

RX-family C/C++ Compiler Package

REJ06J0101-0100

Application Notes: Installation Guide - Renesas IDE Startup Guide,
RX Edition

Rev.1.00
Apr 20, 2010

This document explains the flow of operations for using the RX-family C/C++ compiler package, from product installation to project creation, building, and debugging, for first-time users.

| | |
|---|----|
| 1. Introduction..... | 2 |
| 2. Installation | 3 |
| 3. Setting Up a Project | 3 |
| 3.1 Starting high-performance embedded workshop | 3 |
| 3.2 Creating a workspace..... | 3 |
| 3.2.1 Workspaces and projects | 3 |
| 3.2.2 Creating a workspace..... | 4 |
| 3.2.3 Selecting the target MCU | 5 |
| 3.2.4 Selecting MCU options..... | 5 |
| 3.2.5 Setting up auto-generated files | 6 |
| 3.2.6 Selecting standard libraries..... | 6 |
| 3.2.7 Setting up the stack area | 7 |
| 3.2.8 Setting a reset vector | 7 |
| 3.2.9 Selecting a debugging target | 7 |
| 3.2.10 Selecting debugger options..... | 8 |
| 3.2.11 Completing workspace creation | 8 |
| 3.3 Setting up folders in a workspace | 9 |
| 4. Performing a Build..... | 10 |
| 4.1 About builds..... | 10 |
| 4.2 Build options | 10 |
| 4.2.1 Setting build options | 10 |
| 4.2.2 Option settings targets | 11 |
| 4.3 Configuration | 11 |
| 4.4 Performing a build | 11 |
| 5. Debugging..... | 13 |
| 5.1 Debugging sessions..... | 13 |
| 5.2 Target connections..... | 13 |
| 5.3 Downloading debugging target programs | 14 |
| 5.4 Executing and debugging programs | 14 |
| 5.4.1 Reset | 15 |
| 5.4.2 Changing the source program display mode | 15 |
| 5.4.3 Step execution (step in and step over)..... | 15 |
| 5.4.4 Referencing and settings register values | 16 |
| 5.4.5 Referencing function and variable definitions | 17 |
| 5.4.6 Setting up breakpoints | 17 |
| 5.4.7 Executing a program | 17 |
| 5.4.8 Watching | 17 |
| 5.4.9 Revising and re-evaluating programs | 18 |
| Website and Support..... | 19 |

1. Introduction

The RX-family C/C++ compiler package enables efficient creation of programs that take advantage of RX-family functionality and performance, using the IDE (Integrated Development Environment) and C/C++ compiler. It also includes a simulator, to enable program debugging and performance evaluation even without production machines.

Figure 1 shows the configuration of the Renesas development environment.

The related tools (software) shown in color are included with the compiler package.

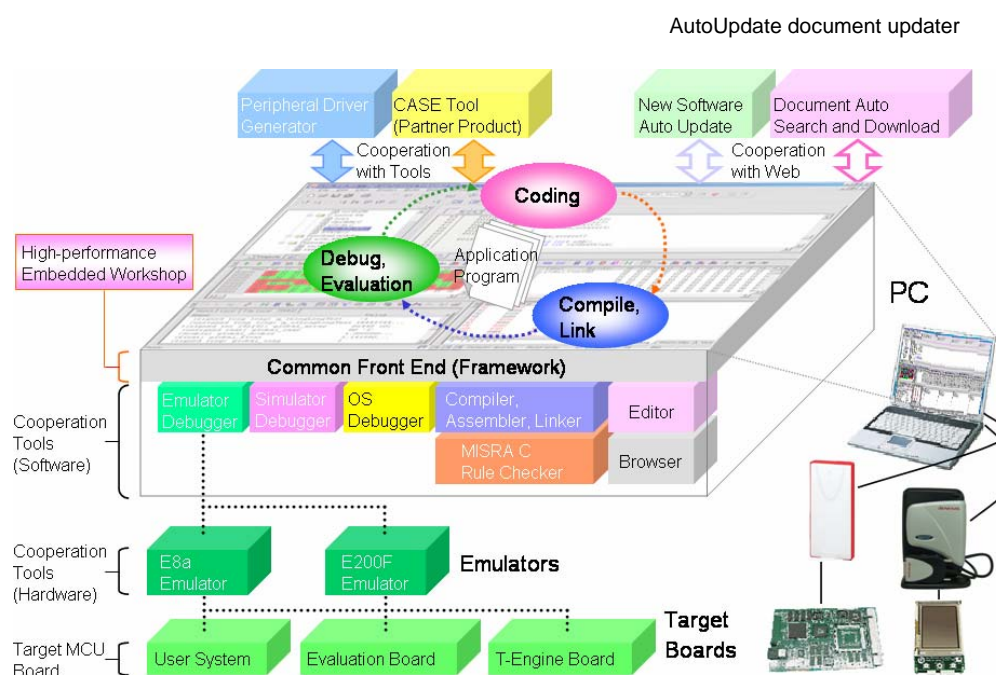


Figure 1 Configuration of the Renesas development environment

• High-performance Embedded Workshop (herein as *HEW*)

This is the Renesas integrated development environment. It is a common front end for using development tools.

In addition to an editor, browser, and project setup functionality, it contains a variety of features to perform the range of operations from coding to debugging.

• Compiler, assembler, and linker

This is a compiler made by Renesas, to extract maximum performance from the RX family.

The compiler, assembler, and linker are called the *toolchain* together. When used with HEW's build functionality, this toolchain can be used to create programs that run on microcomputers.

Note: Using the toolchain to create, from a source program, a program that runs on a microcomputer is called a *build*.

• Simulator

This allows RX operation to be simulated and evaluated on a computer. In addition to breaks, traces, and other debugging functionality, this also provides interrupt functionality and timer functionality.

The gray parts of Figure 1 indicate Renesas development tools other than those in the compiler package.

Additional installation can be performed to extend the functionality that can be used with HEW, as shown in Figure 1. For example, the E20 emulator can be installed as shown in E20 emulator control for Linked tools (software) in Figure 1, so that HEW can be used to perform evaluation with the E20 emulator.

The Renesas development environment, centered around the High-performance Embedded Workshop (HEW) IDE, enabled the creation of simple yet advanced development environments.

2. Installation

Insert the product CD-ROM, and follow the instructions in the installer. If HEW is already installed, the installer overwrites the existing HEW instance.

Even when HEW is already installed, the products started by the installation manager (Figure 2) during installation can be created in a folder other than that of the HEW environment.



Figure 2 Installation Manager

3. Setting Up a Project

The following uses RX610 as an example to explain how to set up a project.

3.1 Starting high-performance embedded workshop

From the Windows **Start** menu, choose **High-performance Embedded Workshop**. As shown in Figure 3, a welcome dialog box is displayed.

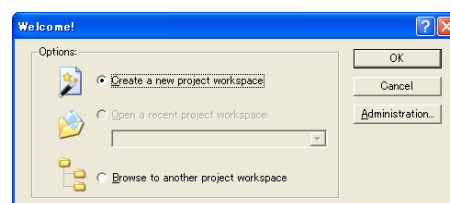


Figure 3 Welcome dialog box

Note: The contents displayed in the **Start** menu may differ depending on the state of tool installation.

Note: When multiple HEW environments are set up from the installation manager, the active HEW is started.

3.2 Creating a workspace

3.2.1 Workspaces and projects

HEW is based on *workspaces* and *projects*.

• Workspaces

This is the most significant management unit for creating programs using HEW. At least one project is required in each workspace, and this project is automatically created when the workspace is created. Note that a workspace can also have multiple projects.

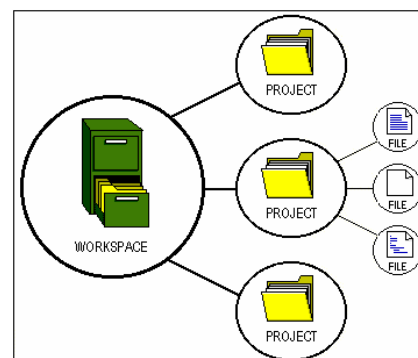


Figure 4 Workspaces and projects

- **Projects**

When programs are created, special functionality may be spun into a library, and layered in modules. In this case, projects for use as libraries can be created and added.

3.2.2 Creating a workspace

Click the **OK** button shown in Figure 3 to display the New Project Workspace dialog box. As shown in Figure 5, select **Application** for **Project type**, and fill in **Workspace name**.

When changing the project name, fill in **Workspace name** first, and then change **Project name**.

Note: If the specified directory already exists, the workspace cannot be created.

Note: When installing multiple compiler packages, specify **RX** for **CPU type**.

Note: When using the RX-family C/C++ compiler package, select **Renesas RX Standard** for **Toolchain**.

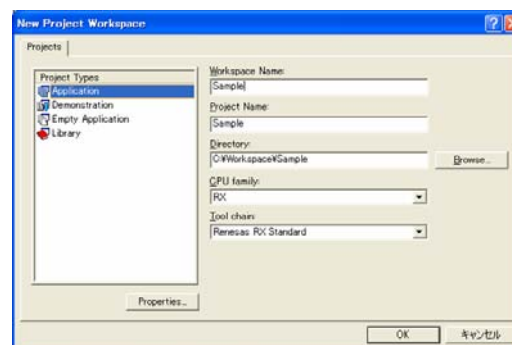


Figure 5 “New project workspace” dialog box

3.2.3 Selecting the target MCU

Select the toolchain version used and target MCU. As shown in Figure 6, select **RX600** for **CPU series** and **RX610** for **CPU type**.

Note: If the desired MCU is not displayed, update the revision of the compiler package to the latest version. The MCU may be added.

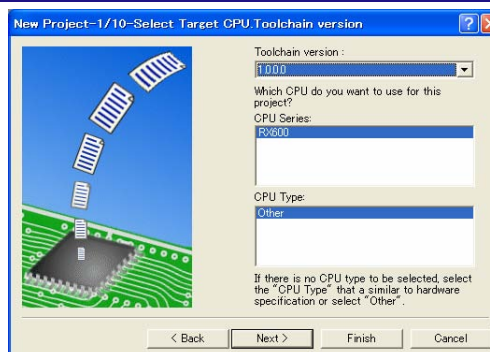


Figure 6 Selecting a target MCU

3.2.4 Selecting MCU options

In the Options dialog box shown in Figure 7 and Figure 8, select the common options for the compiler, assembler, and linker. The selected options can also be changed after the project is set up. Leave the settings as they are, and click **Next**.

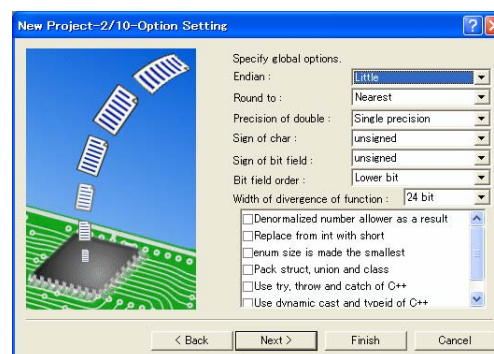


Figure 7 Selecting MCU options (1)

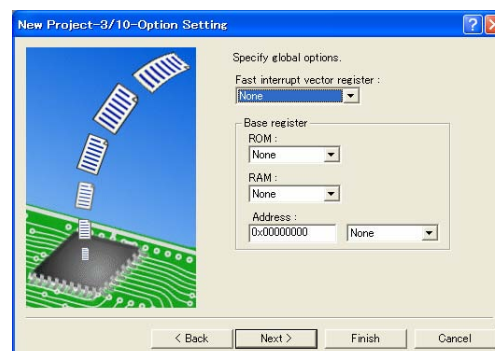


Figure 8 Selecting MCU options (2)

3.2.5 Setting up auto-generated files

In the Generated Files dialog box shown in Figure 9, set the files to be automatically generated by HEW.

- **Using the I/O library**

Specify the number of streams for standard input and output.

Note: When standard input and output streams are used, the low-level interface routines must be implemented. For details about how to implement low-level interface routines, see the compiler manual.

- **Using heap memory**

Select this to use the malloc, realloc, and calloc functions and C++ (new operators).

- **Generating the main() function**

Choose from assembly, C, or C++ languages. C is used by default.

- **I/O register definition file**

This generates the MCU header files that can be used with C/C++.

- **Generating the hardware setup function**

This generates the Hardware Setup function for performing initial MCU settings.

Note: The initial MCU settings differ depending on the user system.
Have the user implement the auto-generated Hardware Setup function.

Here, choose **C/C++ source file** for **Generate hardware setup function**, and click **Next**.

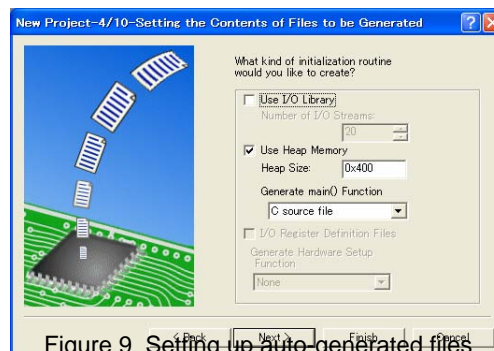
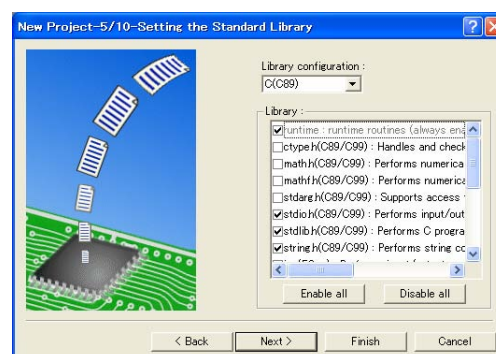


Figure 9 Setting up auto-generated files



3.2.6 Selecting standard libraries

In the Standard Library dialog box shown in Figure 10, select the standard libraries to be used. The libraries can also be selected after the project is set up. Leave the settings as they are, and click **Next**.

3.2.7 Setting up the stack area

In the Stack Area dialog box shown in Figure 11, specify the user stack size and interrupt stack size.

The initial value of the stack pointer is set near the last area in the RAM for each MCU. The stack pointer value and stack size can also be changed after the project is set up. Leave the settings as they are, and click **Next**.

Note: Slow access to the stack area can severely degrade performance. We recommend setting this to an internal RAM area to which access is fast.

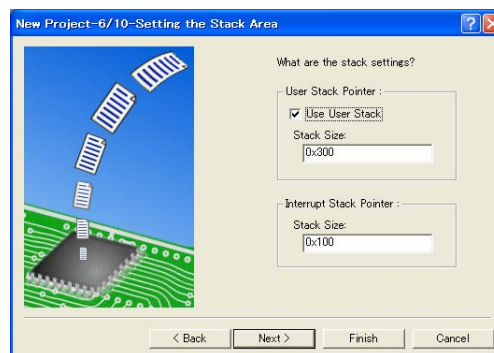


Figure 11 Setting up the stack area

3.2.8 Setting a reset vector

In the Vector dialog box shown in Figure 12, specify whether to generate a reset vector. The optimum setting will be performed for each MCU. Leave the settings as they are, and click **Next**.

This completes setup for auto-generated files.

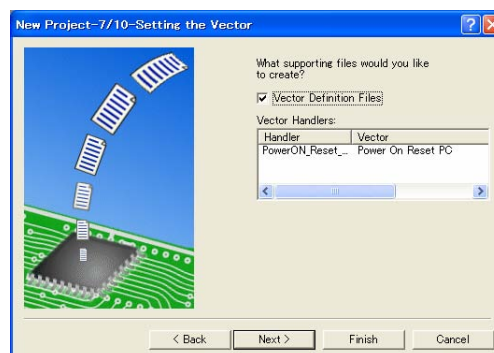


Figure 12 Setting a reset vector

3.2.9 Selecting a debugging target

In the Debugger dialog box shown in Figure 13, select whether to perform debugging in the generated programs and other targets. The simulator and emulator conforming to the RX family selected in 3.2.3 (Figure 6) are displayed automatically (the emulator is automatically displayed only when the corresponding emulator product is installed). Select **RX600 Simulator**, and click **Next**.

Note: Multiple targets can also be selected. In this case, debugging can be performed for one of the targets selected from the debugging session, as discussed in Section 5.1.

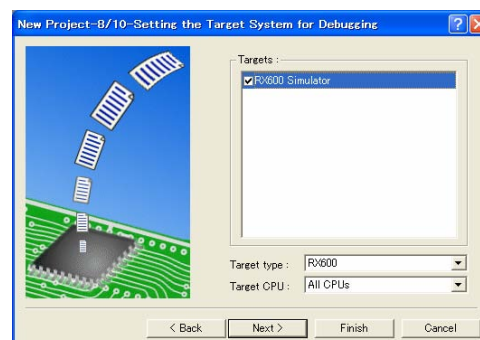


Figure 13 Selecting a debugging target

3.2.10 Selecting debugger options

In the Debugger Options dialog box shown in Figure 14, set the operations for the target specified in Figure 13. There is no particular reason to change these settings. Leave the settings as they are, and click **Next**.

Note: If multiple targets are selected, an options dialog box is displayed for each target. There is no need to change any of these settings.

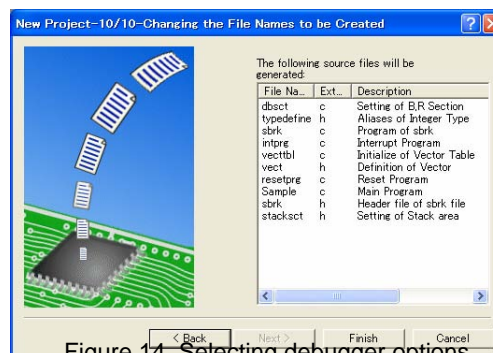


Figure 14 Selecting debugger options

3.2.11 Completing workspace creation

This concludes the settings needed to create a workspace. A list of the names of the files automatically generated by HEW is displayed in the Generated File Names dialog box shown in Figure 15, along with a description of each file.

Click the **Finish** button shown in Figure 15 to display the Overview dialog box shown in Figure 16. This allows you to check detailed information about the workspace set up in this chapter. If the **Save as summary...** checkbox shown at the bottom of Figure 16 is selected, the details displayed in the Overview dialog box are saved as a ReadMe.txt file in the project folder (see 3.3 for details).

Note: The information saved in ReadMe.txt is that from when the project was set up. It does not contain any information changed after project setup.

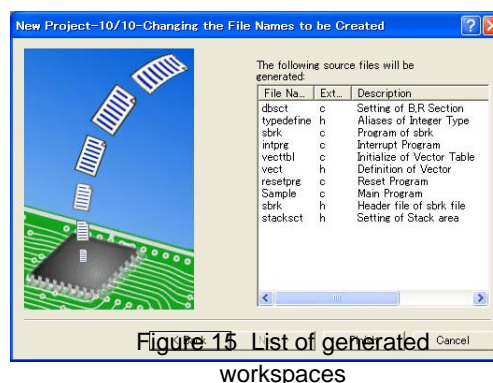


Figure 15 List of generated workspaces

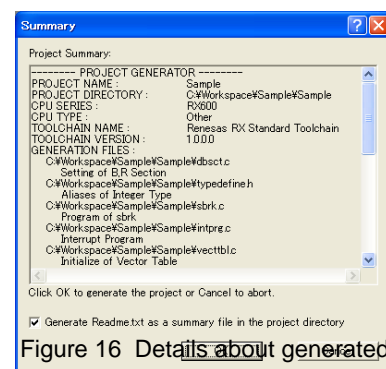


Figure 16 Details about generated files

Click the **OK** button shown in Figure 16 to start HEW with the workspace open, as shown in Figure 17. The terms used in Figure 17 are as follows:

- **Toolbar**

This is a collection of buttons that facilitate each operation.

- **Workspace window**

This displays the projects contained in the workspace, and a list of each the files contained in each project. Double-click these files to display their contents in the editor window. The workspace window consists of four tabs, which in addition to the above, can be used to set up browser functionality and test functionality.

- **Editor window**

This allows the contents of source files to be displayed and edited.

Since a C-source-level debugger is used, debugging can be performed easily by editing the source program, rebuilding, and then debugging.

- **Output window**

This displays the build results and execution results.

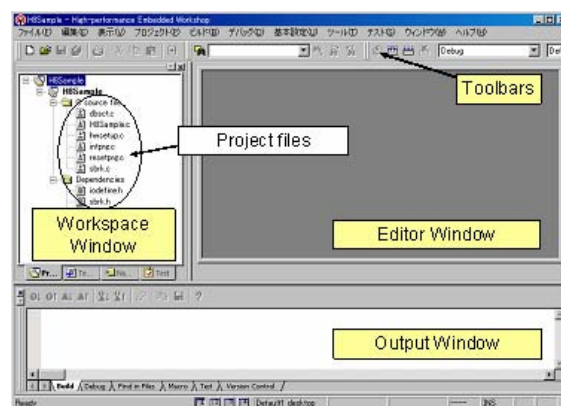


Figure 17 Completing workspace creation

3.3 Setting up folders in a workspace

Figure 18 shows the configuration of the auto-generated folders and files in 3.2. The auto-generated source files are kept in the project folder.

Note: The workspace and project folders also contain the settings files (*.hws, *.hwp, and *.hsf) automatically generated by HEW. Do not manually edit these settings files.

Note: User programs can be created in any folder, and then added to a project. We recommend adding any new folders below the workspace folder.

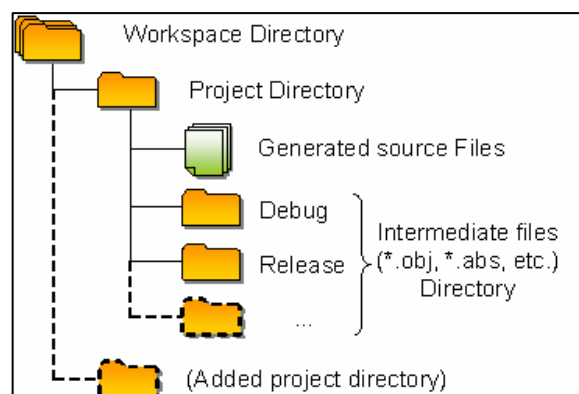


Figure 18 Configuration of auto-generated folders and files

4. Performing a Build

4.1 About builds

The process of using the toolchain to create, from a source program, a program that can run on a microcomputer is called a *build*. Figure 19 shows the build process.

- **Setting up standard libraries**

With the RX-family C/C++ compiler package, optimization settings can also be set for libraries. As such, the libraries specified in 3.2.6 are set up during the initial build.

Note: Library setup may take a while.

Note: Libraries are recreated whenever settings are changed for build options (4.2), in the **CPU** tab or **Standard libraries** tab.

- **Compiling**

Start the compiler, and compile the C/C++ source programs (*.c and *.cpp).

- **Assembling**

Start the assembler, and assemble the source programs in assembly language (*.src and *.s).

- **Linking**

Link the intermediate files (*.obj) and libraries (*.lib), to create program that runs on a microcomputer. S-type files (*.mot) and HEX files (*.hex) can also be output for embedding in the product (such as for ROM writers) at the same time as the load module files (*.abs) used during debugging.

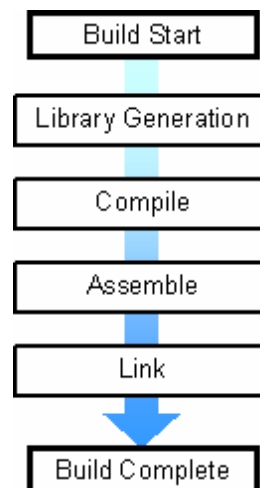


Figure 19 Build flow

4.2 Build options

4.2.1 Setting build options

From **Build**, choose **RX Standard Toolchain...** to display the RX Standard Toolchain dialog box shown in Figure 20. All options settings for the toolchain can be performed in this dialog box.

- **Compiler tab**

This sets the compiler options.

- **Assembler tab**

This sets the assembler options.

- **Optimization linker tab**

This sets the linker options.

Note: Since the RX linker made by Renesas contains optimization functionality, it is also called an *optimization linker*.

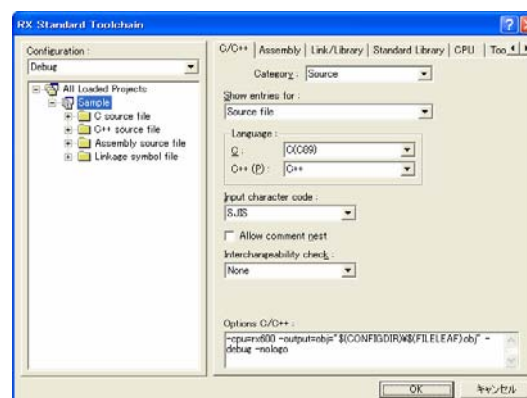


Figure 20 Setting build options

- **Standard libraries** tab

This sets the options during library setup.

- **RTOS** tab

This sets the RTOS options.

- **CPU** tab

This allows the common MCU option (3.2.4) to be set through builds.

- **General** tab

This sets the Hew options.

4.2.2 Option settings targets

The options settings common to all project files are set during the initial settings, but options can also be set for each file.

Individual options can be set for the settings target shown in the left-hand side of the window shown in Figure 21, which also shows a sample specification. The **Ctrl** and **Shift** keys can also be used to select the same operations for multiple files at once.

In this case, since no particular options are changed, close the options dialog box.

4.3 Configuration

The options set in 4.2 are saved to facilitate program evaluation with various options settings.

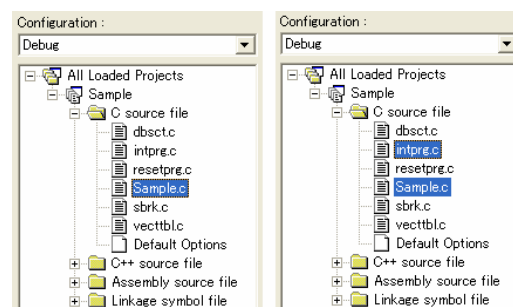
HEW also allows project options settings information to be registered and managed in units called *configurations*.

Initially, the Debug and Release configurations exist. The build results for each configuration (options settings) are saved and managed in individual folders, as shown in Debug and Release in Figure 18. Configurations for various options settings can be generated and saved to make evaluation of the option settings easier. Configurations can be switched from **Configuration** in the **Build** toolbar, as shown in Figure 22.

Note: In the initial Release configuration, the debugging information output in the Debug configuration is suppressed.
As such, the Release configuration and Debug configuration output the same program.

4.4 Performing a build

To perform a build, either press the **F7** key, or choose **Build** from the **Build** menu under the **HEW** menu, or click the **Build** button at the top of the toolbar, as shown in Figure 23.



(single file specified) (multiple files specified)

Figure 21 Selecting options settings targets

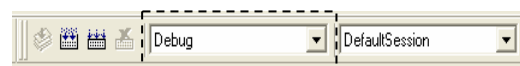


Figure 22 Build toolbar



Figure 23 Build button

After the build, the program has been successfully created, unless any errors are displayed in the output window shown in Figure 17.

Note: The Options dialog box shown in Figure 20 must be closed when a build is performed. The Options dialog box can be closed to make sure that options have been set for the toolchain.

Note: If any error or warnings are output, press the **F8** key to display the corresponding line in the editor (when a compiler or assembler error occurs). Likewise, place the cursor on the error or warning line in the output window and press **F1** to display help about the error, as shown in Figure 24.

This concludes automatic generation of programs through HEW.



Figure 24 Error help (displayed by the F1 key)

5. Debugging

This chapter explains how to transfer programs created in the previous chapter to a target, for debugging.

5.1 Debugging sessions

Just as option settings can be managed as a *configuration*, HEW allows debugging environments to be registered and managed as a *debugging session*. Debugging sessions can be switched from the **Debugging session** drop-down list in the **Build** toolbar, as shown in Figure 25. The information registered and saved in a debugging session is as follows:

- **Connection target**

This specifies the target for which program debugging is to be performed. The simulator or emulator (for an installed product) can be selected. Initially, the target selected in 3.2.9 is registered.

- **Debugging target program**

This specifies the load module (such as *.abs or *.mot). Initially, the *.abs file for the selected configuration is registered.

- **Window information during debugging**

This saves a snapshot of the information used for each window during debugging (such as the register, memory, and watch windows). This means that debugging sessions can be registered and saved, and then selected when debugging is restarted, to instantly restore the previously saved debugging environment, as shown in Figure 26.



Of the two drop-down lists, the right one is the debugging session

Figure 25 Build toolbar

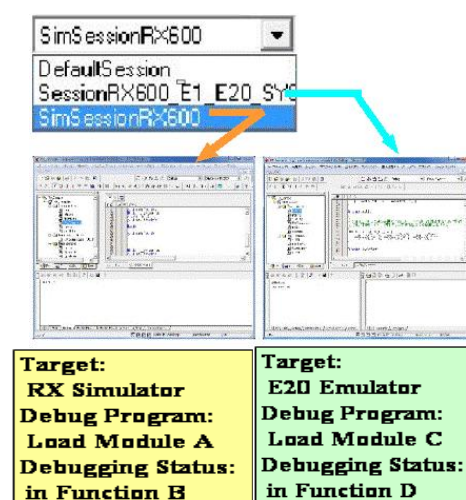


Figure 26 Restoring a debugging environment

5.2 Target connections

To connect to a target, select the desired target from the **Debugging session** drop-down shown in Figure 25. Since the **SimSessionRX600** debugging session was registered by specification during project setup (3.2.9), choose the **SimSessionRX600** debugging session. When a debugging session is selected, HEW starts the software linked to the target (Figure 1), and establishes a connection. When the simulator is started, the Simulator Settings dialog box shown in Figure 27 is displayed. The Simulator Settings dialog box is used to set the simulator CPU configuration and supported peripheral module functionality. Peripheral module functionality can be used for any module indicated by **Module Name** (the compare match timer and interrupt controller in Figure 27).

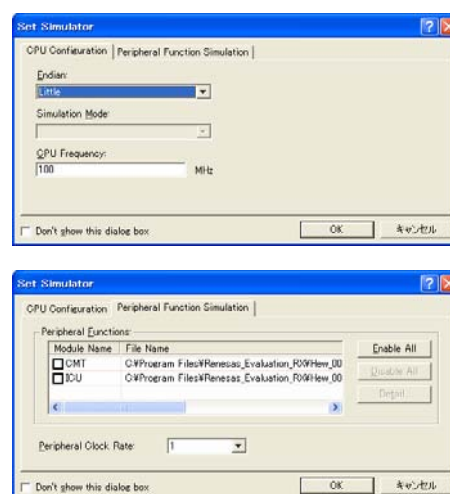


Figure 27 Simulator Setup dialog box

Here, leave everything unselected, and then click the **OK** button. The toolbar group for debugging is added to the toolbar at the top of HEW as shown in Figure 28, to allow simulation (program execution) and debugging.

Note: When an emulator is used, emulator-specific settings may need to be set during connection. For details about emulator connection settings, see the emulator users' manual or application notes.

The connection to the target (simulator) is complete once **Connected** is displayed in **Debug** tab of the output window, as shown in Figure 17.



Figure 28 Toolbars displayed before and after the simulator is run (top: before, bottom: after)

5.3 Downloading debugging target programs

Next, download the debugging target program. During connection with the target, the **Download modules** folder and load module file shown at the top of Figure 29 are added to the workspace window, as shown in Figure 17. Double-click the load module to download it, and display the downward arrow indicating a completed download in the load module file, as shown at the bottom of Figure 29.

Note: The number shown in the right-hand side of the load module file is the offset address. Set **Offset address** to its initial value of 0.

Note: When an emulator is used and download is performed to an external memory area, the port, bus controller, and other settings need to be performed to ensure access to the corresponding area.

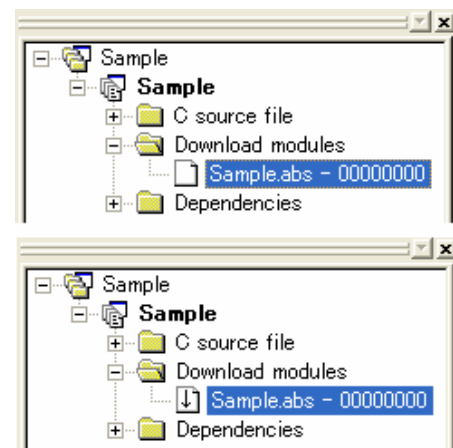


Figure 29 Downloading load modules (top: before download, bottom: after download)

5.4 Executing and debugging programs

The previous section completed the preparation necessary for program execution and debugging. This section explains how to actually perform simple execution and debugging.

5.4.1 Reset

First, perform reset to check that branching is performed to the reset vector.

To perform reset, click the **CPU reset** button in the toolbar, as shown in

Figure 30. The power-on reset address and initial stack pointer value are obtained from the reset vector, and then execution stops at the start of the PowerON_Reset_PC function, as shown in Figure 31.

Here, HEW stops the program, and automatically displays the corresponding source program if a corresponding source line exists.

Note: If the source program is not displayed, check the following:

- During build, specify debugging options for both the compiler and linker if no debugging options are specified.
- Perform build again if the source file is not stored in the same location as during build.



Figure 30 CPU Reset button

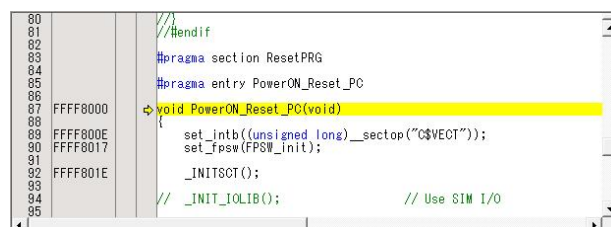


Figure 31 Status after reset

5.4.2 Changing the source program display mode

The **Display mode** can be switched by clicking the button in the upper-left of the editor window, as shown in Figure 32.

- **Source display mode** button (left side, Figure 32)

This is the mode for displaying and editing the source program, as shown in Figure 31.

Note: Source programs can only be edited in the source display mode.

- **Mixed display mode** button (center, Figure 32)

This mode displays the source program and assembly code together. This is convenient for analyzing object code details, since the assembly code associated with the source program is displayed, as shown in Figure 33.

- **Reverse assembly display mode** button (right side, Figure 32)

This mode is only for reverse assembly display. This allows the object code contents to be checked, even when no library or source program exists.

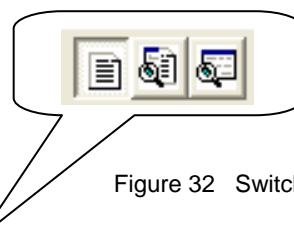


Figure 32 Switch display mode button

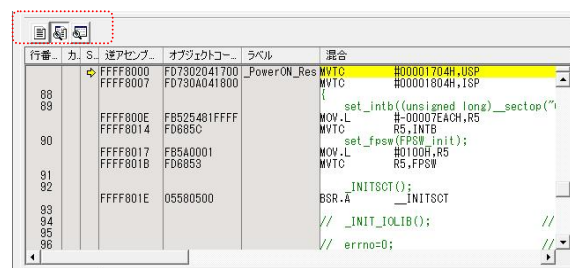


Figure 33 Mixed display mode

5.4.3 Step execution (step in and step over)

Click the step execution button to perform step execution for the program.



Figure 34 Step execution buttons
(left: Step in, right: Step over)

- **Step in** button (left side, Figure 34)

This performs step execution. Click this at the start of a function to stop at the next line within the function.

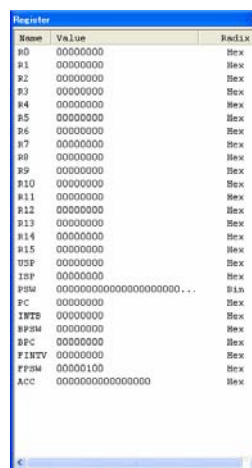
- **Step over** button (right side, Figure 34)

This performs step execution. Click this at the start of a function to stop at the next line after the function executes.

Note: If no source program in which to stop exists during step execution (such as for a library), the window switches the Reverse Assembly window.

5.4.4 Referencing and settings register values

Click the **Register** button shown in Figure 35 to display the Register window (left side, Figure 36). For example, step execution can be performed with the register window open to check that the value of the program counter has been updated. The register value can also be changed. Double-click the register to be changed to display the Register Value Settings dialog box (right side, Figure 36).



| Name | Value | Radix |
|-------|-----------------------------|-------|
| R0 | 00000000 | Hex |
| R1 | 00000000 | Hex |
| R2 | 00000000 | Hex |
| R3 | 00000000 | Hex |
| R4 | 00000000 | Hex |
| R5 | 00000000 | Hex |
| R6 | 00000000 | Hex |
| R7 | 00000000 | Hex |
| R8 | 00000000 | Hex |
| R9 | 00000000 | Hex |
| R10 | 00000000 | Hex |
| R11 | 00000000 | Hex |
| R12 | 00000000 | Hex |
| R13 | 00000000 | Hex |
| R14 | 00000000 | Hex |
| R15 | 00000000 | Hex |
| DSP | 00000000 | Hex |
| PSH | 000000000000000000000000... | Hex |
| PC | 00000000 | Hex |
| INTB | 00000000 | Hex |
| RPSW | 00000000 | Hex |
| RPC | 00000000 | Hex |
| RINTV | 00000000 | Hex |
| RPSW | 00001100 | Hex |
| ACC | 000000000000000000000000 | Hex |



Figure 35 Register button



Figure 36 Referencing and setting register values

5.4.5 Referencing function and variable definitions

The workspace window consists of four tabs, as shown in Figure 17.

Choose the third tab from the left, **Navigation**. A list of the functions and global variables within the program is displayed. In **C Functions**, double-click **main(void)** as shown in Figure 37. The source program for the definition section of the main function is displayed.

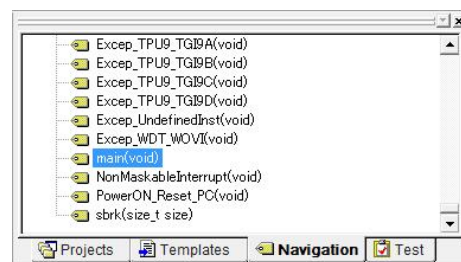


Figure 37 Referencing function and variable definitions

5.4.6 Setting up breakpoints

Next, perform breakpoint settings.

When the operations in 5.4.5 are performed, since the mouse cursor is placed at the start of the main function, either press the **F9** key or double-click the **S/W (software)** **breakpoint** column in the editor window. The ● symbol is displayed in the **S/W breakpoint** column, as shown in Figure 38.

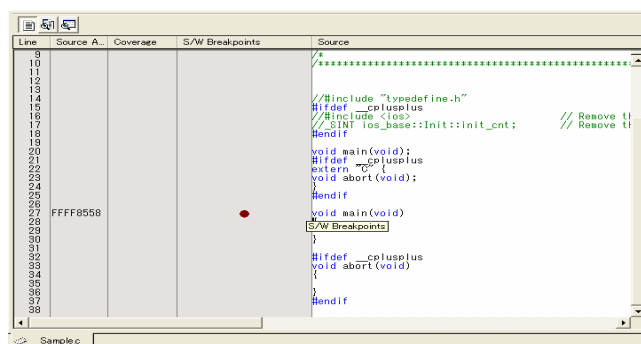


Figure 38 Setting up breakpoints

5.4.7 Executing a program

To execute a program, press the **F5** key, or click the **Execute** button, as shown in Figure 39. The program is executed from the stop location, and then stops wherever a breakpoint is set.



Figure 39 Execute button

Note: If the source program address is the same over multiple lines as shown in Figure 39, the breakpoint and stop location (indicated by an arrow) may be displayed in different locations.



Figure 40 Watch button

5.4.8 Watching

To browse and edit variables, click the **Watch** button, as shown in Figure 40. The Watch window is displayed as shown in Figure 41, so register the name of the variable to browse or edit. Also, the variables in the editor window can be registered by double-clicking to highlight them, and then dragging and dropping the highlighted variables directly into the Watch window.

Note: When projects are set up as shown in Chapter 3, variables cannot be defined in the source files for auto-generated projects.

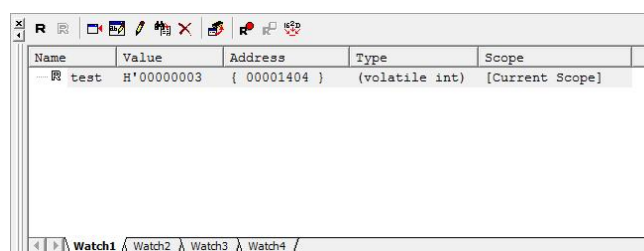


Figure 41 Watch window

5.4.9 Revising and re-evaluating programs

Programs that need to be revised during debugging can be easily re-evaluated as follows:

- **Edit the corresponding source in the Source window mode, as explained in 5.4.2.**
- **Press the F7 key or click the Build button to perform the build again, as explained in 4.4.**
- **Download the program again according to the procedures in 5.3 (if the program is set to be automatically downloaded after build is performed again, it does not need to be downloaded manually).**

Website and Support

- Renesas Electronics Website
<http://www.renesas.com/>
- Inquiries
<http://www.renesas.com/inquiry>

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

Revision Record

[illegible]

Notice

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



SALES OFFICES

Renesas Electronics Corporation

<http://www.renesas.com>

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.

2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited

1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH

Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.

7th Floor, Quantum Plaza, No.27 ZhichunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.

Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited

Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852-2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.

7F, No. 363 Fu Shing North Road Taipei, Taiwan, R.O.C.
Tel: +886-2-8175-9600, Fax: +886-2-8175-9670

Renesas Electronics Singapore Pte. Ltd.

1 harbourFront Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.

Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.

11F., Samik Lavied* or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141