RENESAS Tool News

RENESAS TOOL NEWS on July 1, 2006: 060701/tn1

The High-performance Embedded Workshop (IDE) Revised to V.4.01.00

We have revised the High-performance Embedded Workshop from V.4.00.03 to V.4.01.00.

---

1. **Versions to Be Updated**
   The High-performance Embedded Workshop* V.2.2 through V.4.00.03

   * The High-performance Embedded Workshop is bundled with the software products such as compilers that it manages. To check for the version number of your High-performance Embedded Workshop, open the Help menu and select the About High-performance Embedded Workshop command.

2. **Descriptions of Revision**
   2.1 **Functions Introduced**

   (1) Macro-Recording Support
   Supported the function of recording operations concerning project management, builds, and debugging to generate commands on the High-performance Embedded Workshop command line as well as the function of executing recorded commands.

   (2) Test Support
   Supported the function of saving the contents of the window selected by the Create New Test Image dialog box as a test-image file.
   In addition, supported the function of comparing the contents of test-image files with the results of execution of macros generated with the Macro-Recording Support

function and of conventional command batches of the High-performance Embedded Workshop.

2.2 **Functions Improved**

(1) The Map window

- The two types of window, the Map Section Information and the Map Symbol Information window are available.
- Information on sections can be added, modified, and deleted.
- Because the functions of searching for, filtering, and sorting information on symbols have been introduced, any symbols can easily be searched for.
- Selecting a symbol among the symbol information list and double- clicking it make a jump to the line that defines the symbol in a source file displayed in the Editor window.
- The contents of the Map window can be printed out.

(2) The Output window

- Introduced the tool bar that enables you easily to perform operations of referencing lines containing information on build errors and of clearing the contents of the window.
- Because icons appear on the message lines of the errors detected at build, each message can easily be searched for.
- The colors of text in the error, warning, and information message lines can be customized.

(3) The Workspace window

- Sub-folders can be created in workspaces, which makes file management on workspaces much easier.
- All the folders provided in Windows(R) Explorer can be drugged and dropped into workspaces. So more than one source file in a folder is registered at one time.

(4) The Editor window

- Column headers introduced under the Toolbar of the window. They are useful for adjusting the width of each column as you like.

When the Disassembly mode is switched to the Source or Mixed mode under the condition that addresses displayed in the Disassemble mode are out of the address range of the source file that is opened, the dialog box for choosing whether to open another source file or not can be displayed.

- An option to prohibit editing all the files opened in the Editor window while the debugger is operating is introduced.
- The address line pointed to by the program counter (PC) can be highlighted in color (yellow by default).
- Jumps can be made from the source codes displayed in the Editor window to #define directives, functions, and lines defining classes.
- The Smart Editor function can be used for the C source files as well as the C++ source files.
- The names of #define macros, functions, and classes defined in a project can be displayed in a list by opening a pop-up window.

(5) The Disassembly window

- Column headers introduced under the Toolbar of the window. They are useful for adjusting the width of each column as you like.
- The address line pointed to by the program counter (PC) can be highlighted in color (yellow by default).
- A command for refreshing the contents of the window introduced. Using this command updates the contents of the window at any time.

(6) Selection of editors

Introduced an option to select an external editor or the editor in the High-performance Embedded Workshop to open files.

(7) Builds

- Introduced an option not to auto-update the dependencies of files in the project. This option prohibits the background processing of file updates.
- The intermediate and output files of a build in each configuration directory of a project can be deleted. This is the same function as "clean" of the Make utility.
- The current directory where Make files are executed can be set using a dialog box.

- The path names used in a Make file created by the High-performance Embedded Workshop can be selected out of the following three:
  - All relative paths
  - All absolute paths
  - Relative paths for the directories under the Workspace directory and absolute paths for the others (Whether to support this function or not depends on the specifications of tool chains.)
- For error messages dispatched by external tools as well as the tool chains managed by the High-performance Embedded Workshop, jumps are made to the lines in the source file where the errors have been detected provided that the format of error messages are registered.

(8) Download of modules

- The Download A New Module and Recently Downloaded Modules added to the File menu.
- Modules can be downloaded only by double-clicking their names in the Download modules folder displayed by clicking the Projects tab of the Workspace window.

(9) Debugging

- Introduced an option to display the source files containing main functions in the Editor window after downloading modules.
- Introduced an option to execute the program up to the beginning of its main function after resetting the CPU.
- Introduced an option not to step in the library functions with no debug information (printf and others). (Whether to support this function or not depends on the specifications of debuggers.)
- Introduced an option not to access the memory on the target until the execution of batch files of commands are ended when executing the batch files automatically after connecting the target with the High-performance Embedded Workshop.
- Introduced an option not to access any memory locations except those within the range displayed in the Disassembly window.

(10) Creation of workspaces

When a workspace of a debugger-dedicated project (Debugger only - xxxxxx) is opened, no build menus are displayed.
Also the Editor, Search, Templates, Bookmarks, Default Window, and standard tool bar are not displayed by default.

(11) Extension of TCL command syntax
The High-performance Embedded Workshop command can be used as a parameter of TCL command "set". Using the High-performance Embedded Workshop command enables referencing the results of execution of any High-performance Embedded Workshop commands and assigning them to variables.

In the example below, the result of execution of the High- performance Embedded Workshop command "memory_display 300 10" is assigned to variable "md_300_10", and the result of the assignment can be referenced.
Example:

```
set md_300_10 [memory_display 300 10]
```

As a result of the above extension, the result of execution of "memory_display 300 10" is not displayed for the following descriptions:

```
for {set i 0} {$i < 2} {incr i} {
   memory_display 300 10
}
```

To output the results of High-performance Embedded Workshop commands to the Command Line window, use TCL command "puts".

```
for {set i 0} {$i < 2} {incr i} {
   puts [memory_display 300 10]
}
```

(12) Time to displaying static variables (only when the C/C++ compiler packages and the emulator debuggers for the SuperH RISC engine, H8, H8S, and H8SX families used)
Even when load modules with a large amount of debug information are under debugging, time to expand and

display variables in the Watch window becomes much shorter.

(13) The Watch window (only when the C/C++ compiler packages and the emulator debuggers for the SuperH RISC engine, H8, H8S, and H8SX families used)
- When the contents of the Watch window are updated with Auto Update being valid, the updated values can be outputted to files at the same time.
- When modules are reloaded to the simulator debugger or emulator, the status of expansion and the settings of Auto Update before reloading modules can be maintained if no changes are made to the types of the symbols registered with the Watch window.

2.3 **Problems Fixed**

The following seven known problems have been fixed:

(1) On using the jumping function
For details see RENESAS TOOL NEWS Doc. No. RSO-HEW_1-051016D, published on October 16, 2005.

(2) On displaying an incorrect coverage value in a Statistic column of the Code Coverage window
For details see RENESAS TOOL NEWS Doc. No. RSO-HEW-051201D, published on December 1, 2005.

(3) On loading load modules created in the Intel HEX format
For details see RENESAS TOOL NEWS Doc. No. RSO-HEW-060201D, published on February 1, 2006.

(4) On sorting watch points in the ASM Watch window
For details see RENESAS TOOL NEWS Doc. No. RSO-HEW_1-060216D, published on February 16, 2006.

(5) On using the RAM monitoring function in the ASM Watch window
For details see RENESAS TOOL NEWS Doc. No. RSO-HEW_2-060216D, published on February 16, 2006.

(6) On displaying the address or size field in the ASM Watch window
For details see RENESAS TOOL NEWS Doc. No. RSO-HEW_3-060216D, published on February 16, 2006.

(7) On editing the value of the I/O register

For details see RENESAS TOOL NEWS Doc. No. RSO-HEW-060316D, published on March 16, 2006.

In addition to the above, the following newly found problems have been fixed:

(8) On the FILE_LOAD command
If the protection by the password of Windows(R) is started* while a batch file containing FILE_LOAD commands is executed, no FILE_LOAD commands can be executed.

* The state where the Security dialog box of Windows(R) is opened. (This dialog box is opened by pressing the Ctrl+Alt+Del keys.)

(9) On adjusting the width of windows
The width of windows may not be adjusted by the mouse.
On the other hand, it may be modified if the height of windows is adjusted.

(10) On the order of source files displayed in the Notebook
When source files are displayed in the Notebook*, changes may be made to the order of displayed files if the workspace is reopened.

* When "Show files in notebook" check box displayed by clicking the Editor tab of the Options dialog box is checked.
This dialog box is opened by selecting the Options command of the Setup menu.

(11) On displaying the source files saved by an external editor

The High-performance Embedded Workshop is forcibly terminated if source files are opened in the Mixed mode in the Editor window of the High-performance Embedded Workshop, the source files are edited and saved by an external editor, and then the display mode in the Editor window is switched from Mixed to Source.

(12) On software breakpoints
When a session read, information on the software breakpoints that have been set at the last saving of the

session may incorrectly be restored.

(13) On the order of linking files
If the version of the tool chain used for a project is changed, files are linked in alphabetical order.

(14) On dialog boxes displayed in multi-display configuration
When a workspace created, the same dialog box may be opened on two or more screens depending on how to manipulate the dialog box.

(15) On selecting a menu during build
The Build File "file-name" command in pop-up menu in the Editor window becomes effective during build.

(16) On the Watch window (only when the C/C++ compiler packages and the emulator debuggers for the SuperH RISC engine, H8, H8S, and H8SX families used)
When the value of a watch item edited and both the initial value in the Edit Value dialog box for editing the value and that of in-place editing begin with D'-, no negative numbers in decimal can be inputted unless D'-s are replaced with -D's.

3. How to Update Your Product

Free-of-charge update is available if you are using the product concerned.
To update yours online, download the update program from the Download Site and execute it.

4. Notices

(1) No components except the High-performance Embedded Workshop (for example, C compilers, emulators, etc.) are affected by this update.

(2) In order to update the High-performance Embedded Workshop bundled with the C/C++ compiler package for the SuperH RISC engine family, the compiler package must be updated to V.7.1.03 or V.7.1.04 in advance; then update the High-performance Embedded Workshop to V.4.01.00.
So, the C/C++ compiler package for the H8, H8S, and H8SX families must be updated to V.5.0.05 or V.5.0.06 before updating the High-performance Embedded

Workshop.

[Disclaimer]
The past news contents have been based on information at the time of publication. Now changed or invalid information may be included. The URLs in the Tool News also may be subject to change or become invalid without prior notice.