for detail)

## [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.Trace.Get(3)
    1851   00h00min00s003ms696μs000ns   0x000003be   cmp r11, r14
    1852   00h00min00s003ms700μs000ns   0x000003c0   blt _func_static3+0x2c
    1853   00h00min00s003ms702μs000ns   0x000003c2   jarl _errfunc, lp
>>>debugger.XTrace.Dump(10, "C:/test/TestTrace.txt")
>>>
```

---

**debugger.Trace.Information**

This function displays conditional trace information.

**[Specification format]**

```
debugger.Trace.Information()
```

**[Argument(s)]**

None

**[Return value]**

List of conditional trace information (see the TraceEventInfo class for detail)

**[Detailed description]**

- This function displays conditional trace information is shown by the following format.

```
trace-event-number Trace state start-address - end-address
```

**[Example of use]**

```
>>>ti = debugger.Trace.Information()
1 Trace Enable main - sub
>>>print ti[0].Number
1
>>>print ti[0].Name
Trace
>>>
```

---

---

**debugger.Trace.Set**

This function sets conditional trace information.

**[Specification format]**

```
debugger.Trace.Set(TraceCondition)
```

**[Argument(s)]**

| Argument | Description |
|----------|-------------|
| *TraceCondition* | Specify a condition of a conditional trace. See the TraceCondition class for creating a conditional trace. |

**[Return value]**

Set trace event number (numerical value)

**[Detailed description]**

- This function sets a conditional trace according to the contents specified with *TraceCondition*.
- The specified conditional trace is registered with the following name.

```
Trace
```

**[Example of use]**

```
>>>tc = TraceCondition()
>>>tc.StartAddress = "main"
>>>tc.EndAddress = "chData"
>>>tc.EndData = 0x20
>>>tc.EndTraceType = TraceType.Write
>>>ts_number = debugger.Trace.Set(tc)
1
>>>print ts_number
1
```

**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 |
|----------|-------------|
| *fileName* | Specify a file name. |
| *address1* | Specify an upload start address. |
| *address2* | Specify an upload end address. |
| *force* | Specify whether to overwrite.<br>　True: Overwrite<br>　False: Do not overwrite (default). |

**[Return value]**

If the memory data was uploaded successfully: True
If there was an error when uploading the memory data: False

**[Detailed description]**

- This function saves the memory data from *address1* to *address2* in binary format.

**[Example of use]**

```
>>>debugger.Upload.Binary("C:/test/testBinary.bin", 0x1000, 0x2000, True)
True
>>>
```

---

**debugger.Upload.Coverage**

This function saves the coverage data. [IECUBE][IECUBE2][Simulator]

## [Specification format]

```
debugger.Upload.Coverage(fileName, force = False)
```

## [Argument(s)]

| Argument | Description |
|---|---|
| *fileName* | Specify a file name. |
| *force* | Specify whether to overwrite.<br>True: Overwrite<br>False: Do not overwrite (default). |

## [Return value]

If the memory data was uploaded successfully: True

If there was an error when uploading the memory data: False

## [Detailed description]

- This function saves the coverage data to a file.

## [Example of use]

```
>>>debugger.Upload.Coverage("C:/test/coverageData.csrcv")
True
>>>
```

---

### 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 |
|---|---|
| *fileName* | Specify a file name. |
| *address1* | Specify an upload start address. |
| *address2* | Specify an upload end address. |
| *force* | Specify whether to overwrite.<br>  True: Overwrite<br>  False: Do not overwrite (default). |

**[Return value]**

If the memory data was uploaded successfully: True

If there was an error when uploading the memory data: False

**[Detailed description]**

  - This function saves the memory data from *address1* to *address2* in Intel format.

**[Example of use]**

```
>>>debugger.Upload.Intel("C:/test/testIntel.hex", 0x1000, 0x2000, True)
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 |
|----------|-------------|
| fileName | Specify a file name. |
| address1 | Specify an upload start address. |
| address2 | Specify an upload end address. |
| force | Specify whether to overwrite.<br>True: Overwrite<br>False: Do not overwrite (default). |

**[Return value]**

If the memory data was uploaded successfully: True

If there was an error when uploading the memory data: False

**[Detailed description]**

- This function saves the memory data from *address1* to *address2* in ID-tagged Intel format.

**[Example of use]**

```
>>>debugger.Upload.IntelIdTag("C:/test/testIdTagIntel.hex", 0x1000, 0x2000, True)
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 |
|----------|-------------|
| fileName | Specify a file name. |
| address1 | Specify an upload start address. |
| address2 | Specify an upload end address. |
| force | Specify whether to overwrite.<br>  True: Overwrite<br>  False: Do not overwrite (default). |

**[Return value]**

If the memory data was uploaded successfully: True

If there was an error when uploading the memory data: False

**[Detailed description]**

- This function saves the memory data from *address1* to *address2* in Motorola format.

**[Example of use]**

```
>>>debugger.Upload.Motorola("C:/test/testMotorola.hex", 0x1000, 0x2000, True)
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 |
|---|---|
| *fileName* | Specify a file name. |
| *address1* | Specify an upload start address. |
| *address2* | Specify an upload end address. |
| *force* | Specify whether to overwrite. <br> True: Overwrite <br> False: Do not overwrite (default). |

**[Return value]**

If the memory data was uploaded successfully: True
If there was an error when uploading the memory data: False

**[Detailed description]**

- This function saves the memory data from *address1* to *address2* in ID-tagged Motorola format.

**[Example of use]**

```
>>>debugger.Upload.MotorolaIdTag("C:/test/testIdTagMotorola.hex", 0x1000, 0x2000, True)
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 |
|---|---|
| *fileName* | Specify a file name. |
| *address1* | Specify an upload start address. |
| *address2* | Specify an upload end address. |
| *force* | Specify whether to overwrite.<br>True: Overwrite<br>False: Do not overwrite (default). |

**[Return value]**

If the memory data was uploaded successfully: True

If there was an error when uploading the memory data: False

**[Detailed description]**

   - This function saves the memory data from *address1* to *address2* in Techtronics format.

**[Example of use]**

```
>>>debugger.Upload.Tektronix("C:/test/testTektronix.hex", 0x1000, 0x2000, True)
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 |
|---|---|
| fileName | Specify a file name. |
| address1 | Specify an upload start address. |
| address2 | Specify an upload end address. |
| force | Specify whether to overwrite.<br>  True: Overwrite<br>  False: Do not overwrite (default). |

**[Return value]**

If the memory data was uploaded successfully: True

If there was an error when uploading the memory data: False

**[Detailed description]**

- This function saves the memory data from *address1* to *address2* in ID-tagged Techtronics format.

**[Example of use]**

```
>>>debugger.Upload.TektronixIdTag("C:/test/testIdTagTektronix.hex", 0x1000, 0x2000, True)
True
>>>
```

---

---

**debugger.Watch.GetValue**

This function refers to a variable value.

**[Specification format]**

```
debugger.Watch.GetValue(variableName, encode = Encoding.Default, watchOption =
WatchOption.Auto)
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| variableName | Specify the variable name, register name, or I/O register name/SFR register name to reference. |
| encode | Specify the encoding to use when displaying strings.<br>By default, the system encoding is used.<br>The encoding name conforms to the .NET specifications.<br>Examples: Encoding.utf-8, Encoding.euc-jp |
| watchOption | Specify an option.<br>The options that can be specified are shown below.<br><br>{table 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.SignedDecimal | Display in signed decimal format. |
| WatchOption.UnsignedDecimal | Display in unsigned decimal format. |
| WatchOption.Hexdecimal | Display in hexadecimal format. |
| WatchOption.String | Display as a string. |
| WatchOption.Sizeof | Display the variable size in decimal format. |
| WatchOption.Float | Display in float type. |
| WatchOption.Double | Display in double type. |

**[Return value]**

The displayed value is returned in the format specified by *watchOption*.

When *watchOption* is specified as "WatchOption.Auto", the format is returned to match the variable value.

However, if the return value is a double type, it is returned as a string (when *watchOption* is specified as "WatchOption.Double", or *watchOption* is specified as "WatchOption.Auto" and the return value is a double type).

**[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 |
|---|---|
| *variableName* | Specify the variable name, register name, and I/O register name or SFR register name to set. |
| *value* | Specify the value to set. |

### [Return value]

If a variable value was set successfully: True

If there was an error when setting a variable value: False

### [Detailed description]

- This function sets the value specified by *value* in the variable, register, and I/O register or 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)
True
>>>debugger.Watch.GetValue("testVal")
100
>>>debugger.Watch.GetValue("testVal", WatchOption.Hexdecimal)
0x64
>>>debugger.Watch.GetValue("testVal", WatchOption.Binary)
0b1100100
>>>debugger.Watch.SetValue("testVal", 0x256)
True
>>>debugger.Watch.GetValue("testVal", WatchOption.Hexdecimal)
0x256
```

---

---

### debugger.Where

This function displays a stack backtrace.

**[Specification format]**

```
debugger.Where()
```

**[Argument(s)]**

None

**[Return value]**

List of a backtrace (see the StackInfo property for detail)

**[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. [RL78][78K0R]**

**[Example of use]**

```
>>>debugger.Where()
1: test2.c#sub2#13
--- Information below might be inaccurate.
2:func.c#func#34
>>>
```

---

---

**debugger.Whereami**

This function displays a location.

**[Specification format]**

```
debugger.Whereami(address)
```

**[Argument(s)]**

| Argument | Description |
|----------|-------------|
| address | Specify the address of the location to display. |

**[Return value]**

Strings of the location

**[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. [IECUBE][IECUBE2][Simulator]

**[Specification format]**

```
debugger.XCoverage.Clear()
```

**[Argument(s)]**

None

**[Return value]**

If the coverage memory was cleared successfully: True

If there was an error when clearing the coverage memory: False

**[Detailed description]**

　　- This function clears the coverage memory.

**[Example of use]**

```
>>>debugger.XCoverageClear()
True
>>>
```

---

---

**debugger.XCoverage.GetCoverage**

This function gets the coverage. [IECUBE][IECUBE2][Simulator]

**[Specification format]**

```
debugger.XCoverage.GetCoverage(funcName, progName = "", fileName = "")
```

**[Argument(s)]**

| Argument | Description |
|---|---|
| funcName | Specify the function name to retrieve coverage for. |
| progName | Specify the name of the load module containing the function.<br>If there is only one load module, then this can be omitted (default). |
| fileName | Specify the name of the file containing the function.<br>If it is a global function, then this can be omitted (default). |

**Caution**　　**If two or more parameters are specified, then three parameters must be specified.**

**[Return value]**

Value without "%" (numeric value)

**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", "C:/test/Test.out", "C:/test/Test.c")
81.50%
>>>
```

---

**debugger.XRunBreak.Delete**

This function deletes XRunBreak information. [V850 Simulator]

**[Specification format]**

```
debugger.XRunBreak.Delete()
```

**[Argument(s)]**

None

**[Return value]**

If XRunBreak information was deleted successfully: True

If there was an error when deleting XRunBreak information: False

**[Detailed description]**

- This function deletes XRunBreak information.

**[Example of use]**

```
>>>debugger.XRunBreak.Refer()
None
>>>debugger.XRunBreak.Set(1, TimeType.S, True)
True
>>>debugger.XRunBreak.Refer()
1Second Periodic
>>>debugger.XRunBreak.Delete()
True
>>>debugger.XRunBreak.Refer()
None
```

---

**debugger.XRunBreak.Refer**

This function displays XRunBreak setting information. [V850 Simulator]

**[Specification format]**

```
debugger.XRunBreak.Refer()
```

**[Argument(s)]**

None

**[Return value]**

List of period time value and period information (TimeType) (see the XRunBreakInfo property for detail)

**[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)
True
>>>debugger.XRunBreak.Refer()
1Second Periodic
```

---

---

debugger.XRunBreak.Set

---

This function configures XRunBreak information. [V850 Simulator]

## [Specification format]

```
debugger.XRunBreak.Set(time, timeType = TimeType.Ms, periodic = False)
```

## [Argument(s)]

| Argument | Description |
|---|---|
| *time* | Specify the break time. |
| *timeType* | Specify the break time unit. <br> The units that can be specified are shown below. <br><br> | Type | Description | <br> |---|---| <br> | TimeType.Min | Minute unit | <br> | TimeType.S | Second unit | <br> | TimeType.Ms | Millisecond unit (default) | <br> | TimeType.Us | Microsecond unit | <br> | TimeType.Ns | Nanosecond unit | |
| *periodic* | Specify whether to call the callback every time the specified time elapses. <br>    True: Call at every specified time interval. <br>    False: Call one time only (default). |

## [Return value]

If XRunBreak information was configured successfully: True

If there was an error when configuring XRunBreak information: False

## [Detailed description]

- This function configures XRunBreak information.
- The XRunBreak calling interval depends on the simulator.
- Register the Python function that is processed after the specified time passes.  See "Hook" for detail.

**Caution**    **If you use the following operations while program is running after the XRunBreak information is set, please use these operations after program is stopped.**
   **- Resets the CPU**
   **- Resets the CPU and then executes the program from the reset address**
   **- Set/Remove Breakpoints**

---

**[Example of use]**

```
>>>debugger.XRunBreak.Refer()
None
>>>debugger.XRunBreak.Set(1, TimeType.S, True)
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]**

List of timing information (see the XTimeInfo property for detail)

**[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. [IECUBE][IECUBE2][Simulator]

**[Specification format]**

```
debugger.XTrace.Clear()
```

**[Argument(s)]**

None

**[Return value]**

If the trace memory was cleared successfully: True

If there was an error when clearing the trace memory: False

**[Detailed description]**

- This function clears the trace memory.

**[Example of use]**

```
>>>debugger.XTrace.Clear()
False
>>>
```

---

**debugger.XTrace.Dump**

This function dumps the trace data. [IECUBE][IECUBE2][Simulator]

## [Specification format]

```
debugger.XTrace.Dump(frameCount, fileName = "", append = False)
```

## [Argument(s)]

| Argument | Description |
|---|---|
| *frameCount* | Specify the number of dumps. |
| *fileName* | Specify the name of the file to dump to (default: not specified). |
| *append* | Specify whether to append trace data to the file.<br>    True: Append trace data to the file.<br>    False: Do not append trace data to the file (default). |

## [Return value]

List of trace information (see the TraceInfo property for detail)

## [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)
    1851  00h00min00s003ms696μs000ns  0x000003be  cmp r11, r14
    1852  00h00min00s003ms700μs000ns  0x000003c0  blt _func_static3+0x2c
    1853  00h00min00s003ms702μs000ns  0x000003c2  jarl _errfunc, lp
>>>debugger.XTrace.Dump(10, "C:/test/TestTrace.txt")
>>>
```

### G.3.6　CubeSuite+ Python class

Below is a list of CubeSuite+ Python classes.

**Table G-6.　CubeSuite+ Python Class**

| Class Name | Function Description |
|---|---|
| BreakCondition | This class creates a break condition. |
| BreakpointInfo | This class holds break point information. |
| BuildCompletedEventArgs | This class holds the parameters when a build completes. |
| DisassembleInfo | This class holds disassembly information. |
| DownloadInfo | This class holds download information. |
| FunctionInfo | This class holds function information. |
| MapInfo | This class holds map information. |
| StackInfo | This class holds stack information. |
| TimerCondition | This class creates conditions of a conditional timer. |
| TimerEventInfo | This class holds conditional timer event information. |
| TimerInfo | This class holds conditional timer information. |
| TraceCondition | This class creates conditions of a conditional trace. |
| TraceEventInfo | This class holds conditional trace event information. |
| TraceInfo | This class holds trace information. |
| VariableInfo | This class holds variable information. |
| XRunBreakInfo | This class holds XRunBreak information. |
| XTimeInfo | This class holds timer information. |

---

**BreakCondition**

This class creates a break condition.

## [Type]

```
class BreakCondition:
        Address = ""
        Data = None
        AccessSize = None
        BreakType = BreakType.Hardware
```

## [Variable]

| Variable | Description |
|---|---|
| Address | Specify the address at which to set a break.<br>Must be specified. |
| Data | Specify the number to set as a break condition for the data.<br>If "None" is specified, then the data condition is ignored. |
| AccessSize | Specify the access size (8, 16, or 32).<br>If "None" is specified, then all access sizes will be specified. |
| BreakType | Specify the break type.<br>The break types that can be specified are shown below.<br><br><table><tr><td>Type</td><td>Description</td></tr><tr><td>BreakType.Software</td><td>Software break (except a simulator)</td></tr><tr><td>BreakType.Hardware</td><td>Hardware break (default)</td></tr><tr><td>BreakType.Read</td><td>Data read break</td></tr><tr><td>BreakType.Write</td><td>Data write break</td></tr><tr><td>BreakType.Access</td><td>Data access break</td></tr></table> |

## [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

>>>
```

**BreakpointInfo**

This class holds break point information (return value of the debugger.Breakpoint.Information function).

**[Type]**

```
class BreakpointInfo:
        Number = 0
        Name = None
        Enable = True
        BreakType = BreakType.Hardware
        Address1 = None
        Address2 = None
        Address3 = None
        Address4 = None
```

**[Variable]**

| Variable | Description | |
|---|---|---|
| Number | This holds the event number. | |
| Name | This holds the name of the break point. | |
| Enable | This holds whether the break point is enabled or not.<br>   True: Enabled<br>   False: Disabled | |
| BreakType | This holds the break type. | |
| | Type | Description |
| | BreakType.Software | Software break (except a simulator) |
| | BreakType.Hardware | Hardware break |
| | BreakType.Read | Data read break |
| | BreakType.Write | Data write break |
| Address1 | This holds address information 1 as a string. | |
| Address2 | This holds address information 2 as a string (Only for combined breaks). | |
| Address3 | This holds address information 3 as a string (Only for combined breaks). | |
| Address4 | This holds address information 4 as a string (Only for combined breaks). | |

**[Detailed description]**

- BreakpointInfo is a class, and it is passed as the return value when the debugger.Breakpoint.Information function is executed.

**[Example of use]**

```
>>>info = debugger.Breakpoint.Information()
   1 Break0001 Enable  test1.c#_main+2
   2 Break0002 Disable test2.c#_sub4+10
>>>print info[0].Number
1
>>>print info[0].Name
Break0001
>>>print info[0].BreakType
Hardware
>>>print info[0].Enable
True
>>>print info[0].Address1
test1.c#_main+2
>>>print info[0].Address2
None
>>>print info[1].Number
2
>>>print info[1].Name
Break0002
>>>print info[1].BreakType
Hardware
>>>print info[1].Enable
False
>>>print info[1].Address1
test2.c#_sub4+10
>>>print info[1].Address2
None
>>>
```

---

**BuildCompletedEventArgs**

This class holds the parameters when a build completes.

## [Type]

```
class BuildCompletedEventArgs:
        Error = None
        Cancelled = False
        HasBuildError = False
        HasBuildWarning = False
```

## [Variable]

| Variable | Description |
|---|---|
| Error | When an exception occurs in the build, this holds the error contents (System.Exception). |
| Cancelled | This holds whether the build execution was canceled or not. |
| HasBuildError | This holds whether an error occurred in the build or not. |
| HasBuildWarning | This holds whether a warning occurred in the build or not. |

## [Detailed description]

- BreakCompletedEventArgs is a class, and it is passed as the argument only when the build.BuildCompleted event is issued.
It is not therefore possible to generate an instance of this class.

## [Example of use]

```
>>>def buildCompleted(sender, e):
... print "Error = {0}".format(e.Error)
... print "BuildError = " + e.HasBuildError.ToString()
... print "BuildWarning = " + e.HasBuildWarning.ToString()
... print "BuildCancelled = " + e.Cancelled.ToString()
...
>>>build.BuildCompleted += buildCompleted   ... Event connection
>>>build.All(True)
Error = None
BuildError = False
BuildWarning = False
BuildCancelled = False
True
>>>                                ... When an exception occurs, displayed as follows
>>>build.All(True)
Error = System.Exception:An error occurred during build.(E0203001)
BuildError = False
```

---

```
BuildWarning = False

BuildCancelled = False

False

>>>

>>>                                        ... When a build error occurs, displayed as follows

>>>build.All(True)

Error = None

BuildError = True

BuildWarning = False

BuildCancelled = False

False

>>>
```

---

**DisassembleInfo**

This class holds disassembly information (return value of the debugger.Assemble.Disassemble function).

## [Type]

```
class DisassembleInfo:
        Address = 0
        Code = None
        Mnemonic = None
```

## [Variable]

| Variable | Description |
|----------|-------------|
| Address | This holds the address. |
| Code | This holds code information as a collection of bytes. |
| Mnemonic | This holds mnemonic information. |

## [Detailed description]

- DisassembleInfo is a class, and it is the structure of the return value from the debugger.Assemble.Disassemble function.

## [Example of use]

```
>>>info = debugger.Assemble.Disassemble("main", 4)     ...Disassemble command
0x000002DC     B51D      br _main+0x36
0x000002DE     0132      mov0x1, r6
0x000002E0     60FF3800  jarl _func_static1, lp
0x000002E4     63570100  st.w r10, 0x0[sp]
>>>print info[0].Address
732
>>>print info[0].Code[0]
181
>>>print info[0].Code[1]
29
>>>print Mnemonic
br _main+0x36
>>>print info[3].Address
740
>>>print info[3].Code[0]
99
>>>print info[3].Code[1]
87
>>>print info[3].Code[2]
```

---

```
1
>>>print info[3].Code[3]
0
>>>print info[3].Mnemonic
st.w r10, 0x0[sp]
>>>
```

---

**DownloadInfo**

This class holds download information (return value of the debugger.Download.Information function).

**[Type]**

```
class DownloadInfo:
        Number = None
        Name = None
        ObjectDownload = True
        SymbolDownload = False
```

**[Variable]**

| Variable | Description |
|---|---|
| Number | This holds the download number. |
| Name | This holds the file name. |
| ObjectDownload | This holds whether object information has been downloaded or not.<br>  True: Object information has been downloaded.<br>  False: Object information has not been downloaded. |
| SymbolDownload | This holds whether symbol information has been downloaded or not.<br>  True: Symbol information has been downloaded.<br>  False: Symbol information has not been downloaded. |

**[Detailed description]**

- DownloadInfo is a class, and it is the structure of the return value from the debugger.Download.Information function.

**[Example of use]**

```
>>>info = debugger.Download.Information()
   1: DefaultBuild¥sample.out
>>>print info[0].Number
1
>>>print info[0].Name
DefaultBuild¥sample.out
>>>print info[0].ObjectDownload
True
>>>print info[0].SymbolDownload
True
>>>
```

---

RENESAS

---

> **FunctionInfo**

This class holds function information (return value of the project.GetFunctionList function).

**[Type]**

```
class FunctionInfo:
        FunctionName = None
        FileName = None
        ReturnType = None
        StartAddress = None
        EndAddress = None
```

**[Variable]**

| Variable | Description |
|---|---|
| FunctionName | This holds the function name. |
| FileName | This holds the full path of the file that the function is defined. |
| ReturnType | This holds the type of the return value. |
| StartAddress | This holds the start address of the function. |
| EndAddress | This holds the end address of the function. |

**[Detailed description]**

- FunctionInfo is a class, and it is the structure of the return value from the project.GetFunctionList function.

**[Example of use]**

```
>>>info = project.GetFunctionList()
func1 int 0x00200 0x00224 C:\project\src\test1.c
func2 int 0x00225 0x002ff C:\project\src\test2.c
>>>print info[0].FunctionName
func1
>>>print info[1].FileName
C:\project\src\test2.c
>>>print info[0].StartAddress
512
>>>
```

---

**MapInfo**

This class holds map information (return value of the debugger.Map.Information function).

**[Type]**

```
class MapInfo:
        Number = 0
        StartAddress = 0
        EndAddress = 0
        AccessSize = 0
        MapTypeName = None
```

**[Variable]**

| Variable | Description |
|----------|-------------|
| Number | This holds the number. |
| StartAddress | This holds the start address of the map area. |
| EndAddress | This holds the end address of the map area. |
| AccessSize | This holds the access size of the map area. |
| MapTypeName | This holds the type name of the map area. |

**[Detailed description]**

- MapInfo is a class, and it is the structure of the return value from the debugger.Map.Information function.

**[Example of use]**

```
>>>info = debugger.Map.Information()    ...Execute Map.Information function
   1: 0x00000000 0x0003FFFF 32 (Internal ROM area)
   2: 0x00040000 0x00048FFF  8 (Non map area)
   3: 0x00049000 0x001003FF  8 (Emulation ROM area)
   4: 0x00100400 0x03FF8FFF  8 (Non map area)
   5: 0x03FF9000 0x03FFEFFF 32 (Internal RAM area)
   6: 0x03FFF000 0x03FFFFFF  8 (I/O register area)
>>>print info[0].StartAddress
0
>>>print info[0].EndAddress
262143
>>>print info[0].AccessSize
32
>>>print info[0].MapTypeName
Internal ROM area
>>>print info[5].StartAddress
67104768
```

```
>>>print info[5].EndAddress

67108863

>>>print info[5].AccessSize

8

>>>print info[5].MapTypeName

I/O register area

>>>
```

---

**StackInfo**

This class holds stack information (return value of the debugger.Where function).

**[Type]**

```
class StackInfo:
        Number = 0
        AddressInfoText = None
```

**[Variable]**

| Variable | Description |
|---|---|
| Number | This holds the stack number. |
| AddressInfoText | This holds the stack address information as a string. |

**[Detailed description]**

- StackInfo is a class, and it is the structure of the return value from the debugger.Where function.

**[Example of use]**

```
>>>info = debugger.Where()
   1: test2.c#
   2: test1.c#main#41
>>>print info[0].Number
1
>>>print info[0].AddressInfoText
test2.c#
>>>info = debugger.Where
1: test2.c#
--- Information below might be inaccurate.
2: test1.c#main#41
>>>print a[1].Number
None
>>>print a[1].AddressInfoText
--- Information below might be inaccurate.
>>>
```

---

**TimerCondition**

This function creates conditions of a conditional timer.

## [Type]

```
class TimerCondition:
        StartAddress = ""
        StartData = ""
        StartTimerType = TimerType.Execution
        EndAddress = ""
        EndData = ""
        EndTimerType = TimerType.Execution
```

## [Variable]

| Variable | Description | |
|---|---|---|
| StartAddress | Specify an address starting timer measurement.<br>Must be specified. | |
| StartData | Specify a data condition (number) of an address starting timer measurement.<br>This specification is ignored if "TimerType.Execution" is specified for StartTimerType. | |
| StartTimerType | Specify the type of timers which start timer measurement.<br>The types that can be specified are shown below. | |
| | Type | Description |
| | TimerType.Execution | Start a timer at execution (default) |
| | TimerType.Read | Start a timer at data read |
| | TimerType.Write | Start a timer at data write |
| | TimerType.Access | Start a timer at data access |
| EndAddress | Specify the type of timers which end timer measurement.<br>Must be specified. | |
| EndData | Specify a data condition (number) of an address ending timer measurement.<br>This specification is ignored if "TimerType.Execution" is specified for EndTimerType. | |
| EndTimerType | Specify the type of timers which end timer measurement.<br>The types that can be specified are shown below. | |
| | Type | Description |
| | TimerType.Execution | End a timer at execution (default) |
| | TimerType.Read | End a timer at data read |
| | TimerType.Write | End a timer at data write |
| | TimerType.Access | End a timer at data access |

## [Detailed description]

- "TimerCondition" is in class format, and the condition of a conditional timer is set in the variable.

  In order to create a condition of a conditional timer, create an instance, and set conditions for that instance.

---

**[Example of use]**

```
>>>execute_timer = TimerCondition()      ... Create instance

>>>execute_timer.StartAddress = "main"

>>>execute_timer.StartTimerType = TimerType.Execution

>>>execute_timer.EndAddress = "sub"

>>>execute_timer.EndTimerType = TimerType.Execution

>>>debugger.Timer.Set(execute_timer)     ... Specify function in which to set the conditional
timer in parameter

1

>>>
```

---

**TimerEventInfo**

This class holds conditional timer event information (return value of the debugger.Timer.Information function).

**[Type]**

```
class TimerEventInfo:
        Number = 0
        Name = ""
        Enable = True
        StartAddress = ""
        StartData = ""
        StartTimerType = TimerType.Execution
        EndAddress = ""
        EndData = ""
        EndTimerType = TimerType.Execution
```

**[Variable]**

| Variable | Description | |
|----------|-------------|---|
| Number | This holds the timer event number. | |
| Name | This holds the name of the timer. | |
| Enable | This holds whether the timer is enabled or not.<br>  True: Enabled<br>  False: Disabled | |
| StartAddress | This holds the address starting timer measurement. | |
| StartData | This holds the data condition (number) of an address starting timer measurement. | |
| StartTimerType | This holds the type of timers which start timer measurement. | |
| | Type | Description |
| | TimerType.Execution | Start a timer at execution |
| | TimerType.Read | Start a timer at data read |
| | TimerType.Write | Start a timer at data write |
| | TimerType.Access | Start a timer at data access |
| EndAddress | This holds the address ending timer measurement. | |
| EndData | This holds the data condition (number) of an address ending timer measurement. | |
| EndTimerType | This holds the type of timers which end timer measurement. | |
| | Type | Description |
| | TimerType.Execution | End a timer at execution |
| | TimerType.Read | End a timer at data read |
| | TimerType.Write | End a timer at data write |
| | TimerType.Access | End a timer at data access |

---

**[Detailed description]**

- TimerEventInfo is a class, and it is passed as the return value when the debugger.Timer.Information function is executed.

**[Example of use]**

```
>>>info = debugger.Timer.Information()
1 PythonTimer0001 Enable main - sub
>>>print info[0].Number
1
>>>print info[0].Name
PythonTimer0001
>>>print info[0].Enable
True
>>>
```

**TimerInfo**

This class holds conditional timer information (return value of the debugger.Timer.Get function).

**[Type]**

```
class TimerInfo:
        Number = 0

        MaxTime = 0

        MaxClockCount = 0

        IsMaxOverflow = False

        MinTime = 0

        MinClockCount = 0

        IsMinOverflow = False

        AverageTime = 0

        AverageClockCount = 0

        IsAverageOverflow = False

        TotalTime = 0

        TotalClockCount = 0

        IsTotalOverflow = False

        PassCount = 0

        IsPassCountOverflow = False
```

**[Variable]**

| Variable | Description |
|---|---|
| Number | This holds the timer event number. |
| MaxTime | This holds the maximum execution time. |
| MaxClockCount | This holds the maximum number of clocks to be executed. |
| IsMaxOverflow | This holds whether the maximum execution time or number of clocks was overflowed.<br>   True: The maximum execution time or number of clocks was overflowed.<br>   False: The maximum execution time or number of clocks was not overflowed. |
| MinTime | This holds the minimum execution time. |
| MinClockCount | This holds the minimum number of clocks to be executed. |
| IsMinOverflow | This holds whether the minimum execution time or number of clocks was overflowed.<br>   True: The minimum execution time or number of clocks was overflowed.<br>   False: The minimum execution time or number of clocks was not overflowed. |
| AverageTime | This holds the average execution time. |
| AverageClockCount | This holds the average execution number of clocks. |
| IsAverageOverflow | This holds whether the average execution time or number of clocks was overflowed.<br>   True: The average execution time or number of clocks was overflowed.<br>   False: The average execution time or number of clocks was not overflowed. |
| TotalTime | This holds the total execution time. |
| TotalClockCount | This holds the total execution number of clocks. |

| Variable | Description |
|---|---|
| IsTotalOverflow | This holds whether the total execution time or number of clocks was overflowed.<br><br>True: The total execution time or number of clocks was overflowed.<br><br>False: The total execution time or number of clocks was not overflowed. |
| PassCount | This holds the pass count. |
| IsPassCountOverflow | This holds whether the pass count was overflowed.<br><br>True: The pass count was overflowed.<br><br>False: The pass count was not overflowed. |

**[Detailed description]**

- TimerInfo is a class, and it is passed as the return value when the debugger.Timer.Get function is executed.

**[Example of use]**

```
>>>info = debugger.Timer.Get()
1 Total: 2000 ns, Pass Count: 4 , Average: 500 ns, Max: 800 ns, Min: 300 ns
>>>print info[0].Number
1
>>>print info[0].MaxTime
800
>>>print info[0].PassCount
4
>>>print info[0].IsMaxOverflow
False
>>>
```

**TraceCondition**

This class creates conditions of a conditional trace.

## [Type]

```
class TraceCondition:
        StartAddress = ""
        StartData = ""
        StartTraceType = TraceType.Execution
        EndAddress = ""
        EndData = ""
        EndTraceType = TraceType.Execution
```

## [Variable]

| Variable | Description | |
|---|---|---|
| StartAddress | Specify an address starting a trace.<br>Must be specified. | |
| StartData | Specify a data condition (number) of an address starting a trace.<br>This specification is ignored if "TraceType.Execution" is specified for StartTraceType. | |
| StartTraceType | Specify the type of timers which start a trace.<br>The types that can be specified are shown below. | |
| | **Type** | **Description** |
| | TraceType.Execution | Start a trace at execution (default) |
| | TraceType.Read | Start a trace at data read |
| | TraceType.Write | Start a trace at data write |
| | TraceType.Access | Start a trace at data access |
| EndAddress | Specify the type of timers which end a trace.<br>Must be specified. | |
| EndData | Specify a data condition (number) of an address ending a trace.<br>This specification is ignored if "TraceType.Execution" is specified for EndTraceType. | |
| EndTraceType | Specify the type of timers which end a trace.<br>The types that can be specified are shown below. | |
| | **Type** | **Description** |
| | TraceType.Execution | Start a trace at execution (default) |
| | TraceType.Read | Start a trace at data read |
| | TraceType.Write | Start a trace at data write |
| | TraceType.Access | Start a trace at data access |

## [Detailed description]

- "TraceCondition" is in class format, and the condition of a conditional trace is set in the variable.

  In order to create a condition of a conditional trace, create an instance, and set conditions for that instance.

**[Example of use]**

```
>>>execute_trace = TraceCondition()      ... Create instance

>>>execute_trace.StartAddress = "main"

>>>execute_trace.StartTraceType = TraceType.Execution

>>>execute_trace.EndAddress = "sub"

>>>execute_trace.EndTraceType = TraceType.Execution

>>>debugger.Trace.Set(execute_trace)     ... Specify function in which to set the conditional
trace in parameter

1

>>>
```

---

**TraceEventInfo**

This class holds conditional trace event information (return value of the debugger.Trace.Information function).

**[Type]**

```
class TraceEventInfo:
        Number = 0
        Name = ""
        Enable = True
        StartAddress = ""
        StartData = ""
        StartTraceType = TraceType.Execution
        EndAddress = ""
        EndData = ""
        EndTraceType = TraceType.Execution
```

**[Variable]**

| Variable | Description | |
|---|---|---|
| Number | This holds the trace event number. | |
| Name | This holds the name of the trace. | |
| Enable | This holds whether the trace is enabled or not.<br>  True: Enabled<br>  False: Disabled | |
| StartAddress | This holds an address starting a trace. | |
| StartData | This holds a data condition (number) of an address starting a trace. | |
| StartTraceType | This holds the type of timers which start a trace. | |
| | Type | Description |
| | TraceType.Execution | Start a trace at execution |
| | TraceType.Read | Start a trace at data read |
| | TraceType.Write | Start a trace at data write |
| | TraceType.Access | Start a trace at data access |
| EndAddress | This holds an address ending a trace. | |
| EndData | This holds a data condition (number) of an address ending a trace. | |
| EndTraceType | This holds the type of timers which end a trace. | |
| | Type | Description |
| | TraceType.Execution | Start a trace at execution |
| | TraceType.Read | Start a trace at data read |
| | TraceType.Write | Start a trace at data write |
| | TraceType.Access | Start a trace at data access |

**[Detailed description]**

- TraceEventInfo is a class, and it is passed as the return value when the debugger.Trace.Information function is executed.

**[Example of use]**

```
>>>info = debugger.Trace.Information()
1 Trace Enable main - sub
>>>print info[0].Number
1
>>>print info[0].Name
Trace
>>>print info[0].Enable
True
>>>
```

---

**TraceInfo**

This class holds trace information (return value of the debugger.XTrace.Dump function).

**[Type]**

```
class TraceInfo:
        FrameNumber = None
        Timestamp = None
        FetchAddress = None
        Mnemonic = None
        ReadAddress = None
        ReadData = None
        WriteAddress = None
        WriteData = None
        VectorAddress = None
        VectorData = None
        IsDma = True
```

**[Variable]**

| Variable | Description |
|---|---|
| FrameNumber | This holds frame number information. |
| Timestamp | This holds time stamp information. |
| FetchAddress | This holds fetch address information. |
| Mnemonic | This holds mnemonic information. |
| ReadAddress | This holds read address information. |
| ReadData | This holds read data information. |
| WriteAddress | This holds write address information. |
| WriteData | This holds write data information. |
| VectorAddress | This holds vector address information. |
| VectorData | This holds the vector data. |
| IsDma | This holds whether the data is DMA or not.<br>   True: The data is DMA.<br>   False: The data is other than DMA. |

**[Detailed description]**

- TraceInfo is a class, and it is the structure of the return value from the debugger.XTrace.Dump function.

**[Example of use]**

```
>>>info = debugger.XTrace.Dump(10)
   853   00h00min00s001ms704us000ns  0x000002c2 movhi 0xffff, gp, r1
   854   00h00min00s001ms706us000ns  0x000002c6 id.w 0x7ff4[r1], r6
   855   00h00min00s001ms706us000ns                               0x03ff9000 R 0x00000000
   856   00h00min00s001ms706us000ns  0x000002ca movhi 0xffff, gp, r1
   857   00h00min00s001ms710us000ns  0x000002ce movea 0x7ff8, r1, r7
   858   00h00min00s001ms712us000ns  0x000002d2 jarl _main+0x36
   859   00h00min00s001ms716us000ns  0x000002dc br _main+0x36
   860   00h00min00s001ms720us000ns  0x00000312 prepare lp, 0x4
   861   00h00min00s001ms720us000ns                               0x03ff9308 W 0x000002d6
   862   00h00min00s001ms724us000ns  0x00000316 br _main+0x2
>>>print info[0].FrameNumber
853
>>>print info[0].Timestamp
1704000
>>>print info[0].FetchAddress
706
>>>print info[0].Mnemonic
movhi 0xffff, gp, r1
>>>print info[0].ReadAddress
None
>>>print info[0].ReadData
None
>>>print info[0].IsDma
False
>>>
>>>print info[2].FrameNumber
855
>>> print info[2].Timestamp
1706000
>>>print info[2].FetchAddress
None
>>>print info[2].Mnemonic
None
>>>print info[2].ReadAddress
67080192
```

---

**VariableInfo**

This class holds variable information (return value of the project.GetVariableList function).

**[Type]**

```
class VariableInfo:
        VariableName = None
        FileName = None
        Attribute = None
        Type = None
        Address = None
        Size = None
```

**[Variable]**

| Variable | Description |
|----------|-------------|
| VariableName | This holds the variable name. |
| FileName | This holds the full path of the file that the variable is defined. |
| Attribute | This holds the attribute. |
| Type | This holds the type. |
| Address | This holds the address. |
| Size | This holds the size. |

**[Detailed description]**

- VariableInfo is a class, and it is the structure of the return value from the project.GetVariableList function.

**[Example of use]**

```
>>>info = project.GetVariableList()
var1 volatile int 0x000014e4 4 C:\project\src\test1.c
var2 static int 0x000014e8 4 C:\project\src\test2.c
>>>print info[0].VariableName
var1
>>>print info[1].FileName
C:\project\src\test2.c
>>>print info[0].Attribute
volatile
>>>print info[0].Type
int
>>>
```

---

---

**XRunBreakInfo**

This class holds XRunBreak information (return value of the debugger.XRunBreak.Refer function).

**[Type]**

```
class XRunBreakInfo:
        Value = 0
        TimeType = Timetype.Min
        IsPeriodic = True
```

**[Variable]**

| Variable | Description | | |
|----------|-------------|---|---|
| Value | This holds the event interval value. | | |
| TimeType | This holds the unit of the interval value. | | |
| | Type | | Description |
| | TimeType.Min | | Minute unit |
| | TimeType.S | | Second unit |
| | TimeType.Ms | | Millisecond unit |
| | TimeType.Us | | Microsecond unit |
| | TimeType.Ns | | Nanosecond unit |
| IsPeriodic | This holds whether the callback is used periodically. | | |

**[Detailed description]**

- XRunBreakInfo is a class, and it is passed as the return value when the debugger.XRunBreak.Refer function is executed.

**[Example of use]**

```
>>>debugger.XRunBreak.Set(10, TimeType.S, True)
>>>info = debugger.XRunBreak.Refer()
10Second Periodic
>>>print info.Value
10
>>>print info.TimeType
S
>>>print info.IsPeriodic
True
>>>
```

<div style="border:1px solid black;padding:8px">

**XTimeInfo**

</div>

This class holds timer information (return value of the debugger.XTime function).

**[Type]**

```
class XTimeInfo:
        Value = 0
        IsCpuClock = False
        IsOverFlow = False
```

**[Variable]**

| Variable | Description |
|---|---|
| Value | This holds the timer measurement. |
| IsCpuClock | This holds whether this is a CPU clock measurement or not.<br>    True: This is a CPU clock measurement.<br>    False: Otherwise. |
| IsOverFlow | This holds whether an overflow has occurred or not.<br>    True: An overflow has occurred.<br>    False: An overflow has not occurred. |

**[Detailed description]**

   - XTimeInfo is a class, and it is the structure of the return value from the debugger.XTime function.

**[Example of use]**

```
>>>info = debugger.XTime()
9820214200nsec
>>>print info.Value
9820214200
>>>print info.IsCpuClock
False
>>>print info.IsOverFlow
False
>>>
```

### G.3.7   CubeSuite+ Python property (common)

Below is a list of CubeSuite+ Python properties (common).

**Table G-7.   CubeSuite+ Python Property (Common)**

| Property Name | Function Description |
|---|---|
| common.ConsoleClear | This property sets or refers to whether to clear the display of the Python console when changing the active project. |
| common.EnableRemotingStartup | This property sets and displays the setting for enabling or disabling the function for linking to an external tool at CubeSuite+ startup. |
| common.Output | This property refers to the return value or the contents of an error. |
| common.ThrowExcept | This property sets or refers to whether to throw an exception during the Python function is executed. |
| common.UseRemoting | This property sets and displays the setting for enabling or disabling the function for linking to an external tool while Python console is operating. |
| common.Version | This property refers to the version of CubeSuite+. |
| common.ViewLine | This property sets or refers to the number of screen lines for the Python console. |
| common.ViewOutput | This property sets and displays the setting for whether or not to display results of Python functions for CubeSuite+ and error messages in the Python console. |

**common.ConsoleClear**

This property sets or refers to whether to clear the display of the Python console when changing the active project.

**[Specification format]**

```
common.ConsoleClear = bool
```

**[Setting(s)]**

| Setting | Description |
|---------|-------------|
| *bool* | Set whether to verify during writes.<br>    True: Verify during writes.<br>    False: Do not verify during writes. |

**[Reference]**

Current set value

**[Detailed description]**

- This property sets or refers to whether to clear the display of the Python console when changing the active project.

**[Example of use]**

```
>>>print common.ConsoleClear
True
>>>common.ConsoleClear = False
```

---

**common.EnableRemotingStartup**

This property sets and displays the setting for enabling or disabling the function for linking to an external tool at Cube-Suite+ startup.

**[Specification format]**

```
common.EnableRemotingStartup = bool
```

**[Setting(s)]**

| Setting | Description |
|---|---|
| *bool* | Set whether to enable or disable the function for linking to an external tool at CubeSuite+ startup.<br><br>True: Enable the function for linking to an external tool (default).<br><br>False: Disable the function for linking to an external tool.<br><br>Use the common.UseRemoting property to enable or disable linking to an external tool while running. |

**[Reference]**

Current set value

**[Detailed description]**

- This property sets and displays the setting for enabling or disabling the function for linking to an external tool at CubeSuite+ startup.

**[Example of use]**

```
>>>print common.EnableRemotingStartup
False
>>>common.EnableRemotingStartup = True
```

---

---

**common.Output**

This property refers to the execution result or the contents of an error.

**[Specification format]**

```
common.Output
```

**[Setting(s)]**

None

**[Reference]**

Execution result or an error message of the CubeSuite+ Python function (strings)

> **Caution**　**Error messages can only be referred to when the common.ThrowExcept property is set not to throw an exception (False).**

> **Remark**　The reference content is retained until the next CubeSuite+ Python function call.

**[Detailed description]**

- This property refers to the execution result or the contents of an error.

**[Example of use]**

```
>>>debugger.Memory.Read("data")
0x0
>>>print common.Output
0
```

---

---

**common.ThrowExcept**

This property sets or refers to whether to throw an exception during the Python function is executed.

**[Specification format]**

```
common.ThrowExcept = bool
```

**[Setting(s)]**

| Setting | Description |
|---------|-------------|
| bool | Set whether to throw an exception during the Python function is executed.<br>True: Throw an exception.<br>False: Do not throw an exception (default). |

**[Reference]**

Current set value

**[Detailed description]**

- This property sets or refers to whether to throw an exception during the Python function is executed.
- To use the try-except statement, set *bool* to "True".

**[Example of use]**

```
>>>print common.ThrowExcept
False
>>>common.ThrowExcept = True
```

---

> **common.UseRemoting**

This property sets and displays the setting for enabling or disabling the function for linking to an external tool while Python console is operating.

**[Specification format]**

```
common.UseRemoting = bool
```

**[Setting(s)]**

| Setting | Description |
|---------|-------------|
| *bool* | Set whether to enable or disable the function for linking to an external tool while Python console is operating.<br><br>   True: Enable the function for linking to an external tool (default).<br><br>   False: Disable the function for linking to an external tool.<br><br>This will be True if the common.EnableRemotingStartup property is set to True on startup, and False otherwise. |

**[Reference]**

Current set value

**[Detailed description]**

- This property sets and displays the setting for enabling or disabling the function for linking to an external tool while Python console is operating.

**[Example of use]s**

```
>>>print common.UseRemoting
False
>>>common.UseRemoting = True
```

---

---

**common.Version**

This property refers to the version of CubeSuite+.

**[Specification format]**

```
common.Version
```

**[Setting(s)]**

None

**[Reference]**

Version of CubeSuite+

**[Detailed description]**

- This property refers to the version of CubeSuite+.

**[Example of use]**

```
>>>print common.Version
V1.02.00  [01 Apr 2012]
```

---

---

**common.ViewLine**

This property sets or refers to the number of screen lines for the Python console.

**[Specification format]**

```
common.ViewLine = number
```

**[Setting(s)]**

| Setting | Description |
|---------|-------------|
| number | Set the number of screen lines for the Python console (default: 10000). |

**[Reference]**

Current set value

**[Detailed description]**

- This property sets or refers to the number of screen lines for the Python console.

**[Example of use]**

```
>>>print common.ViewLine
10000
>>>common.ViewLine = 20000
```

> **common.ViewOutput**

This property sets and displays the setting for whether or not to display results of Python functions for CubeSuite+ and error messages in the Python console.

**[Specification format]**

```
common.ViewOutput = bool
```

**[Setting(s)]**

| Setting | Description |
|---------|-------------|
| *bool* | Set whether or not to display results of Python functions for CubeSuite+ and error messages in the Python console.<br><br>True: Display in the Python console (default).<br><br>False: Do not display in the Python console. |

**[Reference]**

Current set value

**[Detailed description]**

- This property sets and displays the setting for whether or not to display results of Python functions for CubeSuite+ and error messages in the Python console.

**[Example of use]**

```
>>>print common.ViewOutput
False
>>>common.ViewOutput = True
```

**G.3.8    CubeSuite+ Python property (for project)**

Below is a list of CubeSuite+ Python properties (for a project).

**Table G-8.   CubeSuite+ Python Property (For Project)**

| Property Name | Function Description |
| --- | --- |
| project.Device | This property refers to the microcontroller of the active project. |
| project.IsOpen | This property confirms whether the project has been opened. |
| project.Kind | This property refers to the kind of the active project. |
| project.Name | This property refers to the active project file name (without path). |
| project.Nickname | This property refers to the nickname of the microcontroller of the active project. |
| project.Path | This property refers to the active project file name (with path). |

---

**project.Device**

This property refers to the microcontroller of the active project.

**[Specification format]**

```
project.Device
```

**[Setting(s)]**

None

**[Reference]**

Microcontroller of the active project

**[Detailed description]**

- This property refers to the microcontroller of the active project.

**[Example of use]**

```
>>>print project.Device
R5F100LE
```

---

**project.IsOpen**

This property confirms whether the project has been opened.

**[Specification format]**

```
project.IsOpen
```

**[Setting(s)]**

None

**[Reference]**

If the project has been opened: True

If the project has not been opened: False

**[Detailed description]**

- This property confirms whether the project has been opened.

**[Example of use]**

```
>>>print project.IsOpen
True
>>>
```

---

---

**project.Kind**

This property refers to the kind of the active project.

**[Specification format]**

```
project.Kind
```

**[Setting(s)]**

None

**[Reference]**

Kind of active project

| Type | Description |
|---|---|
| Application | Project for application |
| Library | Project for library |
| DebugOnly | Debug-dedicated project |
| Empty | Project for empty application |
| CppApplication | Project for C++ application |
| RI600V4 | Project for RI600V4 |
| RI600PX | Project for RI600PX |
| RI850V4 | Project for RI850V4 |
| RI850MP | Project for RI850MP |
| RI78V4 | Project for RI78V4 |
| MulticoreBootLoader | Project for boot loader for multi-core |
| MulticoreApplication | Project for application for multi-core |

**[Detailed description]**

- This property refers to the kind of the active project.

**[Example of use]**

```
>>>print project.Kind
Application
>>>
```

**project.Name**

This property refers to the active project file name (without path).

**[Specification format]**

```
project.Name
```

**[Setting(s)]**

None

**[Reference]**

Active project file name (without path)

**[Detailed description]**

- This property refers to the active project file name (without path).

**[Example of use]**

```
>>>print project.Name
test.mtpj
```

**project.Nickname**

This property refers to the nickname of the microcontroller of the active project.

**[Specification format]**

```
project.Nickname
```

**[Setting(s)]**

None

**[Reference]**

Nickname of the microcontroller of the active project

**[Detailed description]**

- This property refers to the nickname of the microcontroller of the active project.

**[Example of use]**

```
>>>print project.Nickname
RL78/G13 (ROM:64KB)
```

---

**project.Path**

This property refers to the active project file name (with path).

**[Specification format]**

```
project.Path
```

**[Setting(s)]**

None

**[Reference]**

Active project file name (with path)

**[Detailed description]**

- This property refers to the active project file name (with path).

**[Example of use]**

```
>>>print project.Path
C:/project/test.mtpj
```

**G.3.9 CubeSuite+ Python property (for build tool)**

Below is a list of CubeSuite+ Python properties (for the build tool).

**Table G-9. CubeSuite+ Python Property (For Build Tool)**

| Property Name | Function Description |
|---|---|
| build.Compile.IncludePath | This property sets or refers to the compile options for the active project regarding additional include paths. |
| build.IsBuilding | This property confirms whether a build is running. |
| build.Link.LibraryFile | This property sets or refers to library files of the active project. |
| build.Link.SectionAlignment | This property sets or refers to the link options for the active project regarding section alignment. |
| build.Link.SectionROMtoRAM | This property sets or refers to the link options for the active project regarding sections where symbols are mapped from ROM to RAM. |
| build.Link.SectionStartAddress | This property sets or refers to the link options for the active project regarding the addresses where sections start. |
| build.Link.SectionSymbolFile | This property sets or refers to the link options for the active project regarding sections whose external defined symbols are to be output to a file. |
| build.ROMization.OutputObjectFile | This property sets or refers to the setting for output of a ROMized object file, that is, the value of the ROMization process option for the active project. |

---

**build.Compile.IncludePath**

This property sets or refers to the compile options for the active project regarding additional include paths.

### [Specification format]

```
build.Compile.IncludePath = dirlist
```

### [Setting(s)]

| Setting | Description |
|---------|-------------|
| *dirlist* | Set the additional include paths as a list of strings. |

### [Reference]

List of additional include paths

### [Detailed description]

- This property sets or refers to the compile options for the active project regarding additional include paths.
- Add or change for the referred list to change the setting.

### [Example of use]

```
>>>incpath1 = build.Compile.IncludePath      ... Refer the current setting and add an include path
>>>print incpath1
['include', 'C:\project\inc']
>>>incpath1.append('include2')
>>>build.Compile.IncludePath = incpath1
>>>print build.Compile.IncludePath
['include', 'C:\project\inc', 'include2']
>>>
>>>incpath2 = ['include1', 'include2']       ... Set multiple include paths
>>>build.Compile.IncludePath = incpath2
>>>print build.Compile.IncludePath
['include1', 'include2']
```

**build.IsBuilding**

This property confirms whether a build is running.

**[Specification format]**

```
build.IsBuilding
```

**[Setting(s)]**

None

**[Reference]**

If a build is running: True

If a build is not run: False

**[Detailed description]**

- This property confirms whether a build is running.

**[Example of use]**

```
>>>print build.IsBuilding
False
>>>
```

**build.Link.LibraryFile**

This property sets or refers to library files of the active project.

**[Specification format]**

```
build.Link.LibraryFile = filelist
```

**[Setting(s)]**

| Setting | Description |
|---------|-------------|
| *filelist* | Set the library files of the active project as a list of strings. |

**[Reference]**

List of library files

**[Detailed description]**

  - This property sets or refers to library files of the active project.
  - Add or change for the referred list to change the setting.

**[Example of use]**

```
>>>lib1 = build.Link.LibraryFile        ... Refer the current setting and add a library file
>>>print lib1
['test1.lib', 'test2.lib']
>>>lib1.append("test3.lib")
>>>build.Link.LibraryFile = lib1
>>>print build.Link.LibraryFile
['test1.lib', 'test2.lib', 'test3.lib']
>>>
>>>lib2 = ['test1.lib', 'test2.lib']    ... Set multiple library files
>>>build.Link.LibraryFile = lib2
>>>print build.Link.LibraryFile
['test1.lib', 'test2.lib']
```

| build.Link.SectionAlignment |
| --- |

This property sets or refers to the link options for the active project regarding section alignment. [CC-RH][CC-RX]

### [Specification format]

```
build.Link.SectionAlignment = sectionlist
```

### [Setting(s)]

| Setting | Description |
| --- | --- |
| *sectionlist* | Set section alignment as a list of strings. |

### [Reference]

List of section alignment

### [Detailed description]

- This property sets or refers to the link options for the active project regarding section alignment.
- Add or change for the referred list to change the setting.

### [Example of use]

```
>>>lib1 = build.Link.LibraryFile      ... Refer the current setting and add section alignment
['R_1']
>>>sec1.append('R_2')
>>>build.Link.SectionAlignment = sec1
>>>print build.Link.SectionAlignment
['R_1', 'R_2']
>>>
>>>sec2 = ['R_1', 'R_2']              ... Set multiple section alignment
>>>build.Link.SectionAlignment = sec2
>>>print build.Link.SectionAlignment
['R_1', 'R_2']
```

---

**build.Link.SectionROMtoRAM**

This property sets or refers to the link options for the active project regarding sections where symbols are mapped from ROM to RAM. [CC-RH][CC-RX]

**[Specification format]**

```
build.Link.SectionROMtoRAM = sectionlist
```

**[Setting(s)]**

| Setting | Description |
|---|---|
| *sectionlist* | Set the section that maps symbols from ROM to RAM as a list of strings. |

**[Reference]**

List of the section that maps symbols from ROM to RAM

**[Detailed description]**

- This property sets or refers to the link options for the active project regarding sections where symbols are mapped from ROM to RAM.
- Add or change for the referred list to change the setting.

**[Example of use]**

```
>>>sec = build.Link.SectionROMtoRAM          ... Refer the current setting and add the section
                                                 that maps symbols from ROM to RAM
>>>print sec
['D=R', 'D_1=R_1', 'D_2=R_2']
>>>sec.append('D_3=R_3')
>>>build.Link.SectionROMtoRAM = sec
>>>print build.Link.SectionROMtoRAM
['D=R', 'D_1=R_1', 'D_2=R_2', 'D_3=R_3']
```

---

**build.Link.SectionStartAddress**

This property sets or refers to the link options for the active project regarding the addresses where sections start. [CC-RH][CC-RX]

**[Specification format]**

```
build.Link.SectionStartAddress = section
```

**[Setting(s)]**

| Setting | Description |
|---------|-------------|
| section | Set the start address of the section as strings. |

**[Reference]**

Start address of the section (strings)

**[Detailed description]**

- This property sets or refers to the link options for the active project regarding the addresses where sections start.
- Add or change for the referred strings to change the setting.

**[Example of use]**

```
>>>sec= build.Link.SectionStartAddress       ... Refer the current setting and change the start
                                                  address of the section
>>>print sec
B_1,R_1,B_2,R_2,B,R,SU,SI/01000,PResetPRG/0FFFF8000
>>>sec = "B_1/0200,R_1,B_2,R_2,B,R,SU,SI/01000,PResetPRG/0FFFF8000"
>>>build.Link.SectionStartAddress = sec
>>>print build.Link.SectionStartAddress
B_1/0200,R_1,B_2,R_2,B,R,SU,SI/01000,PResetPRG/0FFFF8000
```

---
**build.Link.SectionSymbolFile**
---

This property sets or refers to the link options for the active project regarding sections whose external defined symbols are to be output to a file. [CC-RH][CC-RX]

## [Specification format]

```
build.Link.SectionSymbolFile = sectionlist
```

## [Setting(s)]

| Setting | Description |
|---|---|
| *sectionlist* | Set the section whose external defined symbols are output to a file as a list of strings. |

## [Reference]

List of the section whose external defined symbols are output to a file

## [Detailed description]

- This property sets or refers to the link options for the active project regarding sections whose external defined symbols are to be output to a file.
- Add or change for the referred list to change the setting.

## [Example of use]

```
>>>sec = build.Link.SectionSymbolFile    ... Refer the current setting and add the section whose
                                              external defined symbols are output to a file
>>>print sec
['R_1', 'R_2']
>>>sec.append('R_3')
>>>build.Link.SectionSymbolFile = sec
>>>print build.Link.SectionSymbolFile
['R_1', 'R_2', 'R_3']
```

---

**build.ROMization.OutputObjectFile**

This property sets or refers to the setting for output of a ROMized object file, that is, the value of the ROMization process option for the active project. [CA850][CX][CA78K0R]

**[Specification format]**

```
build.ROMization.OutputObjectFile = bool
```

**[Setting(s)]**

| Setting | Description |
|---------|-------------|
| *bool* | Set whether or not to output the ROMized object file. |
| | True: Output the ROMized object file. |
| | False: Do not output the ROMized object file. |

**[Reference]**

If the ROMized object file is output: True

If the ROMized object file is not output: False

If the compiler is not supported: None

**[Detailed description]**

- This property sets or refers to the setting for output of a ROMized object file, that is, the value of the ROMization process option for the active project.

**[Example of use]**

```
>>>setting = build.ROMization.OutputObjectFile
>>>print setting
True
>>>build.ROMization.OutputObjectFile = False
>>>print build.ROMization.OutputObjectFile
False
```

### G.3.10   CubeSuite+ Python property (for debug tool)

Below is a list of CubeSuite+ Python properties (for the debug tool).

**Table G-10.   CubeSuite+ Python Property (For Debug Tool)**

| Property Name | Function Description |
|---|---|
| debugger.ADConvertDataInExecution | This property sets or refers to data collected in debugging. |
| debugger.IsMulticore | This property checks whether or not the microcontroller of the active project is multi-core. |
| debugger.Memory.NoVerify | This property switches the write-time verification setting. |
| debugger.Opiton.Coverage<br>debugger.Option.OpenBreak<br>debugger.Option.ReuseCoverageData<br>debugger.Option.Timer<br>debugger.Option.Trace<br>debugger.Option.UseTraceData | This property sets or refers to the options of the debug tool. |
| debugger.ProcessorElement | This property sets or refers to the PE of the multi-core. |
| debugger.XTrace.Addup<br>debugger.XTrace.Complement<br>debugger.XTrace.Mode | This property sets or refers to the tracing options of the debug tool. |

**debugger.ADConvertDataInExecution**

This property sets or refers to data collected in debugging. [Smart Analog]

**[Specification format]**

```
debugger.ADConvertDataInExecution = adConvertDataInExecution
```

**[Setting(s)]**

| Setting | Description |
|---------|-------------|
| *adConvertDataInExecution* | Set whether to collect data during debugging. <br> True: Collect data during debugging. <br> False: Do not collect data during debugging. |

**[Reference]**

Setting for data collection during execution

**[Detailed description]**

- This property sets or refers to data collected in debugging.

**[Example of use]**

```
>>>print debugeer.ADConvertDataInExecution
False
>>>debugger.ADConvertDataInExecution = True
>>>print debugger.ADConvertDataInExecution
True
>>>
```

---

**debugger.IsMulticore**

This property checks whether or not the microcontroller of the active project is multi-core.

**[Specification format]**

```
debugger.IsMulticore
```

**[Setting(s)]**

None

**[Reference]**

When the microcontroller is multi-core: True
When the microcontroller is not multi-core: False

**[Detailed description]**

- This function checks whether or not the microcontroller of the active project is multi-core.

**[Example of use]**

```
>>>print debugger.IsMulticore
False
>>>
```

---

---

**debugger.Memory.NoVerify**

This property switches the write-time verification setting. [Except simulator]

## [Specification format]

```
debugger.Memory.NoVerify = noverify
```

## [Setting(s)]

| Setting | Description |
|---------|-------------|
| *noverify* | Set whether to verify during writes.<br>  True: Verify during writes.<br>  False: Do not verify during writes. |

## [Reference]

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.Opiton.Coverage**

**debugger.Option.OpenBreak**

**debugger.Option.ReuseCoverageData**

**debugger.Option.Timer**

**debugger.Option.Trace**

**debugger.Option.UseTraceData**

---

This property sets or refers to the options of the debug tool.

**[Specification format]**

```
debugger.Opiton.Coverage = coverage
debugger.Option.OpenBreak = openBreak
debugger.Option.ReuseCoverageData = reuseCoverageData
debugger.Option.Timer = timer
debugger.Option.Trace = trace
debugger.Option.UseTraceData = useTraceDataType
```

**[Setting(s)]**

| Setting | Description |
|---|---|
| *coverage* | Set whether to use the coverage function. [IECUBE][IECUBE2][Simulator]<br>    True: Use the coverage function.<br>    False: Do not use the coverage function. |
| *openBreak* | Set whether to use the open break function.<br>    True: Use the open break function.<br>    False: Do not use the open break function. |
| *reuseCoverageData* | Set whether to reuse the coverage result.<br>    True: Reuse the coverage result.<br>    False: Do not reuse the coverage result. |
| *timer* | Set whether to use the timer function.<br>    True: Use the timer function.<br>    False: Do not use the timer function. |
| *trace* | Set whether to use the trace function. [IECUBE][IECUBE2][Simulator]<br>    True: Use the trace function.<br>    False: Do not use the trace function. |

| Setting | Description |
|---|---|
| *useTraceDataType* | Set which function to use the trace data in. [IECUBE [V850]][IECUBE2]<br>The functions that can be specified are shown below.<table><tr><td>Type</td><td>Description</td></tr><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></table> |

**[Reference]**

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 refers to 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.ProcessorElement**

This property sets or refers to the PE of the multi-core. [RH850]

**[Specification format]**

```
debugger.ProcessorElement = number
```

**[Setting(s)]**

| Setting | Description |
|---|---|
| *number* | Set the PE number with the number. |

**[Reference]**

Current set value

**[Detailed description]**

- This function sets or refers to the PE of the multi-core.

    **Caution    When the PE is set, it must be connected to the debugging tool.**

**[Example of use]**

```
>>>print debugger.ProcessorElement
1
>>>debugger.ProcessorElement = 2
>>>print debugger.ProcessorElement
2
>>>
```

---

**debugger.XTrace.Addup**

**debugger.XTrace.Complement**

**debugger.XTrace.Mode**

---

This property sets or refers to the tracing options of the debug tool. [IECUBE][IECUBE2][Simulator]

### [Specification format]

```
debugger.XTrace.Addup = addup [Simulator]

debugger.XTrace.Complement = complement [IECUBE[V850]][IECUBE2[V850]]

debugger.XTrace.Mode = traceMode [Simulator][IECUBE][IECUBE2]
```

### [Setting(s)]

| Setting | Description |
|---|---|
| *addup* | Set whether to add up times/tags.<br>True: Add up times/tags.<br>False: Do not add up times/tags. |
| *complement* | Set whether to supplement the trace.<br>True: Supplement the trace.<br>False: Do not supplement the trace. |
| *traceMode* | Set the trace control mode.<br>The trace control modes that can be specified are shown below.<table><tr><td>Type</td><td>Description</td></tr><tr><td>TraceMode.FullBreak</td><td>Stop program execution and writing of trace data after all trace data has been used up.</td></tr><tr><td>TraceMode.FullStop</td><td>Stop writing trace data after all trace data has been used up.</td></tr><tr><td>TraceMode.NonStop</td><td>Continue writing trace data even if all trace data has been used up.</td></tr></table> |

### [Reference]

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 refers to the tracing options of the debug tool.

**[Example of use]**

```
>>>print debugger.XTrace.Addup
False
>>>debugger.XTrace.Addup = True
>>>print debugger.XTrace.Addup
True
>>>
```

**G.3.11　CubeSuite+ Python event**

Below is a list of CubeSuite+ Python events.

**Table G-11.　CubeSuite+ Python Event**

| Event Name | Function Description |
|---|---|
| build.BuildCompleted | This event informs that a build has been completed. |

**build.BuildCompleted**

This event informs that a build has been completed.

**[Handler format]**

```
build.BuildCompleted(sender, e)
```

**[Handler argument(s)]**

| Argument | Description |
|----------|-------------|
| *sender* | The sender of the build event are passed. |
| *e* | The parameters at the end of build execution are passed. |

**[Return value]**

None

**[Detailed description]**

- This event informs that a build has been completed.

**[Example of use]**

```
>>>def buildCompleted(sender, e):
... print "Error = {0}".format(e.Error)
... print "BuildError = " + e.HasBuildError.ToString()
... print "BuildWarning = " + e.HasBuildWarning.ToString()
... print "BuildCancelled = " + e.Cancelled.ToString()
...
>>>build.BuildCompleted += buildCompleted       ... Event connection
>>>build.All(True)
Error = None
BuildError = False
BuildWarning = False
BuildCancelled = False
True
>>>
>>>build.File("C:/sample/src/test1.c")
Error = None
BuildError = False
BuildWarning = False
BuildCancelled = False
True
>>>
>>>
```

```
>>>build.Clean()

Error = None

BuildError = False

BuildWarning = False

BuildCancelled = False

True

>>>
```

### G.4    Cautions for Python Console

**(1)  Caution for Japanese input**

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

**(2)  Caution for prompt displays**

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

**(3)  Caution for executing scripts for projects without load modules**

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

**(4)  Cautions for forced termination**

If the following operations are performed while a script like an infinite loop is running, then the results of function execution may be an error, because the function execution will be terminated forcibly.

- Forcible termination by selecting "Forcibly terminate" from the context menu or pressing Ctrl+D in the Python Console
- Changing the active project in a project with multiple projects

**APPENDIX  H   INDEX**

Revision Record

| Rev. | Date | Description | |
|---|---|---|---|
| | | **Page** | **Summary** |
| 1.00 | Feb 01, 2014 | **-** | First Edition issued |

# RENESAS

CubeSuite+ V2.02.00

RENESAS

Renesas Electronics Corporation