To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

User's Manual

High-performance Embedded Workshop V.4.03

User's Manual

Renesas Microcomputer Development
Environment System

Renesas Electronics
www.renesas.com

Rev.1.00 2007.04

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

For inquiries about the contents of this document or product, fill in the text file the installer generates in the following directory and email to your local distributor.

\SUPPORT\Product-name\SUPPORT.TXT

Renesas Tools Homepage   http://www.renesas.com /tools

Introduction

The High-performance Embedded Workshop is a powerful development environment for embedded applications targeted at Renesas micro-controllers. The main features are:

- A configurable build engine that allows you to set-up compiler, assembler and linker options by using GUI.
- An integrated text editor with user customizable syntax coloring to improve code readability.
- A configurable environment, which allows you to run your own tools.
- An integrated debugger, which allows you to build and debug in the same application.
- Version control support.

The High-performance Embedded Workshop has been designed with two key aims; firstly to provide you, the user, with a set of powerful development tools and, secondly, to unify and present them in a way that is easy to use.

About This Manual

This manual describes the High-performance Embedded Workshop system. This manual describes information on the basic "look and feel" of the High-performance Embedded Workshop and customizing the High-performance Embedded Workshop environment and detail the build and the debugging functions common to the High-performance Embedded Workshop products. The figures in this document shows the HEW operating with the SuperH™ RISC engine family debugging platform.

The term "debugging platform" refers to an emulator or simulator/debugger that operates with the integrated development environment "High-performance Embedded Workshop".

For details on the debugging platform, see the user's manual or help information packaged with the emulator or simulator.

This manual does not intend to explain how to write C/C++ or assembly language programs, how to use any particular operating system or how best to tailor code for the individual devices. These issues are left to the respective manuals.

Document Conventions

This manual uses the following typographic conventions:

Convention	Meaning
[Menu->Menu Option]	'->' is used to indicate menu options (for example, [File->Save As]).
FILENAME.C	Uppercase names are used to indicate filenames.
"enter this string"	Used to indicate text that must be entered (excluding the "" quotes).
Key + Key	Used to indicate required key presses. For example, CTRL+N means press the CTRL key and then, whilst holding the CTRL key down, press the N key.

Trademarks

Microsoft, MS-DOS, Windows are registered trademarks of Microsoft Corporation.

Visual SourceSafe is a trademark of Microsoft Corporation.

IBM and AT are registered trademark of International Business Machines Corporation.

All brand or product names used in this manual are trademarks or registered trademarks of their respective companies or organizations.

Table Of Contents

1. Overview 1

1.1 Workspaces, projects and files 1

1.2 Main window 1

1.2.1 Title bar 2

1.2.2 Menu bar 2

1.2.3 Toolbars 2

1.2.4 Workspace window 5

1.2.5 Editor window 7

1.2.6 Output window 8

1.2.7 Status bar 11

1.3 Help system 11

1.4 Launching the HEW 11

1.5 Creating a new workspace 12

1.6 Opening a workspace 13

1.7 Using old workspaces 13

1.8 Saving a workspace 13

1.9 Closing a workspace 14

1.10 Exiting the HEW 14

1.11 Component system overview 14

1.12 Overview of Configurations and Sessions 14

1.13 Overview of Macro-Recording Support facility and Test Support facility 18

1.13.1 Example of test procedures 21

1.13.2 Step 1: Recording a macro 23

1.13.3 Step 2: Editing a macro (viewing records) 24

1.13.4 Step 3: Playing a macro 26

1.13.5 Step 4: Creating a test suite 26

1.13.6 Step 5: Editing a test suite 26

1.13.7 Step 6: Creating a test image file 28

1.13.8 Step 7: Modifying the program before the test 30

1.13.9 Step 8: Viewing the test result (unmatched) 31

1.13.10 Step 9: Modifying the program back and executing the test again 31

1.13.11 Step 10: Viewing the test result (matched) 32

2. Build Basics 33

2.1 The build process 33

2.2 Configuring the Projects tab of the Workspace window 33

2.3 Project files 36

2.3.1 Adding files to a project 37

2.3.2 Drag and drop of files and folders 37

2.3.3 Removing files from a project 39

2.3.4 Excluding a project file from build 40

2.3.5 Including a project file in build 40

2.4 User folders in the workspace 41
2.5 File extensions and file groups 42
2.5.1 Associating an application with a file group 42
2.5.2 Creating a new file extension and file group 44
2.5.3 Creating a new file extension 45
2.6 Setting build options 45
2.7 Build configurations 46
2.7.1 Selecting a build configuration 46
2.7.2 Adding a new build configuration 47
2.7.3 Removing a build configuration 47
2.8 Building a project 47
2.8.1 Building individual files 47
2.8.2 Building a project 47
2.8.3 Building multiple projects 48
2.8.4 Stopping tool execution 49
2.8.5 Deleting intermediate and output files produced in building 49
2.8.6 Configuring the Build tab of the Output window 50
2.8.7 Controlling the content of the Build tab of the Output window 52
2.8.8 Displaying out of date files in the Workspace window 52
2.9 File dependencies 53
2.10 Configuring the Workspace window 56
2.11 Inserting a project into the workspace 57
2.12 Setting the current project 58
2.13 Specifying dependencies between projects 59
2.14 Removing a project from the workspace 60
2.15 Relative projects paths in the workspace 60
3. Advanced Build Features 61
3.1 The build process revisited 61
3.1.1 What is a build? 61
3.2 Creating a custom build phase 62
3.3 Ordering build phases 66
3.3.1 Build Order tab 66
3.3.2 Build File Order tab 69
3.3.3 File Mappings tab 69
3.4 Setting custom build phase options 71
3.4.1 Options tab 71
3.4.2 Output Files tab 72
3.4.3 Dependent Files tab 72
3.5 Controlling the build 73
3.6 Logging build output 74
3.7 Changing toolchain version 75
3.8 Generating a makefile 75
3.9 Using a makefile inside the HEW system 77

3.10 Customizing the HEW linkage order 78
4. Using the Editor 81
4.1 Editor window 81
4.2 Working with multiple files 82
4.3 Standard file operations 82
4.3.1 Creating a new file 82
4.3.2 Editing a file 83
4.3.3 Saving a file 83
4.3.4 Opening a file 84
4.3.5 Closing files 85
4.4 Searching and navigating through files 86
4.4.1 Finding text 86
4.4.2 Finding text in multiple files 86
4.4.3 Replacing text 87
4.4.4 Jumping to a specified line 88
4.5 Bookmarks 89
4.6 Printing a file 90
4.7 Configuring text layout 90
4.7.1 Page set-up 90
4.7.2 Changing tabs 91
4.7.3 Auto indentation 92
4.8 Splitting a window 92
4.9 Changing the editor font 93
4.10 Syntax coloring 93
4.10.1 Changing text colors 94
4.10.2 Creating new keywords 94
4.10.3 Enabling/disabling syntax coloring 95
4.11 Templates 96
4.11.1 Defining a template 96
4.11.2 Deleting a template 98
4.11.3 Inserting a template 98
4.12 Brace matching 98
4.13 Setting the read-only attribute for a file 99
4.14 Preventing modification of files while debugging 99
4.15 Managing the editor columns 100
4.16 Showing/hiding the column header 100
4.17 Opening a file within the editor 101
4.18 Tooltip watch 101
4.19 Evaluate an expression 102
5. Tools Administration 103
5.1 Tool locations 103
5.2 HEW registration files 104
5.3 Registering a component 105

5.4 Unregistering a component 106
5.5 Viewing and editing component properties 106
5.6 Uninstalling a component 108
5.7 Technical support 109
5.8 Using On-Demand components 110
5.9 Custom project types 111
6. Customizing the Environment 113
6.1 Customizing the toolbars 113
6.2 Customizing the Tools menu 115
6.3 Using custom placeholders 116
6.4 Using the workspace and project log facilities 117
6.5 Configuring the help system 118
6.6 Keyboard shortcut customization 119
6.7 Scope of a control in the setup 120
6.7.1 Scope of a control in the Customize dialog box 120
6.7.2 Scope of a control in the Options dialog box 121
6.8 Specifying workspace options 121
6.8.1 Open last workspace at start-up 121
6.8.2 Restore files on opening a workspace 121
6.8.3 Display workspace information on opening a workspace 122
6.8.4 Save workspace before executing any phases 122
6.8.5 Prompt before saving a workspace 122
6.8.6 Prompt before saving session 123
6.8.7 Enable auto-backup facilities 123
6.8.8 Change the default directory for new workspaces 124
6.9 Using an external editor 124
6.10 Customizing the font in your views 126
6.11 Using the virtual desktop 127
7. Version Control 129
7.1 Selecting a Version Control System 130
7.2 Importing and exporting a set-up 131
8. Using the Custom Version Control System 133
8.1 Defining Version Control menu options 133
8.1.1 System-defined menu options 134
8.1.2 User-defined menu options 135
8.2 Defining Version Control commands 136
8.3 Specifying arguments 137
8.4 Specifying comments 137
8.5 Executable return code 137
8.6 Specifying file locations 138
8.7 Specifying file locations example 139
8.8 Specifying environment 140
8.9 Controlling execution of a Version Control System 141

8.10 Specifying a user name and password 141
8.11 Usage example of the Custom Version Control System 142
8.11.1 Connecting the HEW with RCS 142
8.11.2 Connecting the HEW with CVS 147
9. Using Visual SourceSafe 155
9.1 Attaching Visual SourceSafe to a workspace 155
9.2 Visual SourceSafe commands 157
9.2.1 Adding files to Visual SourceSafe 158
9.2.2 Removing files from Visual SourceSafe 159
9.2.3 Getting read-only copy of files from Visual SourceSafe 159
9.2.4 Checking out writable copy of files from Visual SourceSafe 160
9.2.5 Checking in writable copy of files into Visual SourceSafe 161
9.2.6 Viewing the status of files in Visual SourceSafe 162
9.2.7 Undoing a check out command in Visual SourceSafe 163
9.2.8 Viewing the history of files in Visual SourceSafe 164
9.3 Visual SourceSafe command options 164
9.4 Version Control setup 165
9.5 Specifying file locations 165
9.6 Adding Visual SourceSafe commands 167
10. Sharing Projects by Network Facilities 169
10.1 Sharing projects by network facilities in a PC with Windows XP Service Pack 2 170
10.2 Enabling network facilities to share projects 171
10.3 Setting the administrator user's password 172
10.4 Adding new users to the system 172
10.5 Changing your password 174
10.6 Note on using the network facilities to share projects 174
11. Comparing Files 175
11.1 Opening the Difference window 175
12. Navigation Facilities 179
12.1 C function and #define navigation component 181
12.2 C++ navigation component 181
12.3 Jump to a definition from the editor 183
12.4 Drag and drop navigation items 184
12.5 Smart edit capability in the HEW editor 185
13. Map 187
13.1 Managing section settings 188
13.1.1 Opening the Map Section Information window 188
13.1.2 Entering/exiting the edit mode 190
13.1.3 Adding a section group 191
13.1.4 Adding a section 192
13.1.5 Adding an overlay group 192
13.1.6 Editing a selected item 193
13.1.7 Setting the primary section 193

13.1.8 Setting a memory map 194
13.1.9 Automatically allocating the memory resource 194
13.1.10 Printing out the section settings tree 194
13.1.11 Viewing unallocated areas 195
13.1.12 Viewing sections of size 0 195
13.1.13 Viewing the source code for the address 195
13.1.14 Printing out the section list 195
13.2 Viewing symbols 195
13.2.1 Opening the Map Symbol Information window 196
13.2.2 Printing out the map list 197
13.2.3 Finding symbols 197
13.2.4 Filtering the symbol information 198
13.2.5 Viewing the source code for the address 199
13.2.6 Printing out the symbol information 199
14. Using the Command Line 201
14.1 Opening the Command Line window 201
14.2 Specifying a command file 202
14.3 Executing a command file 203
14.4 Stopping command execution 203
14.5 Specifying a log file 203
14.6 Starting or stopping logging 203
14.7 Entering a full path to the file 203
14.8 Pasting a placeholder 204
14.9 Selecting all the window contents 204
14.10 Copying the selection onto the clipboard 204
14.11 Cutting out the selection to the clipboard 204
14.12 Pasting the contents of the clipboard 204
14.13 Clearing the window contents 204
14.14 Undoing the last operation 204
14.15 Checking brace matching 205
15. Using the Macro-Recording Support Facility 207
15.1 Macro menu and toolbar 207
15.2 Using the Macro dialog box 208
15.3 Importing a macro file with existing macros 210
15.4 Recording a macro 210
15.5 Functions that can be recorded into macro files 211
15.5.1 Recordable functions (common to all HEW products) 211
15.5.2 Recordable functions (dependent on the debugging platform) 213
15.6 Playing a macro 216
15.7 Editing a macro 216
15.8 Assigning a macro 216
15.9 Configuring the Macro tab of the Output window 217
16. Using the Test Support Facility 219

16.1 Creating a test suite 219
16.2 Opening and closing test suites 220
16.3 Editing a test suite 220
16.4 Adding tests to the test suite 222
16.5 Creating a test image file 223
16.6 Functions that can be saved as test-image data into test-image files 224
16.6.1 Functions that can be saved into test-image files (common to all HEW products) 225
16.6.2 Functions that can be saved into test-image files (dependent on the debugging platform) 228
16.7 Comparing a test image file 234
16.8 Running tests 235
16.9 Using the test browser 236
16.10 Configuring the Test tab of the Workspace window 236
16.11 Configuring the Test tab of the Output window 237
17. Using the Debugger 239
17.1 Preparations for debugging 239
17.1.1 Compiling for debug 239
17.1.2 Selecting a debugging platform 239
17.1.3 Editing project configuration 250
17.1.4 Configuring the debugging platform 250
17.1.5 Downloading modules 254
17.1.6 Debugger sessions 265
17.2 Viewing a Program 270
17.2.1 Opening the Editor window 271
17.2.2 Opening the Disassembly window 277
17.2.3 Looking at the current PC position 283
17.2.4 Highlighting the line at the PC 283
17.3 Operating memory 284
17.3.1 Opening the Memory window 284
17.3.2 Modifying memory contents 286
17.3.3 Selecting a memory range 287
17.3.4 Filling an area of memory with constant data 287
17.3.5 Copying an area of memory 288
17.3.6 Comparing the memory contents 289
17.3.7 Testing an area of memory 289
17.3.8 Saving memory contents in a text file 290
17.3.9 Finding a value in memory 290
17.3.10 Changing the display address 291
17.3.11 Changing the scroll area 291
17.3.12 Starting address to value of the register 292
17.3.13 Tracking the stack pointer position 292
17.3.14 Changing the program display position immediately after downloading 292
17.3.15 Updating the window contents 293
17.3.16 Disabling update of the window contents 293

17.3.17 Changing the data length 293
17.3.18 Changing the radix 293
17.3.19 Changing the code 293
17.3.20 Setting the layout 294
17.3.21 Changing the number of digits displayed 294
17.3.22 Switching display or non-display of measurement result 294
17.3.23 Saving an area of memory 295
17.3.24 Loading a memory area from a file 295
17.3.25 Splitting up the window display 296
17.3.26 Verifying a memory area 296
17.3.27 Customizing the current colors 296
17.4 Displaying memory contents as an Image 297
17.4.1 Opening the Image window 297
17.4.2 Automatically updating the window contents 299
17.4.3 Updating the window contents 300
17.4.4 Displaying the pixel information 300
17.5 Displaying memory contents as Waveforms 300
17.5.1 Opening the Waveform window 300
17.5.2 Automatically updating the window contents 302
17.5.3 Updating the window contents 302
17.5.4 Zoom-in display 302
17.5.5 Zoom-out display 302
17.5.6 Resetting the zoom display 302
17.5.7 Setting the zoom magnification 302
17.5.8 Setting the horizontal scale 302
17.5.9 Non-display of cursor 302
17.5.10 Displaying the sampling information 303
17.6 Looking at I/O memory 303
17.6.1 Opening the IO window 303
17.6.2 Expanding an I/O register display 304
17.6.3 Updating the window contents 305
17.6.4 Loading an I/O file 305
17.6.5 Printing the currently displayed contents 305
17.6.6 Saving the currently displayed contents 305
17.6.7 Modifying I/O register contents 305
17.7 Looking at registers 305
17.7.1 Opening the Register window 306
17.7.2 Changing the register display radix 306
17.7.3 Switching Register Bank 307
17.7.4 Setting the layout 307
17.7.5 Choosing a register to be displayed 308
17.7.6 Modifying register contents 309
17.7.7 Setting the flag value 309

17.7.8 Splitting up the window display 310
17.7.9 Saving register contents 310
17.7.10 Using register contents 310
17.7.11 Customizing the color of the changed values 310
17.8 Resetting the target MCU 310
17.9 Setting PC to the address at cursor 311
17.10 Initializing the debugging platform 311
17.11 Connecting/disconnecting the debugging platform 311
17.12 Executing your program 311
17.12.1 Continuing run 311
17.12.2 Running from reset 312
17.12.3 Running to cursor 312
17.12.4 Running from a specified address 312
17.12.5 Continuing execution to a main function at a reset 313
17.12.6 Single step 313
17.12.7 Multiple steps 314
17.13 Stopping your program 315
17.13.1 Halting execution 315
17.13.2 Standard PC breakpoints 315
17.14 Viewing the current status 316
17.15 Viewing the function call history 317
17.15.1 Opening the Stack Trace window 317
17.15.2 Viewing the source program 318
17.15.3 Specifying the view 318
17.16 Using an external debugger 318
17.16.1 Configuring the Hitachi Debugging Interface to integrate with HEW 319
17.16.2 Configuring the PD debugger to integrate with HEW 319
17.16.3 Configuring an external debugger to integrate with HEW 320
17.17 Synchronizing multiple debugging platforms 321
17.17.1 External HEW synchronization 321
17.17.2 Internal HEW synchronization 321
17.18 Debugging functions dependent on the debugging platform 323
17.18.1 Looking at labels 323
17.18.2 Elf/Dwarf2 support 326
17.18.3 Looking at variables 330
18. Technical Support 339
18.1 Check for updates 339
18.2 Creating a bug report 339

Reference 341

1. List of Menus 343
1.1 List of File Menu 343

1.2 List of Edit Menu....343
1.3 List of View Menu....344
1.4 List of Project Menu....345
1.5 List of Build Menu....345
1.6 List of Debug Menu....346
1.7 List of Setup Menu....347
1.8 List of Tools Menu....347
1.9 List of Test Menu....348
1.10 List of Window Menu....348
1.11 List of Help Menu....348
2. List of Windows....349
3. List of Commands....351
3.1 Command List (Alphabetically Order)....351
3.2 Command List (Listed as the Functions)....353
4. Regular Expressions....355
5. Placeholders....357
5.1 What is a placeholder?....357
5.2 Inserting a placeholder....357
5.3 Available placeholders....359
5.4 Placeholder tips....360
6. I/O File Format....361
7. Symbol File Format....363
8. Keyboard Shortcuts....365
9. Drag and Drop in the HEW Debugger....367
10. Using Labels to View Your Code....369
11. Integrated Toolbars in a Components View....371
12. To Build in Toolchain for HEW1.x....373
13. HMAKE User Guide....375
13.1 Command line....375
13.2 File syntax....375
13.3 Description blocks....376
13.4 Comments....377
13.5 Message commands....378

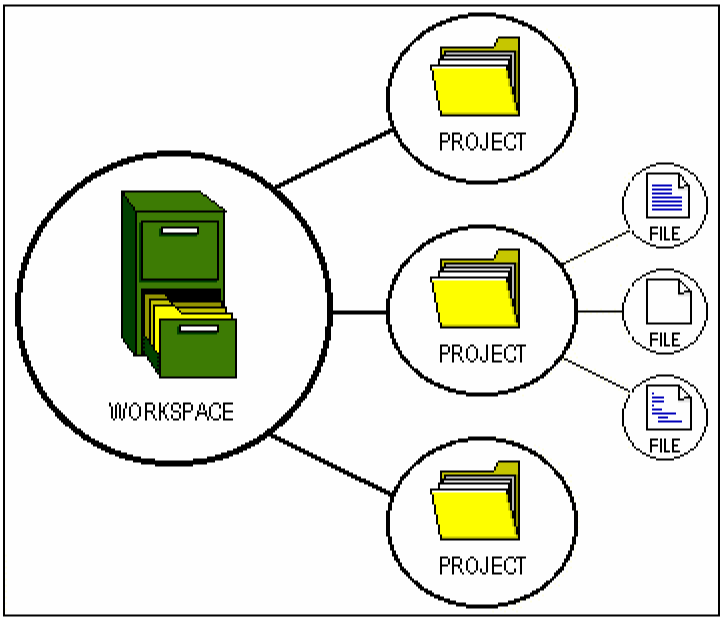
1. Overview

The functions for High-performance Embedded Workshop (HEW) version V.4.03.00 are explained in this manual.

This chapter describes the fundamental concepts of the HEW.

1.1 Workspaces, projects and files

Just as a word processor allows you to create and modify documents, the High-performance Embedded Workshop allows you to create and modify workspaces. A workspace can be thought of as a container of projects and, similarly, a project can be thought of as a container of project files. Thus, each workspace contains one or more projects and each project contains one or more files. The figure below illustrates this graphically:

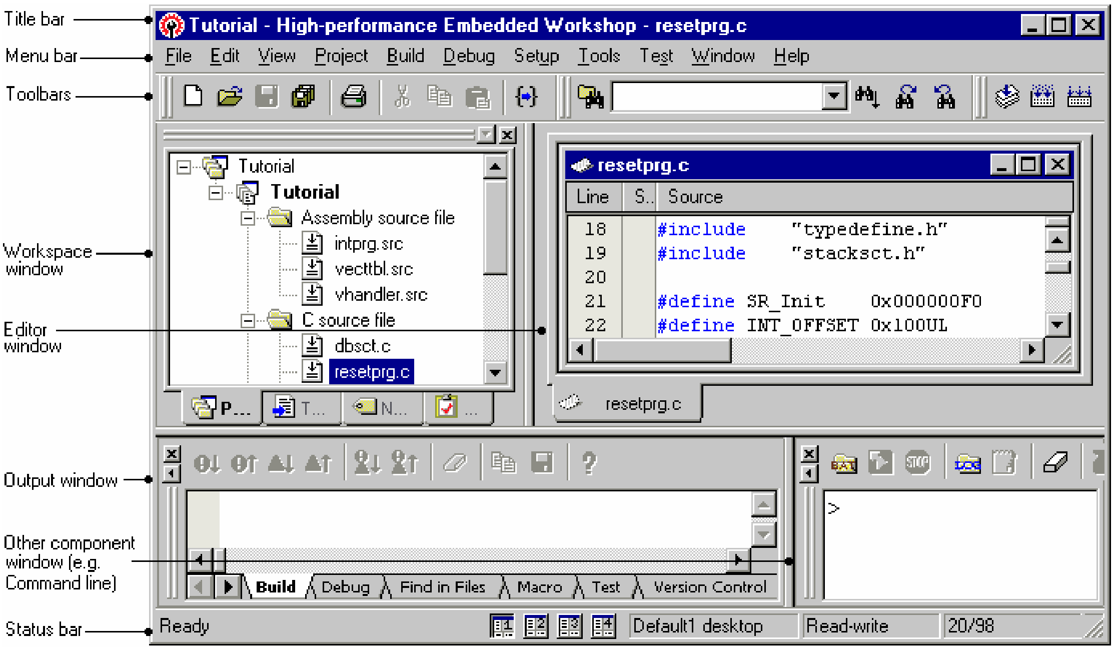


Workspaces allow you to group related projects together. For example, you may have an application that needs to be built for different processors, or you may be developing an application and library at the same time. Projects can also be linked hierarchically within a workspace, which means that when one project is built all of its 'child' projects are built first.

However, workspaces on their own are not very useful – we need to add a project to a workspace and then add files to that project before we can actually do anything.

1.2 Main window

There are three main windows: the workspace window, the editor window and the output window. The workspace window shows the projects and files that are currently in the workspace; the editor window provides file viewing and editing facilities; and the output window shows the results of a various processes (e.g. build, version control commands and so on).



```
#include     "typedefine.h"
#include     "stacksct.h"

#define SR_Init     0x000000F0
#define INT_OFFSET 0x100UL
```

1.2.1 Title bar

The title bar displays the name of current activate project and file. It also contains the standard **Minimize**, **Maximize** and **Close** buttons. Click the **Minimize** button to minimize the HEW on the Windows® task bar. Click the **Maximize** button to force HEW to fill the screen. Click the **Close** button to close the HEW (this has the same effect as selecting the [**File->Exit**] menu option, or pressing ALT+F4).

1.2.2 Menu bar

The menu bar initially contains eleven menus: **File**, **Edit**, **View**, **Project**, **Build** *, **Debug**, **Setup**, **Tools**, **Test**, **Window** and **Help**. All of the menu options are grouped logically under these headings. For instance, if you want to open a file then the **File** menu is where you will find the right menu option; if you want to set-up a tool then the **Tools** menu is the correct selection.

File Edit View Project Build Debug Setup Tools Test Window Help

*. If you use a debug-only project "Debugger only - xxxxxx" created by HEW V.4.01 or a later version, the **Build** menu will not be displayed by default. The **Build** menu is displayed, however, if the debug-only project "Debugger only - xxxxxx" has been created by HEW earlier than V.4.01.

1.2.3 Toolbars

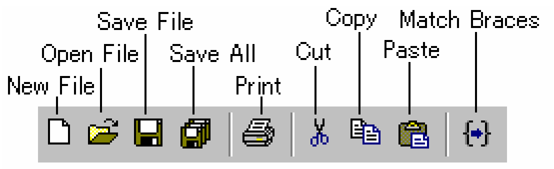
The toolbars provide a shortcut to the options that you will use the most often. There are ten default toolbars: **Editor**, **Search**, **Templates**, **Bookmarks**, **Default Window**, **Standard**, **Version Control**, **Map**, **Macros**, **Debug**, and **Debug Run** (as shown in the figures below).

With the default session, the HEW does not initially display buttons of **Version Control** and peripheral functions on the toolbar.

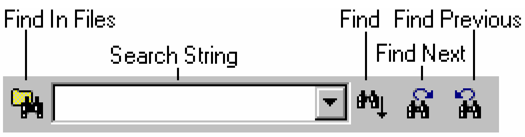
If you use a debug-only project “Debugger only – xxxxxx” created by HEW V.4.01 or a later version, **Editor**, **Search**, **Templates**, **Bookmarks**, **Default Window**, and **Standard** will not be displayed by default.

Toolbars can be created, modified and removed via the [**Tools->Customize**] menu option (see section 6.1, Customizing the toolbars, for further information).

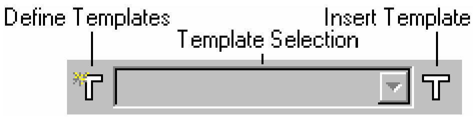
Editor toolbar



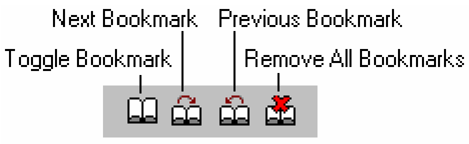
Search toolbar



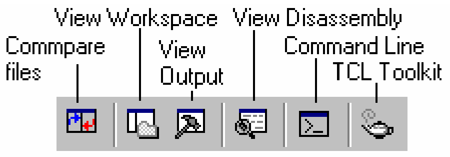
Templates toolbar



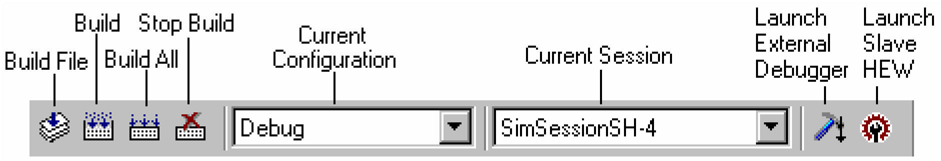
Bookmarks toolbar



Default Window toolbar

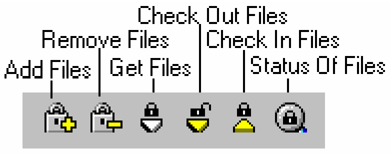


Standard toolbar

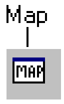


Version Control toolbar

This toolbar is only available when a version control tool is being used in the current project.

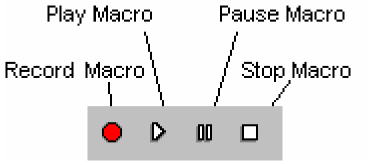


Map toolbar



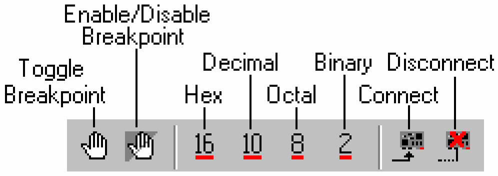
Macros toolbar

You can add a customized button, to which a created macro has been assigned, on the right to the standard macro buttons on the toolbar as shown below.



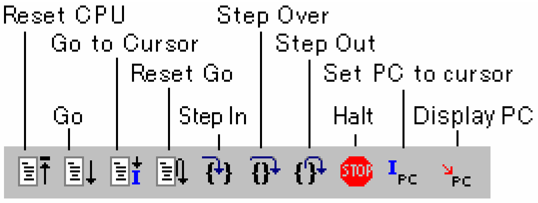
Debug toolbar

This toolbar is only available when a session is being used which has a target attached.

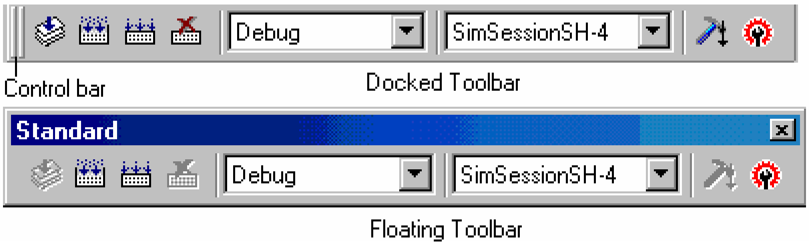


Debug Run toolbar

This toolbar is only available when a session is being used which has a target attached.



When the Standard toolbar is docked, it has a Control bar as shown in the figure below. If you want to move the docked Standard toolbar, click and drag its Control bar to the new location. The figure below shows the Standard toolbar when it is docked and also when it is floating.



To dock a toolbar

Select one of the following operations:

- Double-click on the title bar of a floating toolbar, **OR**
- Drag the title bar of a floating toolbar and draw it toward an edge of a docked window, menu bar, toolbar or the HEW main frame, on whose edge you would like to dock the window, until the shape of the floating bar changes.

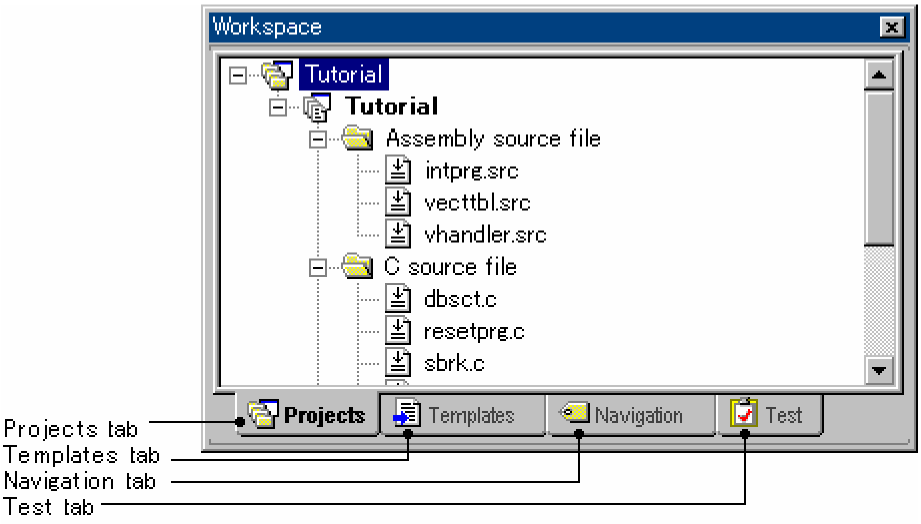
To float a toolbar

Select one of the following operations:

- Double-click on the control bar of a docked toolbar, **OR**
- Drag the control bar of a docked toolbar and draw it away from the edge of the HEW main frame and from an edge of the other docked windows, menu bar or toolbars.

1.2.4 Workspace window

The **Workspace** window has four panes (**Projects**, **Templates**, **Navigation**, and **Test** tab).



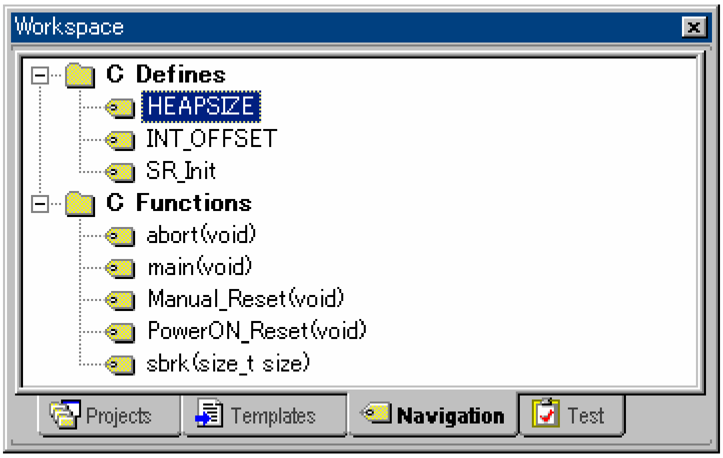
- **Projects tab**

Allows you to show the current workspace, projects and files. You can quickly open any project file or dependent file by double-clicking on its icon. See section 2.2, Configuring the Projects tab of the Workspace window, for more information on the Projects tab.

- **Templates tab**

Allows you to display template settings. See section 4.11, Templates, for more information about a template.

- **Navigation tab**



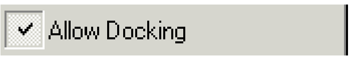
Allows you to provide jumps to various textual constructs within your project's files. What is actually displayed within the **Navigation** tab depends upon what components are currently installed. The figure above shows C macro definitions (C Defines) and C functions (C Functions) conforming to the ANSI standard. See Chapter 12, Navigation Facilities, for more information about a navigation.

- **Test tab**

Allows you to setup or view test suites as part of the test-support facility. See Chapter 16, Using the Test Support Facility, for more information about the test-support facility.

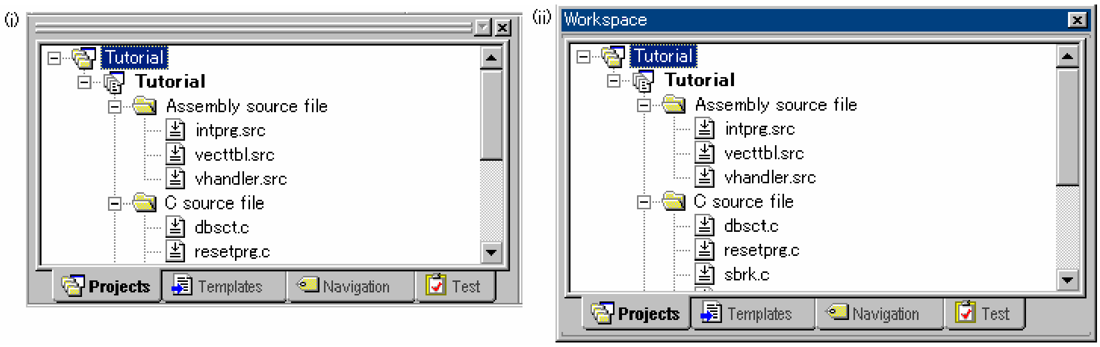
To allow the Workspace window or the Output window docking

Right-click anywhere inside the **Workspace** window or the **Output** window. Then a pop-up menu will be displayed.



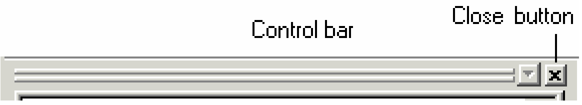
If the **Allow Docking** option is checked, docking is allowed. Otherwise, docking is not allowed. Select the **Allow Docking** option to check or un-check it.

When the **Allow Docking** option is checked, you can dock a window, toolbar or menu bar to the edge of the HEW main window or to the edge of another docked window. You can also float them 'above' the other HEW windows or outside the HEW main window. Figure (i) below shows a docked "Workspace" window, and figure (ii) below shows a floating "Workspace" window.



When the **Workspace** window or the **Output** window is docked, it has a control bar as shown below.

If you want to move a docked window, click and drag its control bar to the new location.



To dock the Workspace window or the Output window

1. Ensure that the **Allow Docking** option is checked on the window's pop-up menu.
2. Select one of the following operations:
   - Double-click on the title bar of a floating window, **OR**
   - Drag the title bar of a floating window and draw it toward an edge of a docked window, menu bar or toolbar, or the HEW main frame, on whose edge you would like to dock the windows.

To float the Workspace window or the Output window

1. Ensure that the **Allow Docking** option is checked on the window's pop-up menu.

2. Select one of the following operations:

- Double-click on the control bar of a docked window, **OR**
- Drag the control bar of a docked window and draw it away from the edge of the HEW main frame and from an edge of the other docked windows, menu bar or toolbar.
- Drag the control bar of a docked window while pressing the "CTRL" key.

To hide the Workspace window or the Output window

Select one of the following operations:

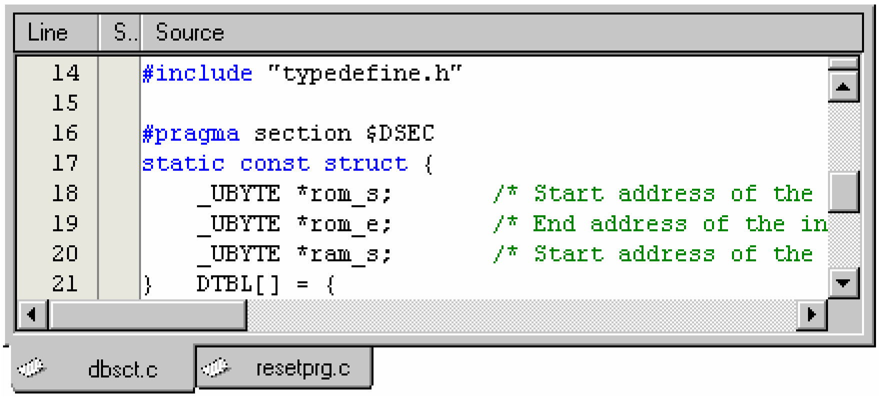
- Click on the close button, which is located in the top right corner of the window, **OR**
- Right-click anywhere inside a floating window and select the **Hide** option on the pop-up menu.

To show the Workspace window or the Output window

Select the [**View->Workspace**] or [**View->Output**] menu option respectively.

1.2.5 Editor window

The editor window is where you will work with the files of your project. The HEW allows you to have many files open at one time, to switch between them, to arrange them and to edit them in whichever order you want to. By default, the editor window is displayed in a notebook style. This means that each file has a separate tab associated with it to aid in navigating between files (see the figure below).



```
#include "typedefine.h"

#pragma section $DSEC
static const struct {
    _UBYTE *rom_s;      /* Start address of the
    _UBYTE *rom_e;      /* End address of the in
    _UBYTE *ram_s;      /* Start address of the
}   DTBL[] = {
```

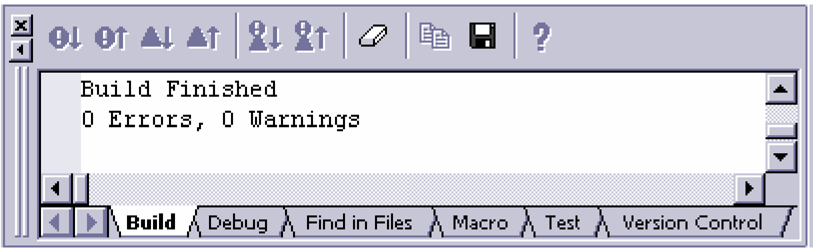
The editor contains a gutter (column) on the left-hand side of the window. The standard column allows the user to configure the position of bookmarks and software breakpoints (*) quickly and easily. If you are unsure what purpose a column has or what the information it is displaying is if you place the mouse pointer over the column a tool tip (pop-up window) is displayed showing its identity.

The editor window can be customized via the **Format Views** dialog box, which can be invoked via the [**Setup->Format Views**] menu option. This dialog box allows you to configure fonts, colors, tabs and so on for the editor window. It also allows the user to change the look of other views, which have been installed by HEW. If you would prefer to use your favorite editor rather than the HEW internal editor then specify your alternative in the **Options** dialog box, which can be invoked via the [**Setup->Options**] menu option. For further details on how to use the editor, see Chapter 4, Using the Editor.

*. Allows you to set software breakpoints only when the debugging platform is connected to the session.

1.2.6 Output window

The **Output** window, by default, has six tabs (**Build**, **Debug**, **Find in Files**, **Macro**, **Test**, and **Version Control** tab) on display.



```
Build Finished
0 Errors, 0 Warnings
```

Build tab

The Build tab shows the output from any build process (e.g. compiler, assembler and so on). If an error is encountered in a source file, an icon and the error message will be displayed in the Build tab, along with the source file name and line number.

If you click on a toolbar button or pop-up menu item associated with display of error messages, the line of the error message will be highlighted and the editor will show the source code of that line (if the line has any source code).

Double-clicking the line of an error message will also show the source code in the editor.

When you attempt an operation to display an error or warning, the status bar shows this error or warning message.

Right-clicking displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up Menu Options	Toolbar button	Function
-	❗↓	Highlights the next occurrence of error and shows the source code in the editor.
-	❗↑	Highlights the previous occurrence of error and shows the source code in the editor.
-	⚠↓	Highlights the next occurrence of warning and shows the source code in the editor.
-	⚠↑	Highlights the previous occurrence of warning and shows the source code in the editor.
Display next Error/Warning/Info	❗⚠↓	Displays the editor that generated the next build error or warning.
Display previous Error/Warning/Info	❗⚠↑	Displays the editor that generated the previous build error or warning.
Help	?	Shows the help information about the line.
Go to Error/Warning/Info	-	Goes to the associated source line.
Clear Window	⌫	Clears the contents of the window.
Save	💾	Saves the contents of the window into a text file.
Copy	📋	Copies the selected contents onto the Windows® clipboard.
Toolbar display	-	Shows or hides the toolbar.
Customize toolbar	-	Customizes toolbar buttons.

The **Build** tab also shows an icon corresponding to the error message output by build execution.

Icon Name	Icon	Error Message Level
Build Error	❗	Error
Build Warning	⚠	Warning
Information	ⓘ	Information

Debug tab

Shows the output from any debugger process. Any debug component that needs to display information will send its output to this window.

Right-clicking displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up Menu Options	Toolbar button	Function
Clear Window	🧽	Clears the contents of the window.
Save	💾	Saves the contents of the window into a text file.
Copy	📋	Copies the selected contents onto the Windows® clipboard.
Toolbar display	-	Shows or hides the toolbar.
Customize toolbar	-	Customizes toolbar buttons.

Find in Files tab

Shows the results of the last **Find in Files** action. To activate find in files, select the [Edit->Find in Files] menu option, or click the Find In Files toolbar button. For further details on using **Find in Files**, see section 4.4.2, Finding text in multiple files.

Right-clicking displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up Menu Options	Toolbar button	Function
Go to Occurrence	-	Go to the associated source line.
Clear Window	🧽	Clears the contents of the window.
Save	💾	Saves the contents of the window into a text file.
Copy	📋	Copies the selected contents onto the Windows® clipboard.
Toolbar display	-	Shows or hides the toolbar.
Customize toolbar	-	Customizes toolbar buttons.

Macro tab

Shows the current records of macros (macro-recording support facility). You can view information such as HEW command-line commands recorded into a HEW macro file from execution of [**Tools->Macro Recording**] to [**Tools->Stop Macro**]. It is also possible to view this information while recording. For details on the macro-recording support facility, see Chapter 15, Using the Macro-Recording Support Facility.

Right-clicking displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up Menu Options	Toolbar button	Function
Clear Window		Clears the contents of the window.
Save		Saves the contents of the window into a text file.
Copy		Copies the selected contents onto the Windows® clipboard.
Toolbar display	-	Shows or hides the toolbar.
Customize toolbar	-	Customizes toolbar buttons.

Test tab

Shows the results and progress of the current test execution (test support facility). The test execution progress shows the current test being executed and the number of tests remaining. If errors occur then these are displayed in this window. For details on the test-support facility, see Chapter 16, Using the Test Support Facility.

Right-clicking displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up Menu Options	Toolbar button	Function
Clear Window		Clears the contents of the window.
Save		Saves the contents of the window into a text file.
Copy		Copies the selected contents onto the Windows® clipboard.
Toolbar display	-	Shows or hides the toolbar.
Customize toolbar	-	Customizes toolbar buttons.

The Version Control tab

Shows the results of version control actions. The tab is only displayed if a version control system is in use. For further details on version control, see Chapter 7, Version Control.

Right-clicking displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up Menu Options	Toolbar button	Function
Clear Window		Clears the contents of the window.
Save		Saves the contents of the window into a text file.
Copy		Copies the selected contents onto the Windows® clipboard.
Toolbar display	-	Shows or hides the toolbar.
Customize toolbar	-	Customizes toolbar buttons.

Press the "SHIFT+ESC" key, and the **Output** window closes.

The color of text or background and font shown in the Output window can be customized in the same manner as in other windows. You can also customize the **Build** tab of the Output window so that the texts in the lines of error messages will be highlighted in a color different from that of the texts in other lines.

To customize the current colors

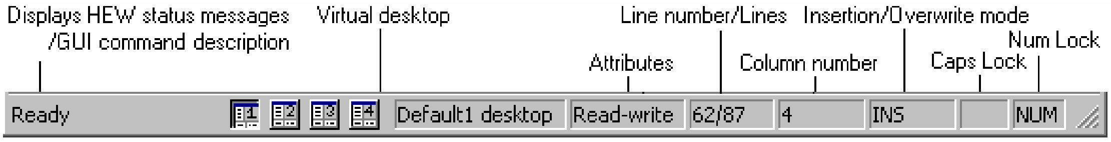
1. Select [**Setup->Format Views**] to open the **Format Views** dialog box.
2. Select the **Output** item from the tree in the left-hand section of the dialog box and expand it.
3. Select the category for which you wish to change the color from the table below.

Category	Foreground Color of the Color Tab (Default)	Background Color of the Color Tab (Default)	Tab(s) to be Applied	Type of Output to be Applied
Text	SYSTEM	SYSTEM	All tabs	All
Build Error Text	Black	White	Build	Error
Build Warning Text	Black	White	Build	Warning
Information Text	Black	White	Build	Information

4. Change the selection in the **Foreground** and **Background** lists of the **Color** tab.
5. Click the OK button.

1.2.7 Status bar

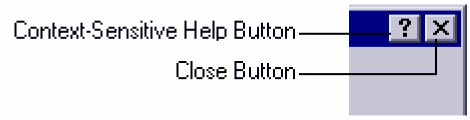
The status bar displays information about the current state of the HEW. The figure below shows the status bar.



1.3 Help system

The **Help** menu is the rightmost menu on the High-performance Embedded Workshop menu bar. It contains the **Help Topics** menu option, which, when selected, takes you to the main High-performance Embedded Workshop help window.

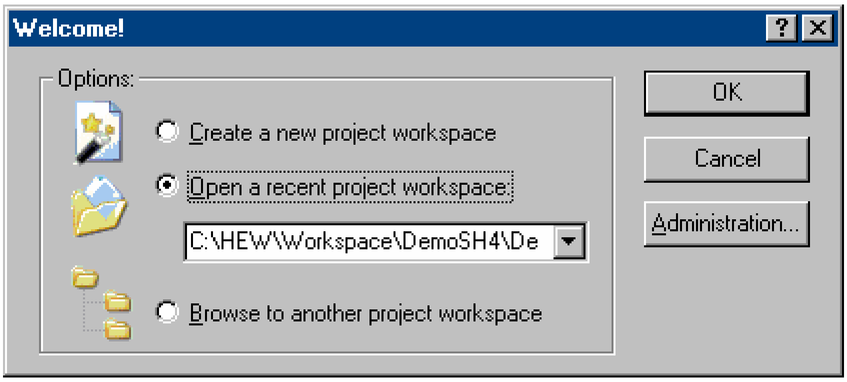
To obtain help on specific dialogs click the context sensitive help button, which is located in the top right-hand corner of each dialog (as shown in the figure below).



When this is clicked, the mouse pointer will change to a pointer with a question mark above it. Whilst the mouse pointer is in this state, click on the part of the dialog that you require assistance on. Alternatively, select the control for which you require help, and press the F1 key.

1.4 Launching the HEW

To initiate the HEW, open the **Start** menu of Windows®, select **Programs**, select **RENESAS**, select **High-performance Embedded Workshop**, and then select the shortcut of the HEW. The **Welcome!** dialog box shown will be displayed after the start-up screen by default.



Welcome!

Options:

○ Create a new project workspace

● Open a recent project workspace:

C:\HEW\Workspace\DemoSH4\De

○ Browse to another project workspace

If you have recently used a workspace, **Open a recent project workspace** is selected by default. Otherwise **Create a new project workspace** is selected.

To create a new workspace, select the **Create a new project workspace** button, and click the OK button.

To open one of the recent project workspaces, select the **Open a recent project workspace** button, select a workspace from the drop-down list, and click the OK button. The **Recent Workspace** list displays the same content as that seen in the workspace most recently used **File** list. This list also appears on the file menu.

To open a workspace by specifying a workspace file (".HWS" file), select the **Browse to another project workspace** button, and click the OK button.

To register or un-register a tool from the HEW, click the **Administration** button. Click the Cancel button to use the HEW without opening a workspace.

If you do not wish to open the **Welcome!** dialog box next time you launch the HEW, select [**Setup->Options**]. The **Options** dialog box is displayed. Remove a tick mark from the **Display Welcome Dialog** checkbox in the **Confirmation** tab.

If you do not wish to view the start-up screen when launching the HEW, remove a tick mark from the **Display Splash Screen** checkbox in the **Confirmation** tab of the **Options** dialog box.

1.5 Creating a new workspace

To create a new workspace

1. Select the **Create a new project workspace** option from the **Welcome!** dialog box and click the OK button or select [**File->New Workspace**]. The **New Project Workspace** dialog box will be displayed.
2. Enter the name of the new workspace into the **Workspace Name** field. This can be up to 32 characters in length and contain letters, numbers, and the underscore character. Especially, do not use a minus sign, or a space. As you enter the workspace name, the HEW will add a sub-directory and **Project Name** for you automatically. This can be changed if desired. This allows the workspace and project name to be different. To select the directory in which you would like to create the workspace, use the **Browse** button or type the directory into the **Directory** field manually.
3. Select the **CPU family** and **Tool chain** upon which you would like to base the workspace. Note that these cannot be changed once the workspace has been created.
4. When a new workspace is created, the HEW will also automatically create a project with the name specified in the **Project Name** field and place it inside the new workspace. The project types list displays all of the available project types (e.g. Application, Library etc.). Select the type of project that you want to create from this list. The project types displayed will be all valid types for the current pair of **CPU family** and **Tool chain**. The project types are classified in three classes: toolchain-only, debug-only (Debugger only - xxxxxx), and full project generator that configures both the debugger and toolchain aspect of the HEW.

5. Click the OK button to create the new workspace and project. This then launches the wizard you have selected to guide you through the creation process.

Note:

It is not possible to create a workspace if one already exists in the same directory.

1.6 Opening a workspace

To open a workspace

1. Select **Browse to another project workspace** option from the **Welcome!** dialog box and click the OK button or select [**File->Open Workspace**]. The **Open Workspace** dialog box will be displayed.
2. Select the workspace file (".HWS" file) that you want to open.
3. Click the **Select** button to open the workspace. If the HEW is set up to display information when a workspace is opened, the **Workspace Properties** dialog box will be displayed. Otherwise, the workspace will be opened.

   Note that whether the **Workspace Properties** dialog box is shown depends on the setting of either the **Show workspace information on workspace open** check box in the Workspace Properties dialog box or the **Display workspace information dialog on opening workspace** check box on the **Workspace** tab of the **Options** dialog box. The **Options** dialog box can be invoked via the [**Setup->Options**] menu option. Click the OK button in the **Workspace Properties** dialog box to open the workspace. Click the Cancel button to stop opening the workspace.

☑ Show workspace information on workspace open

The HEW keeps track of the last four workspaces that you have opened and adds them to the **File** menu under the **Recent Workspaces** submenu. This gives you a shortcut to opening workspaces, which you have used recently.

To open a recently used workspace

Select **Open a recent project workspace** in the **Welcome!** dialog box, select the name of the workspace from the drop-down list, and then click the OK button.

Another way is to select [**File->Recent Workspaces**], and then from this submenu select the name of the workspace.

Note:

The HEW only permits one workspace to be open at a time. Consequently, if you attempt to open a second workspace, the first will be closed before the new one is opened.

1.7 Using old workspaces

The HEW can open any workspace that was created on a previous version of the HEW. This should be automatically upgraded when the workspace is opened. A back-up version of the initial workspace or project file must be saved in the current directory of the file that has been upgraded.

1.8 Saving a workspace

To save a HEW workspace, select the [**File->Save Workspace**] menu option.

1.9 Closing a workspace

To close a HEW workspace, select the [**File->Close Workspace**] menu option. If there are any outstanding changes to the workspace or any of its projects you will be requested whether or not you wish to save them.

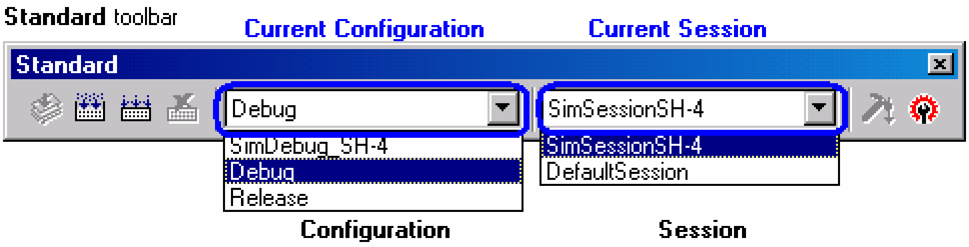
1.10 Exiting the HEW

The HEW can be exited by selecting the [**File–>Exit**] menu option, pressing ALT+F4, or by selecting the close option from the system menu (which is opened by clicking the icon at the upper-left corner of the HEW title bar).

1.11 Component system overview

The HEW allows the user to extend the HEW functionality by adding additional components to the system. This is achieved by registering the component in the **Tools Administration** dialog box. These components can add windows, menus and toolbars to the HEW system. Examples of the components are the debugger and builder components of HEW. The debugger component adds all of the menus and toolbars associated with the debugger and the builder component does the same for the build functionality. The components you have registered in the system will modify the look and feel of HEW. In some cases you may not have some of the menus which you can see in this manual. For instance if the builder component is not installed you will not have the toolchain menu option in the **Build** menu.

1.12 Overview of Configurations and Sessions



The drop-down list box on the Standard toolbar shows the current configuration and session.

This example shows the configuration and session in which the Renesas SuperH Standard toolchain was selected at creation of a project and "SH-4 Simulator" was selected as the target debugging platform.

You can switch between different configurations and sessions by selecting one in the drop-down list box.

It is also possible to add or delete configurations or sessions as necessary.

- **Configuration**

What is configuration?

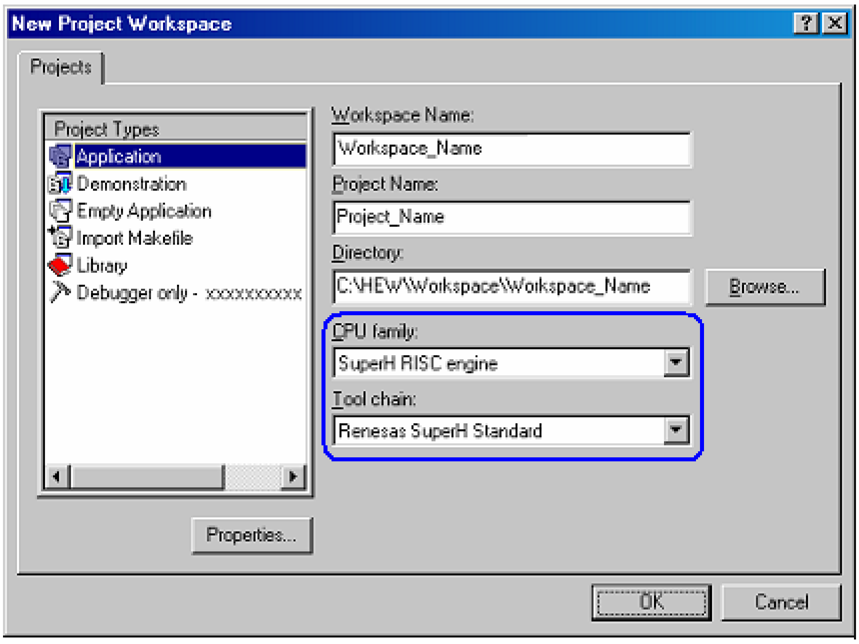
Configurations are the build option settings (e.g., output of debug information or optimization) having their own names. The term "configuration" can also be referred to as "build configuration".

In the figure of the Standard toolbar shown as an example, configurations "SimDebug_SH-4", "Debug", and "Release" are available.

Configurations created by the HEW

When a project is created after selection of the toolchain, the HEW automatically creates configurations "Debug" and "Release".

Selecting "SuperH RISC engine" from CPU family and "Renesas SuperH RISC engine" from Tool chain respectively



New Project Workspace

Projects

Project Types
- Application
- Demonstration
- Empty Application
- Import Makefile
- Library
- Debugger only - xxxxxxxxxx

Workspace Name: Workspace_Name

Project Name: Project_Name

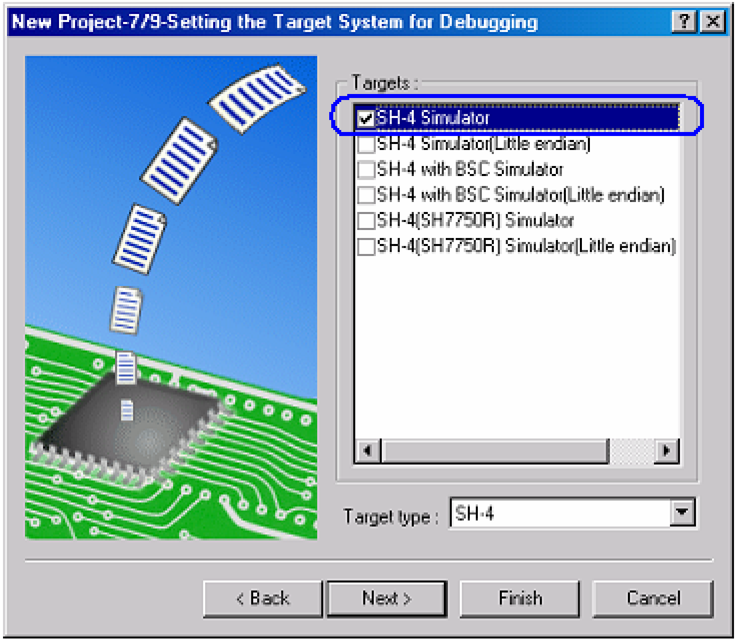
Directory: C:\HEW\Workspace\Workspace_Name

CPU family: SuperH RISC engine

Tool chain: Renesas SuperH Standard

If you have selected a target debugging platform at creation of the project, the HEW creates a configuration suitable for the debugging platform. If you have selected "SH-4 Simulator" in Targets at creation of the project, configuration "SimDebug_SH-4" is created.

Selecting "SH-4 Simulator" in Targets as the debugging platform connected with the HEW



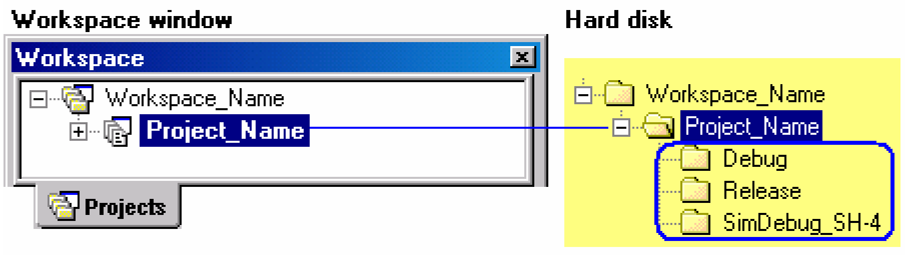
New Project-7/9-Setting the Target System for Debugging

Targets :
- ☑ SH-4 Simulator
- □ SH-4 Simulator(Little endian)
- □ SH-4 with BSC Simulator
- □ SH-4 with BSC Simulator(Little endian)
- □ SH-4(SH7750R) Simulator
- □ SH-4(SH7750R) Simulator(Little endian)

Target type : SH-4

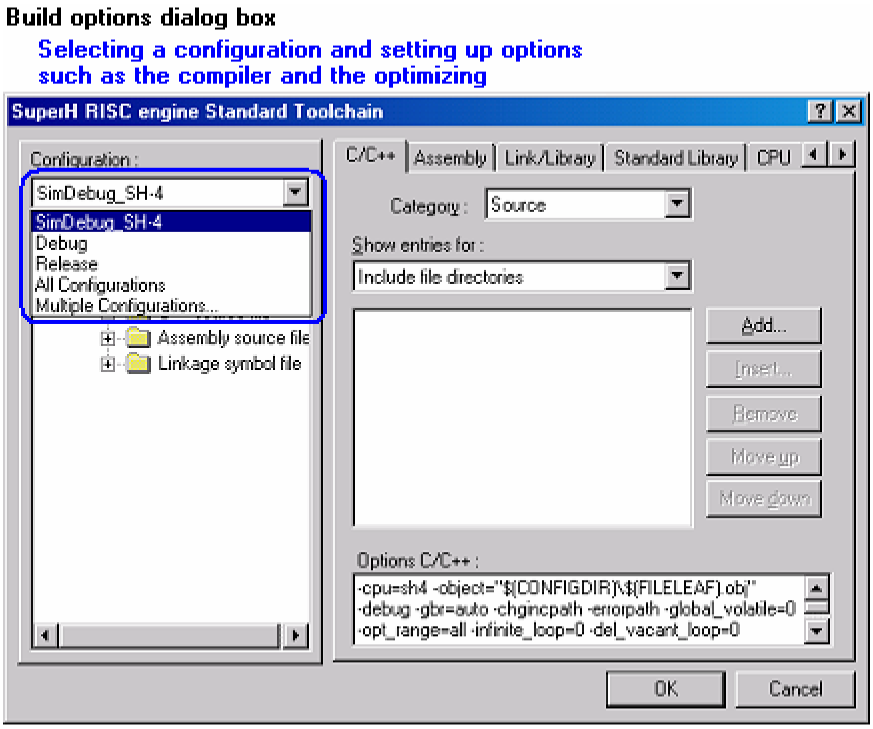
Configuration names can be changed when creating a project. Directories corresponding to each of the configurations are created under the project directory *. These directories have the names of the configurations.

NOTE: *The project directory having the project name is created under the workspace directory used for creation of a new workspace.



How to set up build options

To set up build options kept with a configuration, select the configuration from the drop-down list box in the dialog box for set up build option opened from the Build menu. Options such as the compiler, assembler, and optimizing linkage editor can be set up. The figure below shows the dialog box used for setup of build options for the Renesas SuperH Standard toolchain.



Configuration : SimDebug_SH-4

Category : Source

Show entries for : Include file directories

Options C/C++ :

```
-cpu=sh4 -object="$(CONFIGDIR)\$(FILELEAF).obj"
-debug -gbr=auto -chgincpath -errorpath -global_volatile=0
-opt_range=all -infinite_loop=0 -del_vacant_loop=0
```

See section 2.7, Build configurations, for more information about a configuration.

- **Session**

What is session?

Sessions, which have their own names, are the units to manage various settings such as HEW debugger options * used for debugging with connection to the debugging platform and options specific to the debugging platform, and information on the Memory and Register windows and their positions. The term "session" can also be referred to as "debugger session".

In the figure of the Standard toolbar shown as an example, sessions "SimSessionSH-4" and "DefaultSession" are available. Information on each session is saved in an individual file in the HEW project.

NOTE: *There are various option settings such as the debugging platform to be connected with the HEW, the object format for debugging, download modules, and the timing of connecting the HEW with the debugging platform or executing a command batch by the HEW.

Sessions created by the HEW

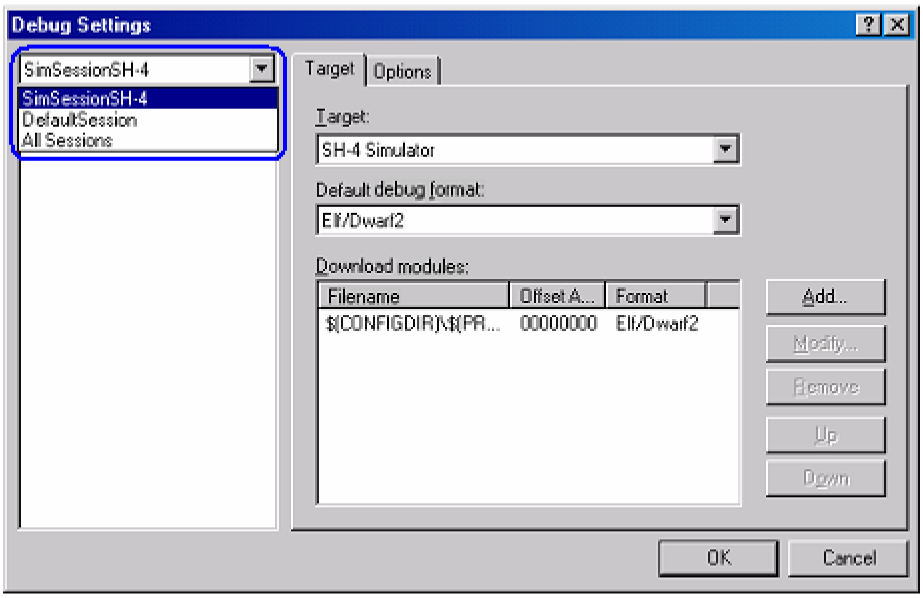
When a project is created after selection of the toolchain, the HEW automatically creates session "DefaultSession". If you have selected a target debugging platform at creation of the project, a session is automatically created for connecting the HEW with the selected debugging platform. For example, if you have selected "SH-4 Simulator" in "Target" at creation of a project, session "SimSessionSH-4" is automatically created.

How to set up HEW debugger options

To set up HEW debugger options kept with a session, select the session from the drop-down list box in the Debug Settings dialog box opened from the Build menu. Options can be set up in the Target and Options tabs.

HEW debugger options dialog box

Selecting a session and setting up options in the Target and Options tabs



Target: SH-4 Simulator

Default debug format: Elf/Dwarf2

Download modules:

Filename	Offset A...	Format
$(CONFIGDIR)\$(PR...	00000000	Elf/Dwarf2

See section 17.1.6, Debugger sessions, for more information about a Debugger sessions.

1.13 Overview of Macro-Recording Support facility and Test Support facility

If the program is modified after completion of the test, this program must be tested again. Re-testing takes a significant number of steps in development of programs.

You can use the test support facility to save the data in specific windows as test-image data into files. This test-image data is for comparison after a test. A sequence of operations before comparison, such as from the start to end of a test, is managed as a test case. You can repeat evaluation of a test by re-testing and comparison of data.

Before creating a test case, a test-script file to be executed in the test must be prepared. This test-script file is in the HEW command-line command format.

You can also use the macro-recording support facility to save some of the user operations (such as those done while debugging) into files in the HEW command-line command format. Use these files as test-script files to be executed in tests.

- **Macro-recording support facility**

The macro-recording support facility allows you to record operations, which are associated with the HEW system applications *1, build *2, and debugging *3, as HEW command-line commands or to execute these recorded commands.

Files to record the operations (HEW macro files) are command line batch files that have “hdc” as the extension and can be modified. These files are stored in the “Macro” directory under the HEW installation directory. By default, the file Default.hdc is created.

Note, however, that HEW is not capable of recording all operations in the HEW system corresponding to the HEW command-line commands. For the operations that can be recorded, a macro record icon (●) is shown in the "Macro Recording" column of the menu list. This indicates that this operation can be recorded into a HEW macro file. Also see section 15.5, Functions that can be recorded into macro files, where the HEW command-line commands to be recorded are listed.

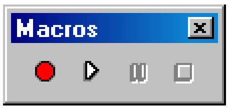
*1. Changing a project, session, or configuration

*2. Compilation and build. Support for this function depends on the debugging platform.

*3. Downloading a module, changing a memory value or register value, setting/deleting a software breakpoint, and running a program

The macro-recording support facility is available in the **Tools** menu and on the **Macros** toolbar.

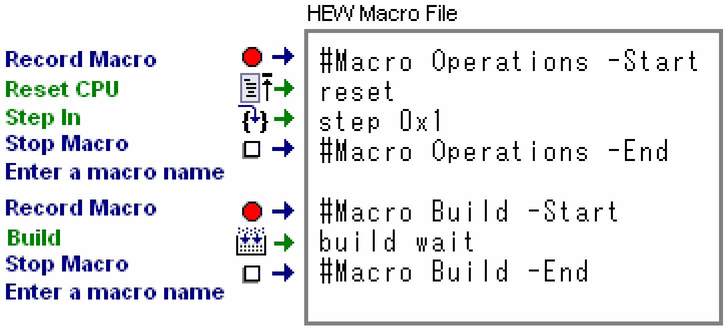
Macros toolbar



If the **Macros** toolbar is hidden, select [**Setup->Customize**] to open the **Customize** dialog box. Select the **Macros** checkbox on the **Toolbars** tab of the **Customize** dialog box.

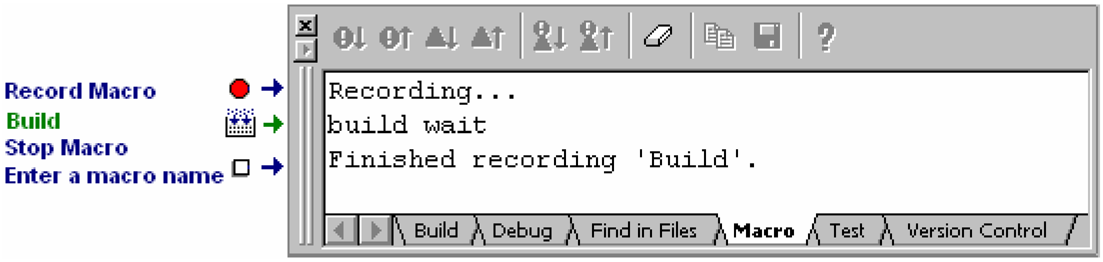
Tools Menu	Macros Toolbar	Function
Macros	-	Opens the **Macro** dialog box. This dialog box has the following features and lists the macros recorded in each of the selected HEW macro files. - Creating a new blank macro file - Importing an existing macro file - Starting a macro record operation - Starting a macro play function - Editing a current macro file - Assigning the selected macro - Removing the selected macro
Macro Recording	●	Starting a macro record operation
Play Macro	▷	The **Select Macro Function** dialog box opens. Choose which macro you wish to execute
-	⏸	Pausing the current macro operation
Stop Macro	□	Stopping the current macro operation

The procedure from selection of **Record Macro** (●) to that of **Stop Macro** consists one macro and one HEW macro file can record multiple macros (a set of HEW command-line commands). A macro includes multiple HEW command-line commands.



```
#Macro Operations -Start
reset
step 0x1
#Macro Operations -End

#Macro Build -Start
build wait
#Macro Build -End
```

The current records of macros are shown in the **Macro** tab of the **Output** window. The "Build" macro in the figure above is outputted as follows.



```
Recording...
build wait
Finished recording 'Build'.
```

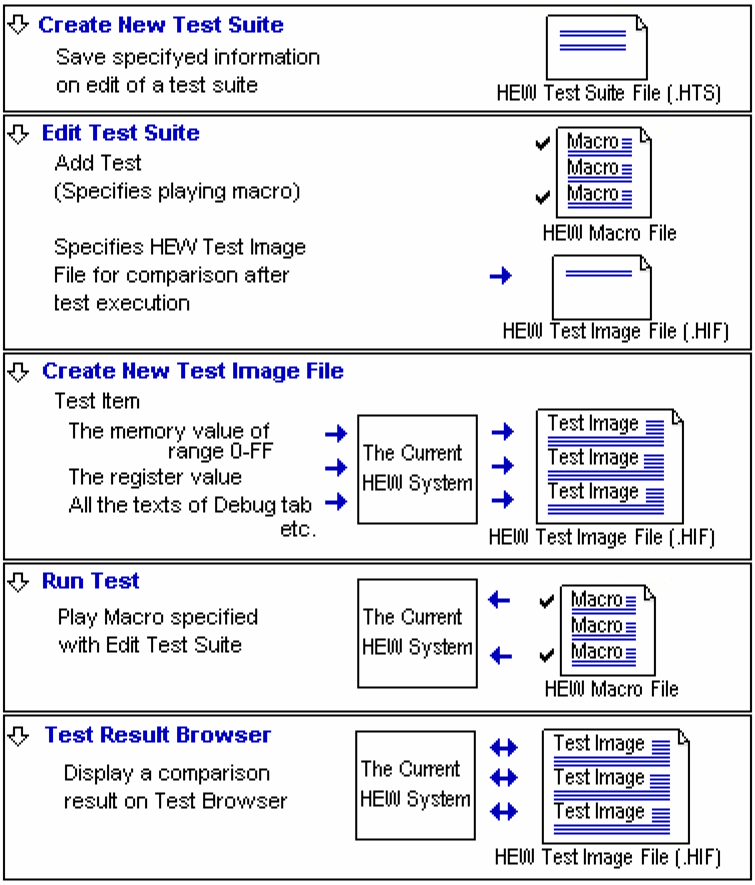
See section 1.13.1, Example of test procedures, for an operation procedure.

- **Test support facility**

The test support facility allows you to create a test image file of the selected test items (test image data), execute a macro created by the macro-recording support facility or an existing HEW command batch file, and compare a test image file with the current HEW system. Test image data can be saved into a test image file (.HIF).

Note, however, that HEW is not capable of acquiring test-image information of all functions in the HEW system. For the items from which test-image information can be acquired, see section 16.6, Functions that can be saved as test-image data into test-image files.

The typical test procedure is outlined in the figure below. See section 1.13.1, Example of test procedures, for a test procedure.



The test support facility is available in the **Test** menu and in the pop-up menu of the **Test** tab of the workspace window.

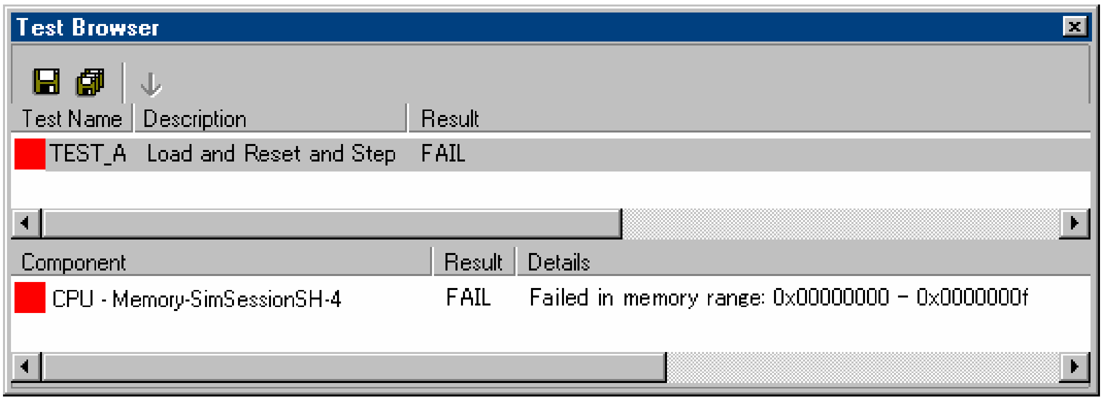
The first step towards using the test support facility is to create a test suite (a set of tests). This is available on the **Test** menu. Once a test suite is created, following operations can also be done in the **Test** tab of the workspace window.

Modify the test suite and specify macros to be executed during the test and the test image file for comparison.

Then select some test items to be saved a test image file. Acquire and save test image of the selected test items, which you wish to use for comparison.

After completion of the test (macros) the selected test items are compared. The result is displayed in the test browser window.

If there is any difference between the test result and the saved the test image file, the result of the test item is shown as “FAIL”. You can view the details of the error by double-clicking the line of this test item.



Test Name	Description	Result
🟥 TEST_A	Load and Reset and Step	FAIL

Component	Result	Details
🟥 CPU - Memory-SimSessionSH-4	FAIL	Failed in memory range: 0x00000000 – 0x0000000f

1.13.1 Example of test procedures

This page introduces an example of test procedures, which is a sequence through build and debugging using the macro-recording support facility and test support facility.

This example uses a program that sorts ten random data items written in C language, which are typically included in the Tutorial project in an emulator debugger or in a project for which "Demonstration" has been selected as the project type in the SuperH RISC engine simulator, in ascending/descending order. This example lets you see a difference in data stored in an array.

Before starting the test in this example, array “a” in the program code must be changed from a local to global variable.

- **Before test**

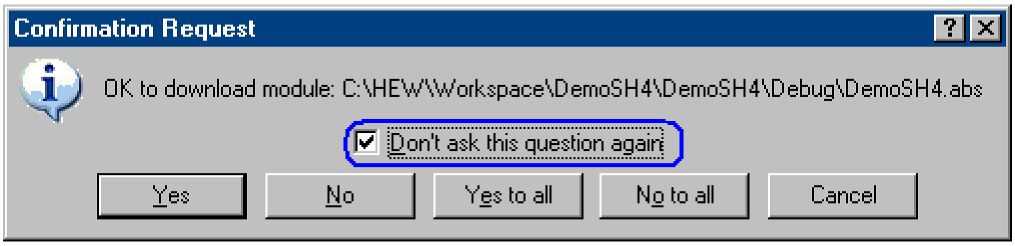
First, open a workspace and connect the HEW to the debugging platform.

1. Create a new project workspace or open an existing project workspace. For creation of a project workspace, see section 17.1.2, Selecting a debugging platform.
2. Select a session, which you wish to use for connection with the debugging platform, from the session list on the **Standard** toolbar. For selection of a session, see section 17.1.6 (1), Selecting a session.

After some operations, a **Confirmation Request** dialog box may appear. If you have executed build all on the entire program, a **Confirmation Request** dialog box appears by default asking if you wish to download the program. To download the program, click the **Yes** button.

Since the action of clicking on the buttons in this confirmation dialog box cannot be recorded into macros, this dialog box will open next time you execute a macro. To continue the macro execution, you should click one of the buttons. The execution must be continued if you are using the test-support facility, which is used to test a sequence of operations. For this reason, make the following setting so that the confirmation dialog box will not appear.

In this example, select the **Don't ask this question again** check box in **Confirmation Request** dialog box.



OK to download module: C:\HEW\Workspace\DemoSH4\DemoSH4\Debug\DemoSH4.abs

☑ Don't ask this question again

Save the HEW environment setting that you wish to use for tests so that the environment for the tests will always be the same.

1. Select [**File->Save Workspace**].
2. Select [**File->Save Session**].

You can always use the same environment to start a test by loading the HEW environment setting that has been saved.

- **Preparing a test script to be used in the example of test procedures**

In this example, use the macro-recording support facility to record the following combination of operations as HEW command-line commands into a macro. This macro should be used as a test script at execution of a test.

Classification	Operation	Recordable HEW Command-Line Commands
Preparation before the running the program - Initialize values - Prepare a program	Reload the session. (See "Step 1 (2)".)	refresh_session command
	Initialize register values ***1.** (See "Step 1 (3)".)	register_set command ***2**
	Build the program for demonstration ***1.** (See "Step 1 (4)".)	build command
Download the program	Download the program for demonstration. (See "Step 1 (5)".)	file_load command
Add variables, run the program, and check the result	Use the memory fill function to set the memory value of array "a" to 0. (See "Step 1 (6)".	memory_fill command
	Reset the CPU. (See "Step 1 (7)".)	reset command
	Run the program until a selected line within a main function is reached. (See "Step 1 (9)".)	go_till command
	Add the array, which stores random data, to the **Watch** window ***1.** (See "Step 1 (10)".)	watch_add command

*1. Support for this function depends on the debugging platform.

*2. Refer to the help files for the emulator or the simulator.

- **Example of test procedures**

Invoke the test support facility and select the data in an array (memory content at the address of an array) as the target of the test. Save test-image information to the test image file that you wish to use for comparison and modify the program so that the number of times to store data into the array will be reduced. Execute the test and see that the data stored in the array has changed. Then modify the program again to restore the number of times to store data and execute the test. Make sure that data in the array is the same that in the saved test image file.

Step 1: Recording a macro

Step 2: Editing a macro (viewing records)

Step 3: Playing a macro

Step 4: Creating a test suite

Step 5: Editing a test suite

Step 6: Creating a test image file

Step 7: Modifying the program before the test

Step 8: Viewing the test result (unmatched)

Step 9: Modifying the program back and executing the test again

Step 10: Viewing the test result (matched)

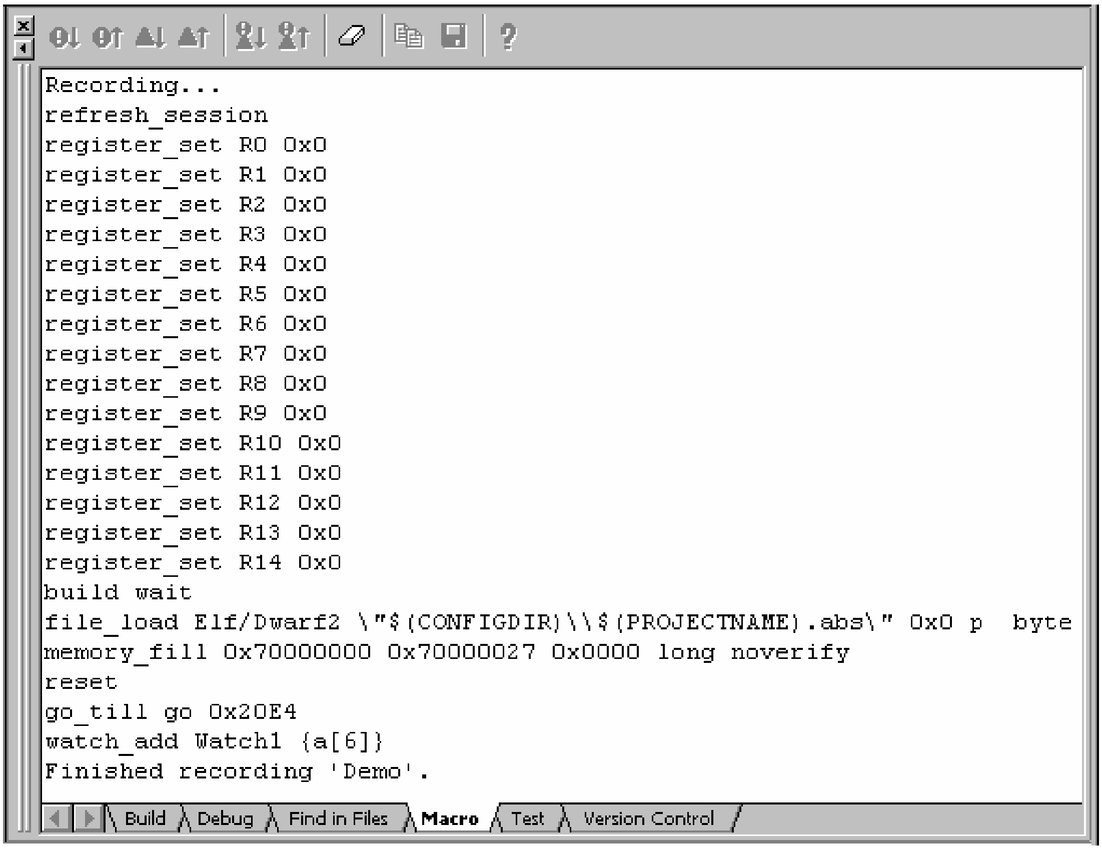
1.13.2 Step 1: Recording a macro

In this example, use the macro-recording support facility to record the following combination of operations as HEW command-line commands.

Recorded HEW command-line commands are output to the **Macro** tab of the **Output** window. However, some operations open another tab.

1. Click the **Record Macro** button (●) on the **Macros** toolbar. The mouse pointer is modified to include the record icon (↖●). The message "Recording..." is shown in the **Macro** tab of the **Output** window.
2. Select [**File->Refresh Session**].
   This records the refresh_session command.
3. Click the **Registers** button (R1) on the **CPU** toolbar to open the **Register** window. Double-click the register, of which you wish to change the value, to invoke the **Set Value** dialog box. Enter 0 in **Value**.
   This records the register_set command.
4. Click the **Build** button () on the **Standard** toolbar to build the program for demonstration.
   This records the build command.
5. Double-click to download the program for demonstration within the **Download module** folder in the **Projects** tab of the workspace window.
   This records the file_load command.
6. Click the **Memory** button () on the **CPU** toolbar to invoke the **Display Address** dialog box. Enter the address of the data field in **Display Address**. Then select **Fill** from the pop-up menu. Enter 0 as the data in the **Fill** dialog box.
   This records the memory_fill command.
7. Click the **Reset CPU** button () on the **Debug Run** toolbar to reset the CPU.
   This records the reset command.
8. Open a source file including a main function in the editor window.
9. Right-click on the last line of the main function and select **Go To Cursor** from the pop-up menu.
   This records the go_till command.
10. Add the array element "a[6]" storing random data to the **Watch** window.
    This records the watch_add command.
11. Click the **Stop Macro** button (□) on the **Macros** toolbar. The mouse pointer returns to be an arrow.
12. The **Add New Macro Function** dialog box is displayed. Enter a macro name (e.g., Demo). The message "Finished recording 'Demo'." is shown in the **Macro** tab of the **Output** window.

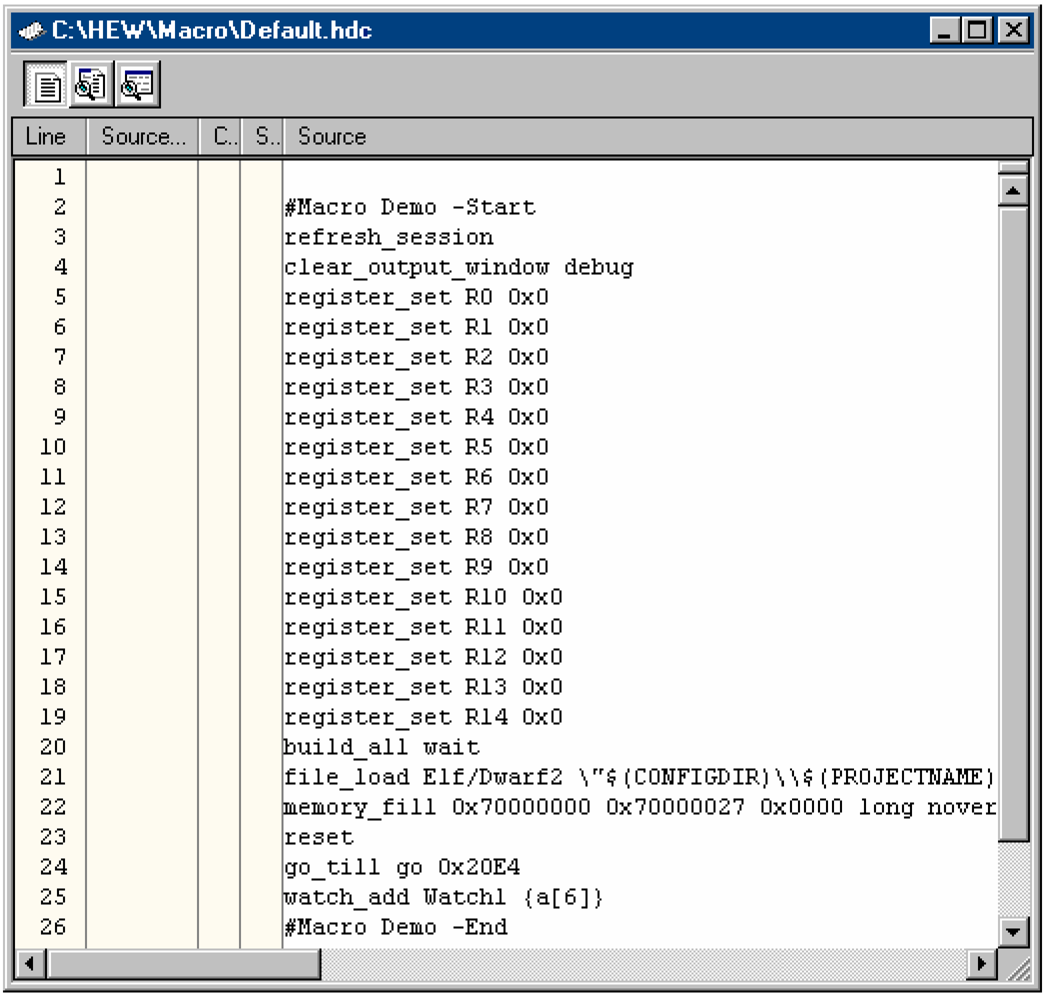
To view the recorded HEW command-line commands, be sure to select the **Macro** tab of the **Output** window.



```
Recording...
refresh_session
register_set R0 0x0
register_set R1 0x0
register_set R2 0x0
register_set R3 0x0
register_set R4 0x0
register_set R5 0x0
register_set R6 0x0
register_set R7 0x0
register_set R8 0x0
register_set R9 0x0
register_set R10 0x0
register_set R11 0x0
register_set R12 0x0
register_set R13 0x0
register_set R14 0x0
build wait
file_load Elf/Dwarf2 \"$(CONFIGDIR)\\$(PROJECTNAME).abs\" 0x0 p  byte
memory_fill 0x70000000 0x70000027 0x0000 long noverify
reset
go_till go 0x20E4
watch_add Watch1 {a[6]}
Finished recording 'Demo'.
```

1.13.3 Step 2: Editing a macro (viewing records)

1. Select [**Tools->Macros**] to invoke the **Macro** dialog box.
2. The HEW macro file “Default”, in which a macro has been recorded, is listed in **Current macro file**.
3. Select the macro “Demo” in **Macro functions** and click the **Edit** button.
4. The editor window shows the HEW macro file “Default.hdc”, in which a macro has been recorded.



```

#Macro Demo -Start
refresh_session
clear_output_window debug
register_set R0 0x0
register_set R1 0x0
register_set R2 0x0
register_set R3 0x0
register_set R4 0x0
register_set R5 0x0
register_set R6 0x0
register_set R7 0x0
register_set R8 0x0
register_set R9 0x0
register_set R10 0x0
register_set R11 0x0
register_set R12 0x0
register_set R13 0x0
register_set R14 0x0
build_all wait
file_load Elf/Dwarf2 \"$(CONFIGDIR)\\$(PROJECTNAME)
memory_fill 0x70000000 0x70000027 0x0000 long nover
reset
go_till go 0x20E4
watch_add Watch1 {a[6]}
#Macro Demo -End
```

Within a HEW macro file, the HEW command-line commands of the operations listed below are included between the lines showing information to identify macros.

Operation	**Recordable HEW Command-Line Commands**
Reload the session.	refresh_session command
Initialize register values ***1.**	register_set command ***2**
Build the program for demonstration ***1.**	build command
Download the program for demonstration.	file_load command
Use the memory fill function to set the memory value of array “a” to 0.	memory_fill command
Reset the CPU.	reset command
Run the program until a selected line within a main function is reached.	go_till command
Add the array, which stores random data, to the **Watch** window ***1.**	watch_add command

***1**. Support for this function depends on the debugging platform.

***2**. Refer to the help files for the emulator or the simulator.

When a macro includes operations such as running a program, the **Debug** tab of the **Output** window shows information including the cause of a stop. This may cause difference in comparison of test results. To clear the contents of the tabs in the **Output** window, the HEW command-line command clear_output_window is available. It is possible to add the line at the next line of **#Macro Demo-Start** to clear the information in the **Debug** tab of the **Output** window.

```
clear_output_window debug
```

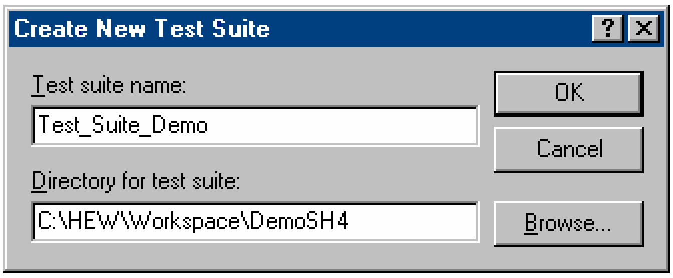
Modify the description as required and click the **Save File** button (💾) on the **Editor** toolbar. Close the HEW macro file “Default.hdc”.

1.13.4 Step 3: Playing a macro

1. Click the **Play Macro** button (▷) on the **Macros** toolbar. The **Select Macro Function** dialog box opens. Selecting the macro “Demo” starts playing the recorded commands.
2. The mouse pointer is modified to include the play icon (↖▷).
3. When the playback of a macro is completed, the mouse cursor returns to be an arrow.

1.13.5 Step 4: Creating a test suite

1. Select **Test->Create New Test Suite** to invoke the **Create New Test Suite** dialog box.

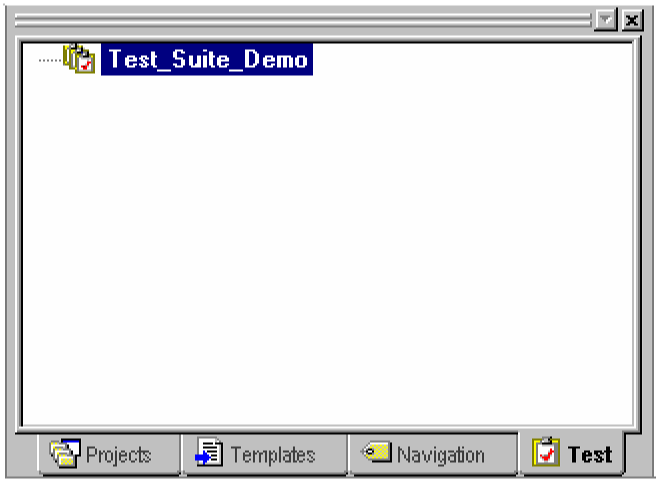


Create New Test Suite

Test suite name: Test_Suite_Demo

Directory for test suite: C:\HEW\Workspace\DemoSH4

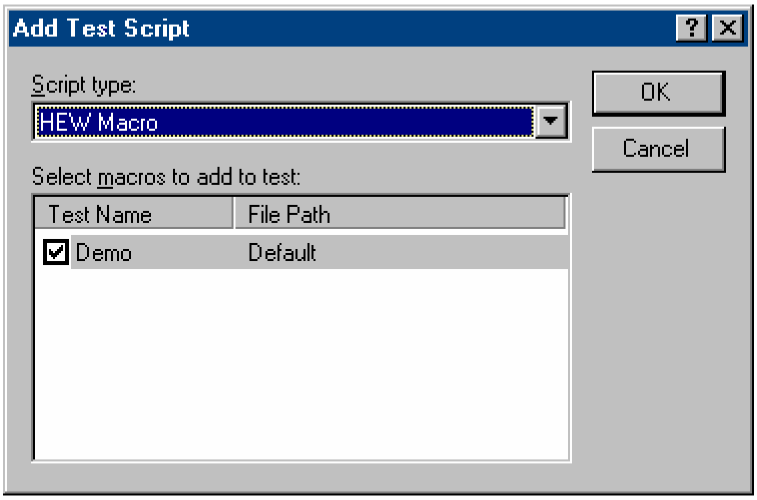
2. Enter “Test_Suite_Demo” in **Test suite name**.
3. Initially the workspace directory is shown in **Directory for test suite**. This can be modified as required.
4. Click the OK button.
5. Displays the **Test** tab of the workspace window. Shows the “Test_Suite_Demo” test-suite icon.



1.13.6 Step 5: Editing a test suite

Select the macro recorded at step 1 for running a test.

1. Right-click on the “Test_Suite_Demo” test-suite icon to display a pop-up menu.
2. Select **Edit Test Suite** to open the **Modify Test Suite** dialog box.
3. Click the **Add** button. The **Add New Test** dialog box opens.
4. Enter “Test_Demo” in **Test name**.
5. Enter “Demonstration” in **Test description**.
6. Click the **Add** button. The **Add Test Script** dialog box opens.
7. Select the “Demo” checkbox in **Select macros to add to test**. Click the **OK** button.



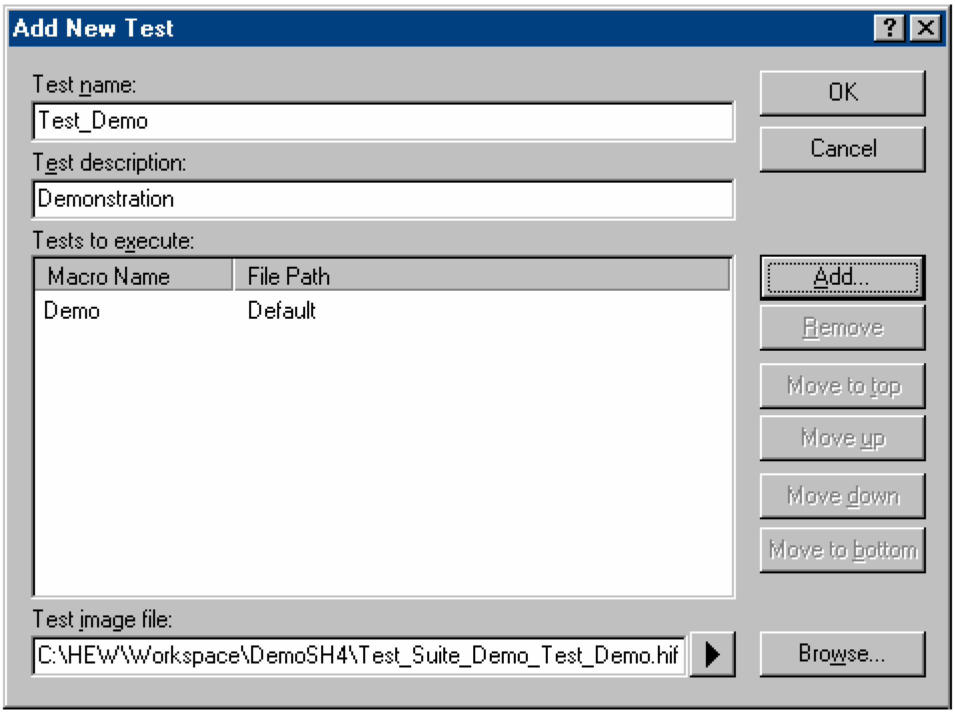
Add Test Script

Script type: HEW Macro

Select macros to add to test:

Test Name	File Path
☑ Demo	Default

8. **Test Image file** shows the test suite name entered at step 4, '_', and test name “Test_Demo” entered in **Name** (test suite name_test name).



Add New Test

Test name: Test_Demo

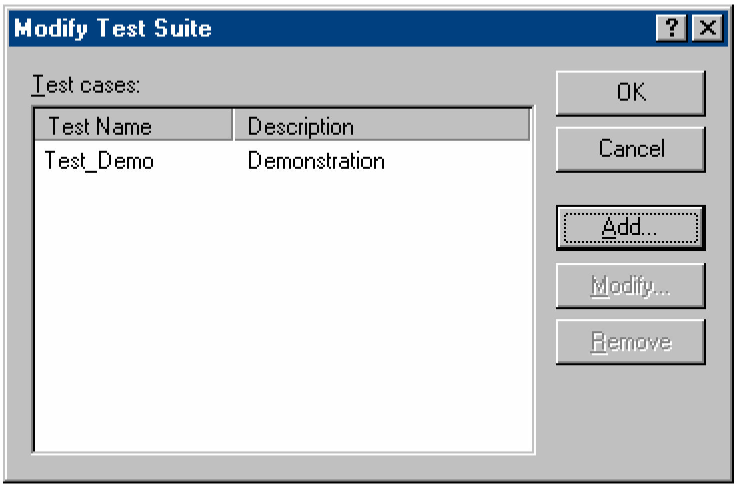
Test description: Demonstration

Tests to execute:

Macro Name	File Path
Demo	Default

Test image file: C:\HEW\Workspace\DemoSH4\Test_Suite_Demo_Test_Demo.hif

9. Click the **OK** button. The test “Test_Demo” is added to the **Test cases** list.

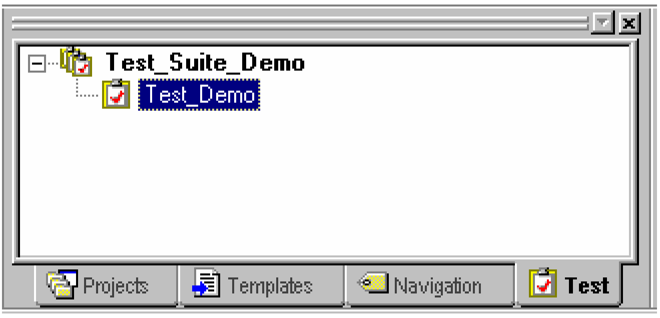


Modify Test Suite

Test cases:

Test Name	Description
Test_Demo	Demonstration

10. Click the **OK** button.
11. The “Test_Demo” test icon appears under the “Suite” test-suite icon.

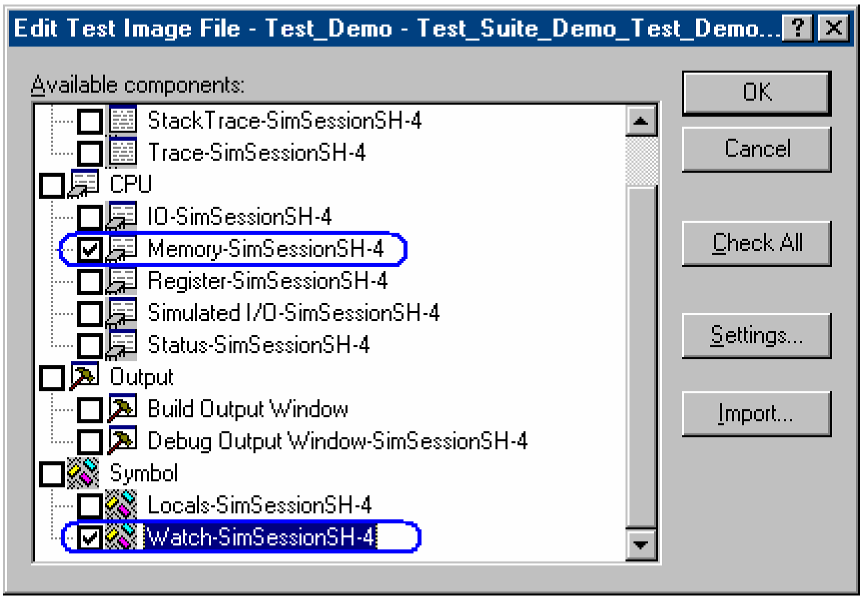


1.13.7 Step 6: Creating a test image file

To compare the **Memory** content at the address of an array or the contents of the **Watch** window, the range of data acquisition must be selected.

The test image of a test item is saved to the test image file "Test_Suite_Demo_Test_Demo.hif" specified at step 5.

1. Right-click on the “Test_Suite_Demo” test-suite icon to display a pop-up menu.
2. Select **Edit Test Image File** to invoke the **Edit Test Image File** dialog box. The title of the dialog box includes the test name “Demo” and test-image file name “Test.hif”. Select the checkboxes [Memory] (under [CPU] in [Item]) and [Watch] (under [Symbol]).
3. Select the checkboxes **Memory-xxxxxx** (under **CPU** in **Available components**) and **Watch-xxxxxx** (under **Symbol**).

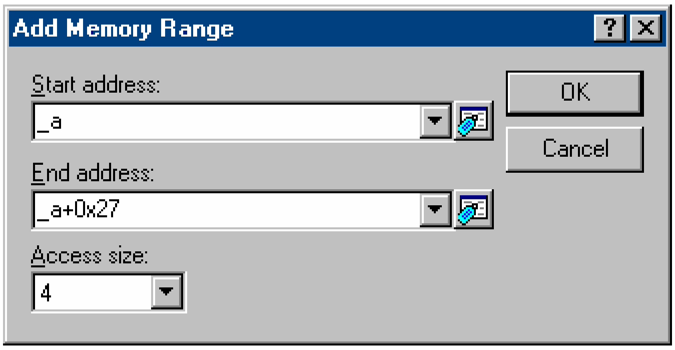


Edit Test Image File - Test_Demo - Test_Suite_Demo_Test_Demo...

Available components:

- □ StackTrace-SimSessionSH-4
- □ Trace-SimSessionSH-4
- □ CPU
  - □ IO-SimSessionSH-4
  - ☑ Memory-SimSessionSH-4
  - □ Register-SimSessionSH-4
  - □ Simulated I/O-SimSessionSH-4
  - □ Status-SimSessionSH-4
- □ Output
  - □ Build Output Window
  - □ Debug Output Window-SimSessionSH-4
- □ Symbol
  - □ Locals-SimSessionSH-4
  - ☑ Watch-SimSessionSH-4

4. Double-click **Memory-xxxxxx** under **CPU** in **Available components**. The **Edit Test Memory Ranges** dialog box opens.
5. Click the **Add** button to display the **Add Memory Range** dialog box.
6. Enter the start address of an array in **Start address** and end address in **End address** and select the size in **Access size**. Click the **OK** button.



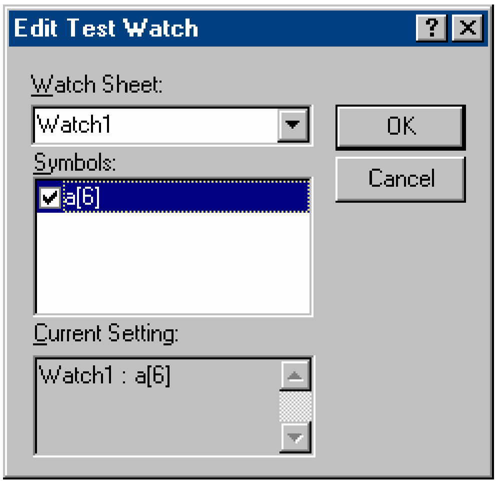
Add Memory Range

Start address: _a

End address: _a+0x27

Access size: 4

7. Click the **OK** button.
8. The content of array element "a[6]" in the [Watch1] sheet is saved into the test-image file. Double-click **Watch-xxxxxx** under **Symbol** in **Available components** of the **Edit Comparison Settings** dialog box. The **Edit Test Watch** dialog box opens. Select the checkbox of "a[6]" in **Symbols** and click the **OK** button.



Edit Test Watch

Watch Sheet: Watch1

Symbols:

☑ a[6]

Current Setting:

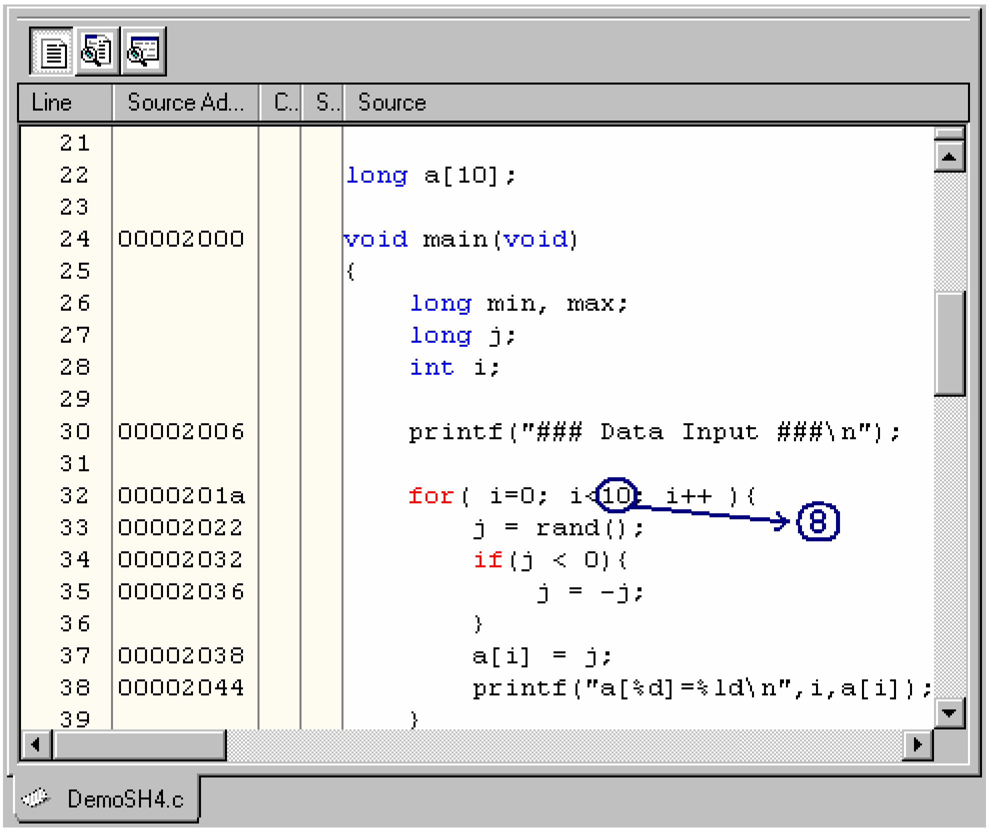
Watch1 : a[6]

9. Click the OK button in the **Edit Test Image File** dialog box.

In the state of current HEW system, the test image specified in test item is acquired, and the test image is saved to the test image file created at step5.

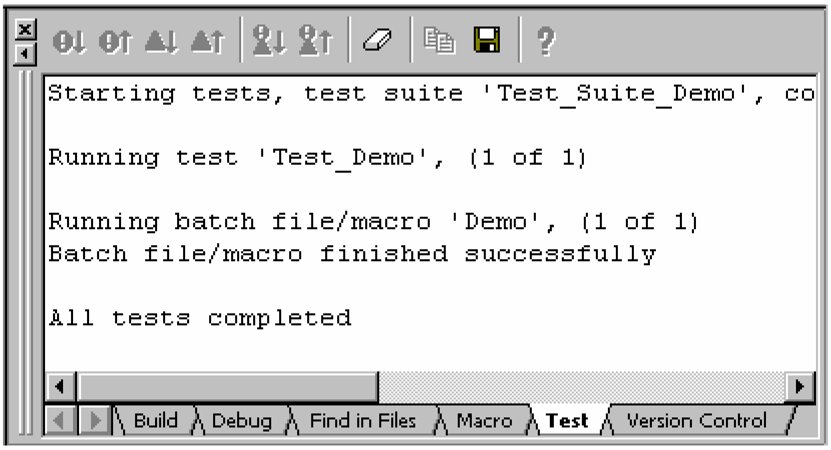
1.13.8 Step 7: Modifying the program before the test

In this example, the target of the test is a selected range of memory data in arrays. Then modify the program to reduce the number of times to store data into the array, so that the saved test-image information and the actual memory data will not match.



```
          long a[10];

00002000  void main(void)
          {
              long min, max;
              long j;
              int i;

00002006      printf("### Data Input ###\n");

0000201a      for( i=0; i<10; i++ ){
00002022          j = rand();
00002032          if(j < 0){
00002036              j = -j;
                  }
00002038          a[i] = j;
00002044          printf("a[%d]=%ld\n",i,a[i]);
              }
```

1. Open a source file including a main function in the editor window and modify the program to reduce the number of times to store data into the array.
2. Click the **Save File** button (💾) on the **Editor** toolbar.
3. Right-click on the “Test_Demo” test icon to display a pop-up menu.
4. Select **Run Test Case** to start the test.
5. The progress and result of the test are shown in the **Test** tab of the **Output** window.

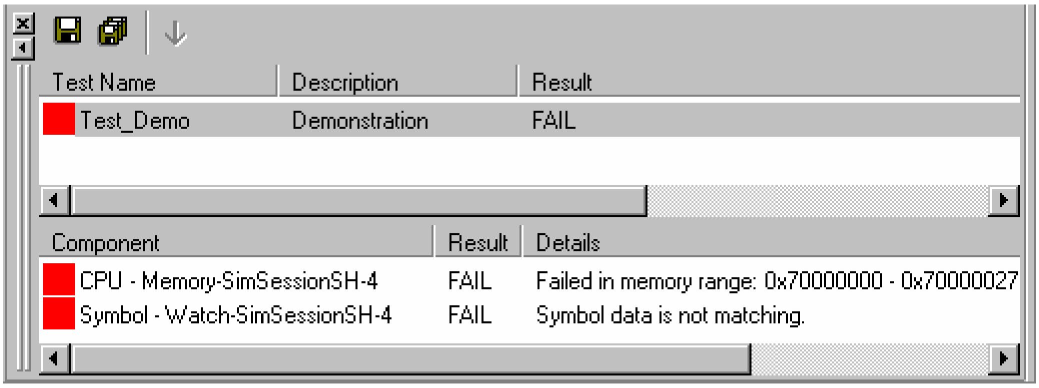


```
Starting tests, test suite 'Test_Suite_Demo', co

Running test 'Test_Demo', (1 of 1)

Running batch file/macro 'Demo', (1 of 1)
Batch file/macro finished successfully

All tests completed
```

6. When the test is completed, the **Test Browser** window opens.

1.13.9 Step 8: Viewing the test result (unmatched)

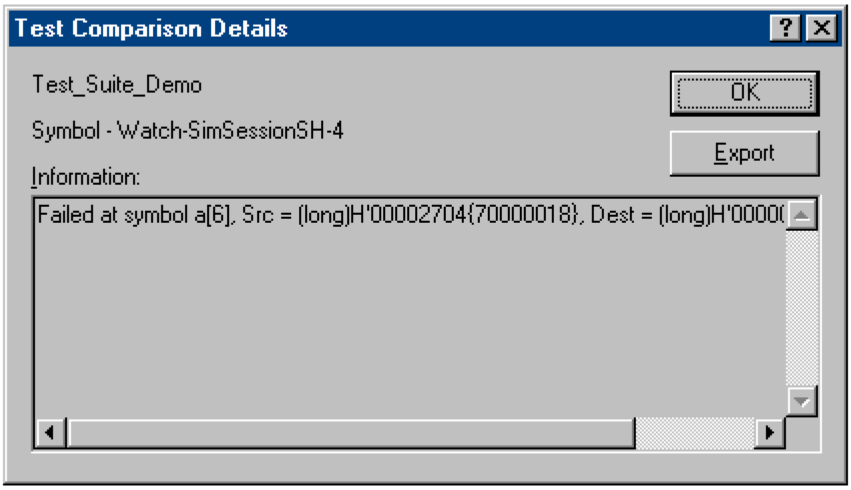
You can view the test result in the Test Browser window.



	Test Name	Description	Result
🟥	Test_Demo	Demonstration	FAIL

	Component	Result	Details
🟥	CPU - Memory-SimSessionSH-4	FAIL	Failed in memory range: 0x70000000 - 0x70000027
🟥	Symbol - Watch-SimSessionSH-4	FAIL	Symbol data is not matching.

1. To view the details of unmatched data, double-click a test item with a red icon in the bottom pane.
2. When **Symbol - Watch-xxxxxx** is double-clicked, for example, the **Test Comparison Details** dialog box appears as shown below. This dialog box shows the values of both the saved test-image information (Src=xxxxxx) and the result of this test (Dest=xxxxxx) regarding elements in arrays.



Test_Suite_Demo

Symbol - Watch-SimSessionSH-4

Information:

```
Failed at symbol a[6], Src = (long)H'00002704{70000018}, Dest = (long)H'0000(
```

3. Clicking the **Export** button saves the test result into a text file.
4. Click the **OK** button.

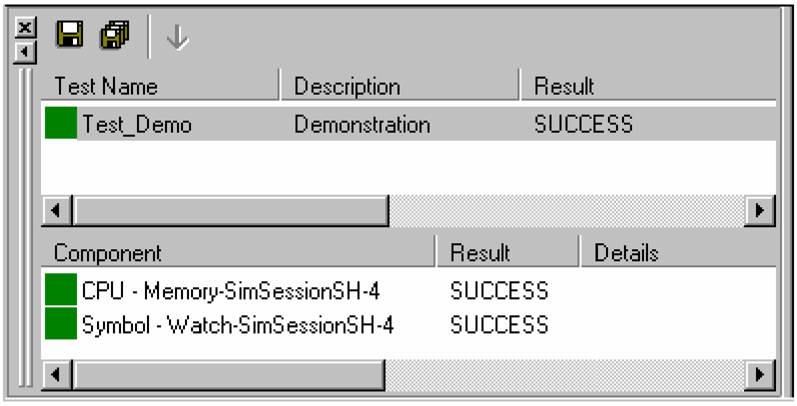
1.13.10 Step 9: Modifying the program back and executing the test again

Modify the program to restore the number of times to store data in the array and then execute the test.

1. Return the number from 8 to 10 and build the program.
2. Click the **Save File** button (💾) on the **Editor** toolbar.
3. Right-click on the “Test_Demo” test icon to display a pop-up menu.
4. Select **Run Test Case** to start the test.
5. When the test is completed, the **Test Browser** window opens.

1.13.11 Step 10: Viewing the test result (matched)

You can view the test result in the Test Browser window. Make sure that the test result now matches the saved test-image information.



Test Name	Description	Result
🟩 Test_Demo	Demonstration	SUCCESS

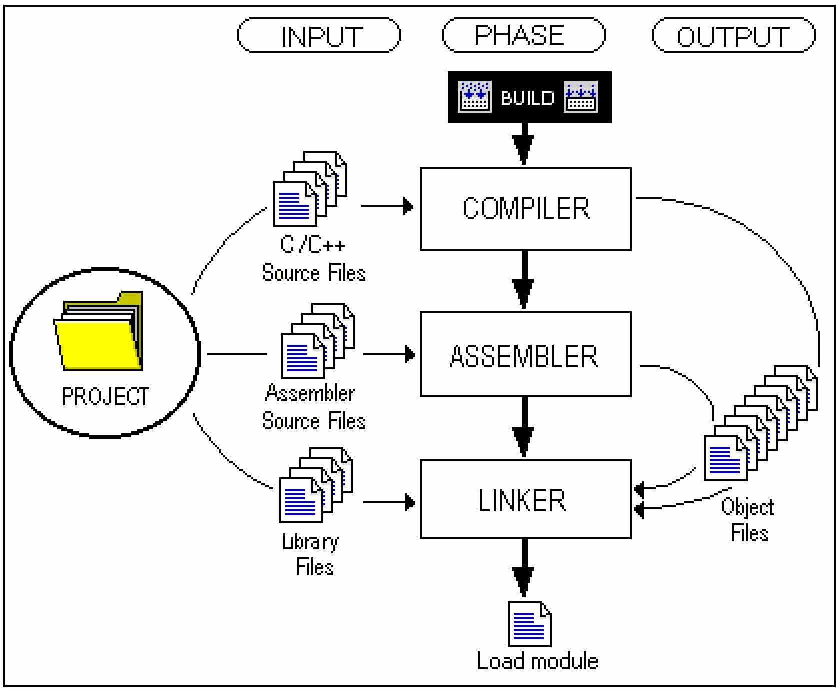
Component	Result	Details
🟩 CPU - Memory-SimSessionSH-4	SUCCESS	
🟩 Symbol - Watch-SimSessionSH-4	SUCCESS	

2. Build Basics

This chapter explains the general basic functions of the High-performance Embedded Workshop (HEW) whilst the more advanced features can be found in chapter 3, Advanced Build Features.

2.1 The build process

The typical build process is outlined in the figure below. This may not be the exact build process that your installation of HEW will use, as it depends upon the tools that were provided with your installation of HEW (you may not have a compiler, for instance). In any case, the principles are the same – each phase of the build takes a set of project files and builds them; if every file builds successfully then the next phase is executed.

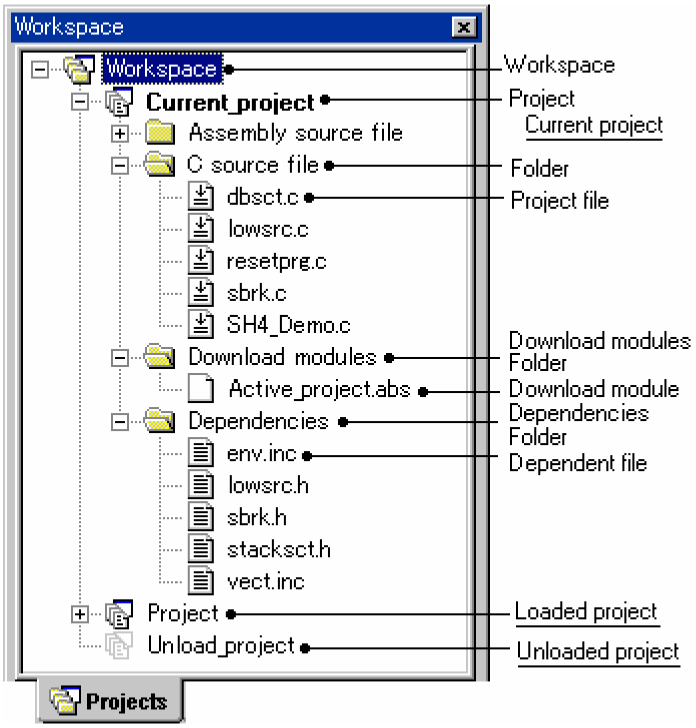


In the example shown in the figure above, the Compiler is the first phase, the Assembler is the second phase and the Linker is the third and final phase. During the Compiler phase, the C/C++ source files from the project are compiled in turn. During the Assembler phase, the assembler source files are assembled in turn. During the Linker phase, all library files and output files from the Compiler and Assembler phases are linked together to produce the load module.

The build process can be customized in several ways. For instance, you can add your own phases, disable phases, delete phases and so on. These advanced build issues are left to chapter 3, Advanced Build Features.

2.2 Configuring the Projects tab of the Workspace window

The **Projects** tab shows the current workspace, projects and files. You can quickly open any project file or dependent file by double-clicking on its icon.



Right-clicking on a workspace opens a pop-up menu containing the available options.

Pop-up Menu Options	Macro Recording	Function
Insert Project	-	Adds a project to workspace.
Load All Projects	-	Loads all projects to workspace.
Collapse	-	Collapses the tree below a workspace icon.
Clean All Projects	●	Deletes intermediate and output files from configurations in all projects in this workspace.
Version Control	-	Executes a version control system.
Configure View	-	Configures the workspace view.
Properties	-	Displays workspace properties.

Right-clicking on a project opens a pop-up menu containing the available options.

Pop-up Menu Options		Macro Recording	Function
Build	Build	●	Builds out of date project files
	Build All	●	Builds project files, regardless of whether the project files are out o date.
	Clean Current Project	●	Deletes intermediate and output files from the current configuration in this project.
	Update All Dependencies	-	Updates all dependencies.
Set as Current Project		●	Sets as current project.
Remove Project		-	Removes a project from workspace.
Unload Project OR Load Project		-	Unloads a project OR Loads a project.
Add Files		-	Adds files to a project.
Remove Files		-	Removes files from project.
Add Folder		-	Add folder to a project.
Expand/Collapse		-	Expands or collapses the tree below a project icon.
Version Control		-	Executes a version control system.
Configure View		-	Configures the workspace view.
Properties		-	Displays project properties.

Right-clicking on a folder (other than **Download module** and **Dependencies**) opens a pop-up menu containing the available options.

Pop-up Menu Options	Macro Recording	Function
Add Folder	-	Add a custom folder.
Rename Folder	-	Rename a folder.
Remove Folder	-	Remove a folder.
Configure View	-	Configures the workspace view.

Right-clicking on a project file opens a pop-up menu containing the available options.

Pop-up Menu Options	Macro Recording	Function
Open <file name>	-	Opens a file in the HEW editor.
Open <file name> in external editor ***1**	-	Opens a file in the external editor.
Build <file name>	●	Builds a file.
Build Options	-	Sets build options.
Add File	-	Adds files to a project.
Remove File	-	Removes files from project.
Exclude Build OR Include Build	-	Excludes a project file from build OR Includes a project file in build.
Version Control	-	Executes a version control system.
Configure View	-	Configures the workspace view.
Show Differences	-	Compares files.
Properties	-	Displays file properties.

***1**. This option is available only when the **Use external editor** checkbox is selected in the **Editor** tab of the **Options** dialog box opened via [**Setup->Options**].

Right-clicking on the **Download modules** folder opens a pop-up menu containing the available options (Available when the debugger is connected).

Pop-up Menu Options	Macro Recording	Function
Download all module	●	Downloads all modules in the **Download modules** list on the **Target** tab of the **Debug Settings** dialog box (i.e. download modules shown under the **Download Modules** folder in the **Projects** tab of the workspace window).
Download A New Module	●	Opens the **Download Module** dialog box, which allows you to add download modules.
Debug Settings	-	Opens the **Debug Settings**, which allows you to modify the debug settings.
Configure View	-	Configures the workspace view.

Right-clicking on a download module opens a pop-up menu containing the available options (Available when the debugger is connected).

Pop-up Menu Options	Macro Recording	Function
Download	●	Downloads modules.
Download (Debug Data Only)	●	Downloads modules (debug data only).
Unload	●	Unloads modules.
Download A New Module	●	Opens the **Download Module** dialog box, which allows you to add download modules.
Remove	-	Removes the selected modules.
Debug Settings	-	Opens the **Debug Settings** dialog box, which allows you to modify the debug settings.
Configure View	-	Configures the workspace view.
Relocate Module ***2**	-	Opens a dialog box which allows you to select one directory for relocating all files retrieved from the download module.
Properties	-	Shows the selected download module setting in the **Download Module** dialog box. This download module setting can be modified. The download module will be unloaded if it has already been downloaded.

***2**. If you use a debug-only project (i.e., "Debugger only - xxxxxx"), this menu item will be displayed.

Right-clicking on a file retrieved from the download module opens a pop-up menu containing the available options (Available only when you use the debug-only project).

Pop-up Menu Options	Macro Recording	Function
Open <file name>	-	Opens a file in the HEW editor.
Add File	-	Adds files to a project.
Configure View	-	Configures the workspace view.
Relocated file(s)	-	Opens a dialog box which allows you to select the same short filename retrieved from the download module at a new location.
Properties	-	Displays file properties.

Right-clicking on the Dependencies folder opens a pop-up menu containing the available options.

Pop-up Menu Options	Macro Recording	Function
Configure View	-	Configures the workspace view.

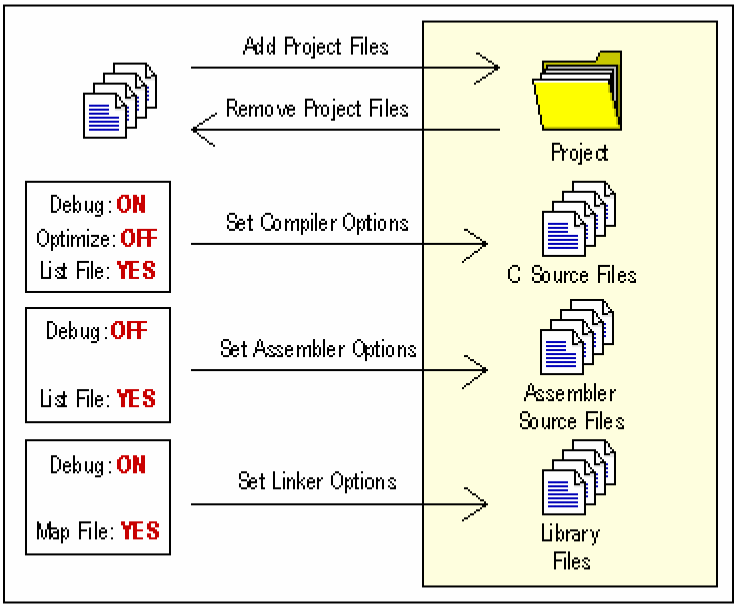
Right-clicking on a dependence file opens a pop-up menu containing the available options.

Pop-up Menu Options	Macro Recording	Function
Version Control	-	Executes a version control system.
Configure View	-	Configures the workspace view.
Show Differences	-	Compares files.
Properties	-	Displays file properties.

For details on "Current project", "Loaded project" and "Unloaded project", see section 2.12, Setting the current project.

2.3 Project files

In order for the HEW to be able to build your application, you must first tell it which files should be in the project and how each file should be built (see the figure below).



2.3.1 Adding files to a project

Before building an application, select files to configure the application.

To add files to a project

1. Files can be added to the active project only. Select one of the following operations:
   - Select the [**Project->Add Files**] menu option, **OR**
   - Right-click on the project within the **Projects** tab of the workspace window and select **Add Files** from the pop-up menu, **OR**
   - Press the "Insert" key when the **Projects** tab of the workspace window is selected.
2. The **Add files to project 'project name'** dialog box will be displayed.
3. Select the file(s) you wish to add to the project.
4. A **Relative Path** checkbox is available at the bottom of the **Add files to project '<project name>'** dialog box. Selecting this checkbox allows the files to be relative project files. These project files are relative to the workspace file and can also be placed outside the workspace structure. Even if you re-allocate the entire source tree, the HEW is able to check the relative position of the files to find them. By default, this checkbox is selected.
5. A **Hide Project Files** checkbox is available at the bottom of the **Add files to project '<project name>'** dialog box. Selecting this checkbox only shows the files not added to the current project. By default, this checkbox is not selected.
6. Click the **Add** button.

There are other ways to add files to a project

- Right-clicking on an open file in the **Editor** window displays a pop-up menu. If the file is already in the project then the **Add File To Project** menu option is disabled. Select the **Add File To Project** menu option to add the file to the current project.
- In the HEW it is also possible to 'drag and drop' files from Windows® Explorer onto the Projects tab of the workspace window. For details, see section 2.3.2, Drag and drop of files and folders.

Note:

If you add a file that has an unrecognized file type to the project, then the file will be added to the project, but certain functions will be disabled for this file. When you double-click on a file with an unrecognized file type in the Workspace window, the open operation is passed to the Windows® operating system (instead of opening the file in the editor). The default 'Open' operation is then carried out as if the file was double-clicked in Windows® Explorer. To view the currently defined extensions, use the **File Extensions** dialog box. See section 2.5, File extensions and file groups, for further information.

2.3.2 Drag and drop of files and folders

It is possible to 'drag and drop' files or folders from Windows® Explorer onto the **Projects** tab of the workspace window. These files or folders can only be dropped onto a project in the active project or a user folder in the active project. Note, however, that you cannot drag and drop files or folders onto multiple folders.

When you drag and drop files

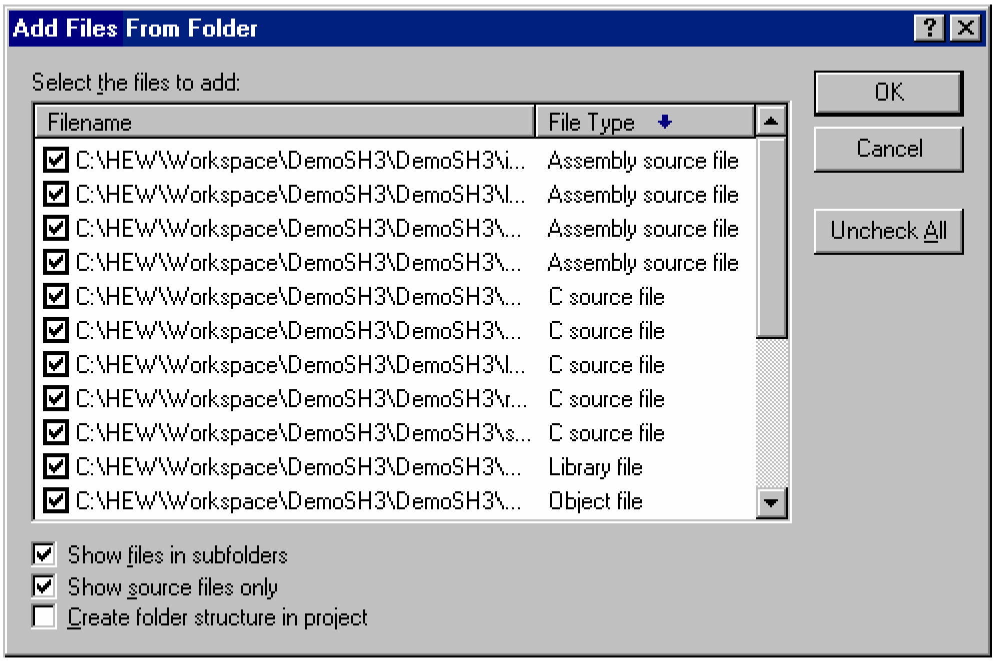
The behavior depends on the destination.

- A file dropped onto the active project is added into the group folder of the file type. If a suitable group folder does not exist, the HEW creates it. For example, if you drop the file "test.c" onto the active project, this file will be added to the group folder "C source file".

- A file dropped onto a user folder is directly added to this folder. Even if a file with the same name is already in the folder, the new file can be added to the folder only when these two files have different paths.
- If a file of an unrecognized file type is added to the project, this file is directly added to the project.

To drag and drop folders

Drag and drop of a folder from the Windows® Explorer opens the **Add Files From Folder** dialog box, which allows you to select the files in the folder that you wish to add to the project. This dialog box shows the names of all files included in the folder that was dropped. **Filename** and **File Type** show the full paths of files and file types, respectively.



Select the files to add:

Filename	File Type
☑ C:\HEW\Workspace\DemoSH3\DemoSH3\i...	Assembly source file
☑ C:\HEW\Workspace\DemoSH3\DemoSH3\l...	Assembly source file
☑ C:\HEW\Workspace\DemoSH3\DemoSH3\...	Assembly source file
☑ C:\HEW\Workspace\DemoSH3\DemoSH3\...	Assembly source file
☑ C:\HEW\Workspace\DemoSH3\DemoSH3\...	C source file
☑ C:\HEW\Workspace\DemoSH3\DemoSH3\...	C source file
☑ C:\HEW\Workspace\DemoSH3\DemoSH3\l...	C source file
☑ C:\HEW\Workspace\DemoSH3\DemoSH3\r...	C source file
☑ C:\HEW\Workspace\DemoSH3\DemoSH3\s...	C source file
☑ C:\HEW\Workspace\DemoSH3\DemoSH3\...	Library file
☑ C:\HEW\Workspace\DemoSH3\DemoSH3\...	Object file

☑ Show files in subfolders
☑ Show source files only
☐ Create folder structure in project

1. In the dialog box, the files are initially sorted by the file type in alphabetical order. If you click a column header (of file names or types), the files will be sorted by the file name or type.
2. You can select multiple files by clicking. If you then click on one of the checkboxes for the selected files or press the Space key, checkboxes for all of the selected files will be switched ON or OFF (depending on the previous state).
3. If the **Show files in subfolders** checkbox is selected, the dialog box also shows the files within subfolders under the folder that was dropped. Otherwise the dialog box only shows the files within the folder that was dropped. By default, this checkbox is selected.
4. If the **Show source files only** checkbox is selected, the dialog box only shows the files having the extension specified for the source files in this project (e.g. C source files). Otherwise the dialog box shows all files. If the HEW cannot recognize the file type, "Unknown type" will be shown in the **File Type** column. By default, this checkbox is selected.
5. If the **Create folder structure in project** checkbox is selected, files are added to the tree where the folder was dropped. For example, when a folder "data" containing some files is dropped onto the tree, the files are added into a user folder "data", which is newly created.
   If the **Create folder structure in project** checkbox is not selected, when you drop a folder onto a user folder, files in the folder are added into this user folder. When a folder is dropped onto the active project tree, files in the folder are added into the group folder of the file type (e.g. a file "main.c" is added into the group folder "C source file"). If a suitable group folder does not exist, the HEW creates it. Files of an unknown type dropped onto the project are directly added to the project.
   By default, this checkbox is not selected.

6. Clicking the **Uncheck All** button deselects all checkboxes, while clicking the **Check All** button selects all checkboxes.
7. Clicking the OK button adds all selected files, which have tick marks in their checkboxes, to the project. Next time this dialog box is opened, it shows the most recently used setting.

2.3.3 Removing files from a project

There are three ways of removing files from a project

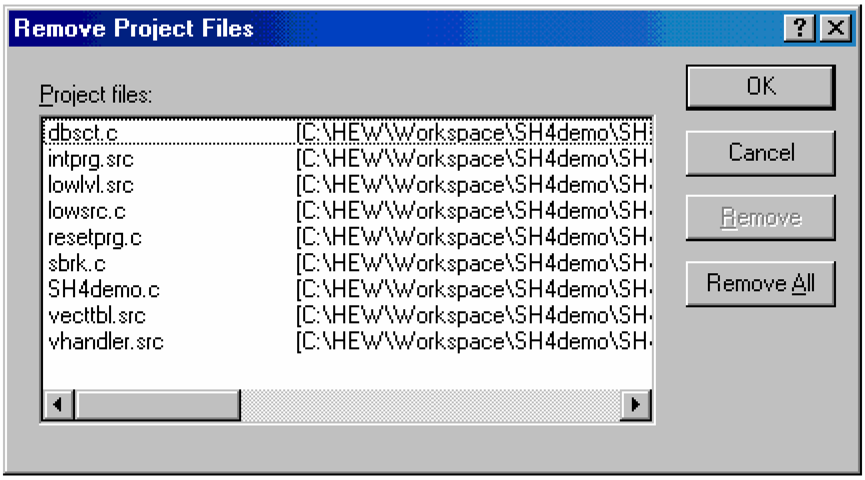
- Files can be individually removed from a project,
- A selection of files can be removed,
- All files can be removed.

To remove a file(s) from a project

1. Select one of the following operations:

- Select the [**Project->Remove Files**] menu option, **OR**
- Right-click on the project within the **Projects** tab of the workspace window and select **Remove Files** from the pop-up menu.

2. The **Remove Project Files** dialog will be displayed.
3. Select the file(s) that you want to remove from the **Project files** list.
4. Click the **Remove** button to remove the file(s), or click the **Remove All** button to remove all files from the list.
5. Click the OK button to remove the files from the project.



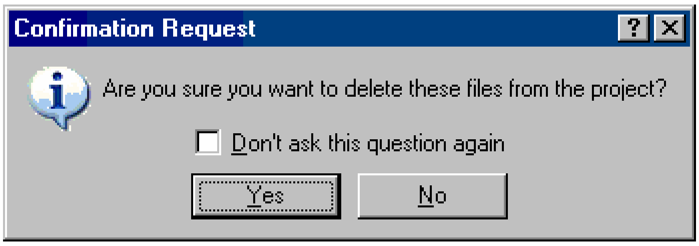
Remove Project Files

Project files:

dbsct.c	[C:\HEW\Workspace\SH4demo\SH
intprg.src	[C:\HEW\Workspace\SH4demo\SH
lowlvl.src	[C:\HEW\Workspace\SH4demo\SH
lowsrc.c	[C:\HEW\Workspace\SH4demo\SH
resetprg.c	[C:\HEW\Workspace\SH4demo\SH
sbrk.c	[C:\HEW\Workspace\SH4demo\SH
SH4demo.c	[C:\HEW\Workspace\SH4demo\SH
vecttbl.src	[C:\HEW\Workspace\SH4demo\SH
vhandler.src	[C:\HEW\Workspace\SH4demo\SH

To remove selected files from a project using the Projects tab of the Workspace window

1. Select the files that you want to remove in the **Projects** tab of the Workspace window. Multiple files can be selected by holding down the SHIFT or CTRL key.
2. Press Delete.
3. A confirmation dialog box opens for you to select whether or not to delete the selected files from the project. To delete the selected files, select Yes. Otherwise select No.



Confirmation Request

Are you sure you want to delete these files from the project?

□ Don't ask this question again

If you do not wish to open this confirmation dialog box, select the **Don’t ask this question again** checkbox. To open this dialog box again, select [**Setup->Options**] to open the **Options** dialog box. Select the **Delete file from project** checkbox on the **Confirmation** tab. By default, this checkbox is selected.

2.3.4 Excluding a project file from build

A file in a project can be excluded from build on a configuration by configuration basis.

To exclude a project file from build

1. Click on the file that you want excluded from build, in the Projects tab of the Workspace window.
   It is also possible to select several files by using a mouse or inputs through the keyboard as follows:
   - To select several files:
     Click the files while pressing the CTRL key.
   - Select several files as a range:
     Click a file as the start of the range. Then keep pressing the SHIFT key and click another file as the end of the selected range.
2. Take either of the two ways listed below. A red cross ( ) will appear on the file’s icon, and the file will be excluded from build.
   - Select **Exclude Build <file>**, where <file> is the name of the selected file, from the pop-up menu opened by right-clicking. <file> is not displayed if several files are selected.
   - Select [**Build->Include/Exclude Build**] menu option.

2.3.5 Including a project file in build

A file that has been excluded from build can be included again.

To include a project file in build

1. Click on a file that has previously been excluded from build, in the Projects tab of the Workspace window.
   It is also possible to select several files by using a mouse or inputs through the keyboard as follows:
   - To select several files:
     Click the files while pressing the CTRL key.
   - Select several files as a range:
     Click a file as the start of the range. Then keep pressing the SHIFT key and click another file as the end of the selected range.
2. Take either of the two ways listed below. The red cross will be removed from the file’s icon ( ), and the file will be included in build.

- Select **Include Build <file>**, where <file> is the name of the selected file, from the pop-up menu opened by clicking the right-mouse button. <file> is not displayed if several files are selected.
- Select [**Build->Include/Exclude Build**] menu option.

2.4 User folders in the workspace

In the High-performance Embedded Workshop it is possible to add folders to the Projects tab of the workspace window. This allows you to logically group your files into certain areas within a project. The folder can be set to any name and this is entered in a dialog box. The operations listed below are not allowed, however, for folders **Download modules** and **Dependencies** that are automatically displayed.

To add a user folder

1. Select the project on the Projects tab of the workspace window.
2. Right-click and select Add Folder.
3. Enter the name in Folder of the Add Folder dialog box.
4. Click OK.
5. You can now drag and drop files into this folder to group them logically.

To add a sub-folder

1. Select the folder on the Projects tab of the workspace window.
2. Right-click and select Add Folder.
3. Enter the name in Folder of the Add Folder dialog box
4. Click OK.
5. You can now drag and drop files into this folder to group them logically.

To drag and drop a folder

Drag a folder in the active project from the Projects tab of the workspace window.

- If the folder is dropped onto the active project, the folder is directly added into the project.
- If the folder is dropped onto a user folder in the active project, the folder is added into the user folder.

You can also drag folders from the Windows® Explorer and drop it into the Projects tab of the workspace window. For details, see section 2.3.2, Drag and drop of files and folders.

To remove a user folder

1. Select the folder on the Projects tab of the workspace window.
2. Select one of the following operations (the folder must be empty):

- Press the Delete key.
- Right-click and select **Remove Folder** from the pop-up menu.

To modify a user folder name

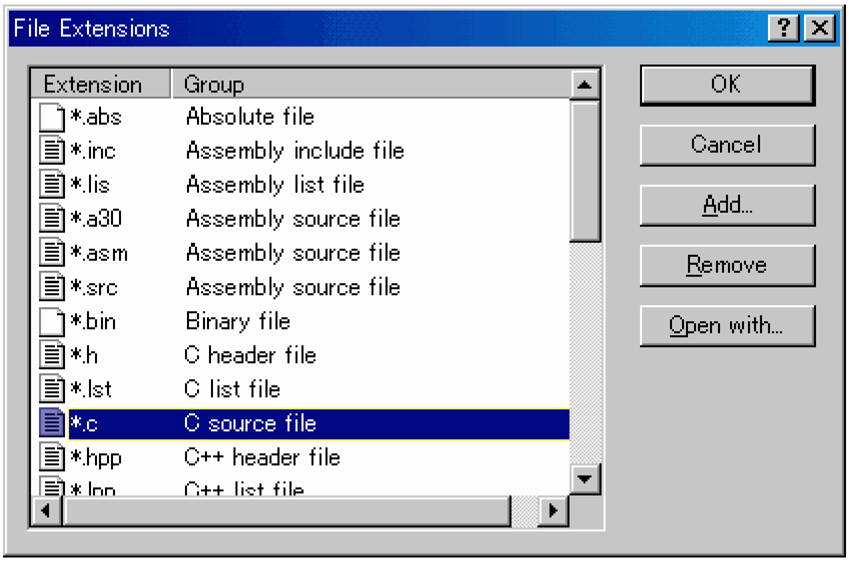
1. Select the folder on the Projects tab of the workspace window.
2. Right-click and select **Rename Folder** from the pop-up menu.

3. Enter the new name in Folder of the Rename Folder dialog box.
4. Click OK.

2.5 File extensions and file groups

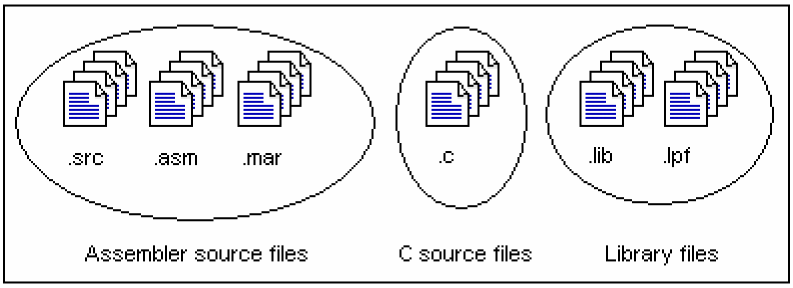
The HEW can identify files by their extension. The system defines certain extensions depending upon the tools that are being used. For example, if you are using a compiler then the .c extension will be in the ‘C source file’ group and will be used as input to the compiler phase. Additionally, the HEW allows you to define your own extensions. For example, if the project you are developing uses assembler source files the default extension may be .src. If you would like to use a different extension instead of .src (e.g. .asm) then you can define a new extension and request that the HEW treats it in the same way as a .src file.

File extensions and file groups can be viewed and modified via the **File Extensions** dialog box, which is invoked by selecting the [**Project->File Extensions**] menu option. This dialog box displays all the extensions and file groups that are defined within the current workspace.



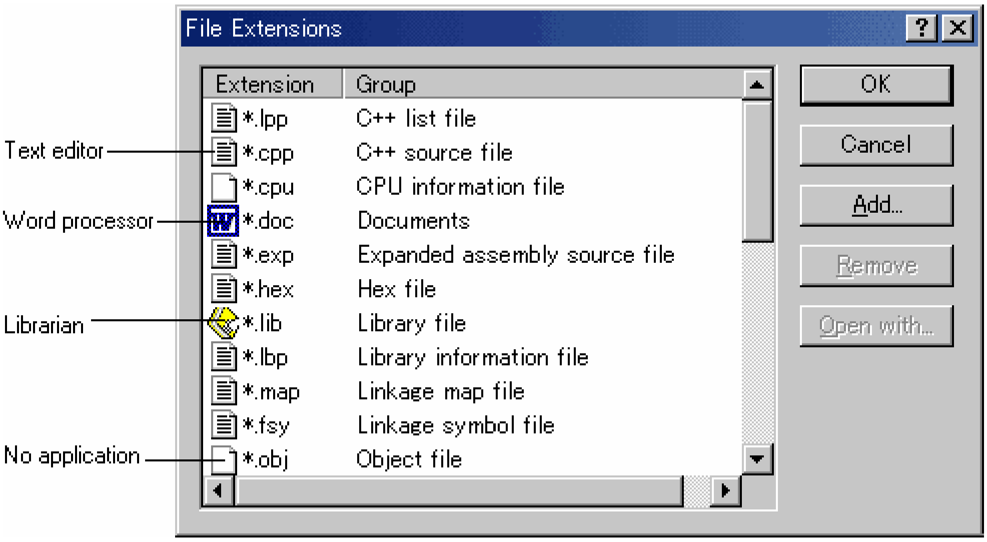
Extension	Group
*.abs	Absolute file
*.inc	Assembly include file
*.lis	Assembly list file
*.a30	Assembly source file
*.asm	Assembly source file
*.src	Assembly source file
*.bin	Binary file
*.h	C header file
*.lst	C list file
*.c	C source file
*.hpp	C++ header file
*.lpp	C++ list file

The **File Extensions** list is divided into two columns. On the left are the file extensions, and on the right are the file groups associated with the extensions. Many file extensions can belong to the same group. For example, assembler source files may have several extensions in a single project (e.g. .src, .asm, .mar etc.).



2.5.1 Associating an application with a file group

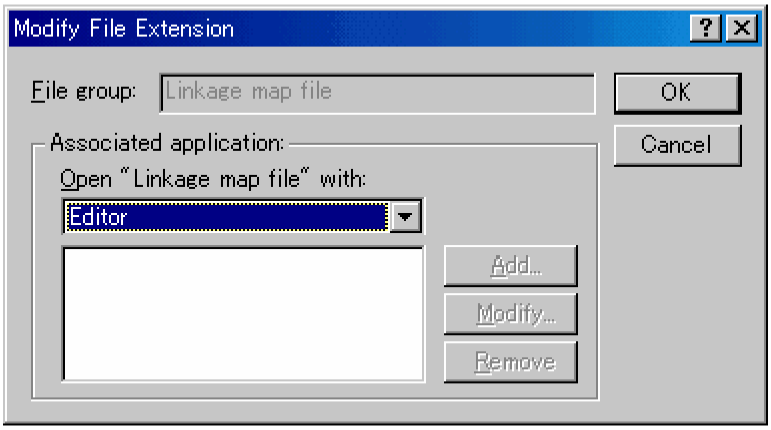
In addition to opening a file with the editor, the **File Extensions** dialog box allows you to associate any application with any file group so that when you double-click on a file in the **Projects** tab of the **Workspace** window, the appropriate application is launched with the file.



Extension	Group
*.lpp	C++ list file
*.cpp	C++ source file
*.cpu	CPU information file
*.doc	Documents
*.exp	Expanded assembly source file
*.hex	Hex file
*.lib	Library file
*.lbp	Library information file
*.map	Linkage map file
*.fsy	Linkage symbol file
*.obj	Object file

To associate an application with a file group

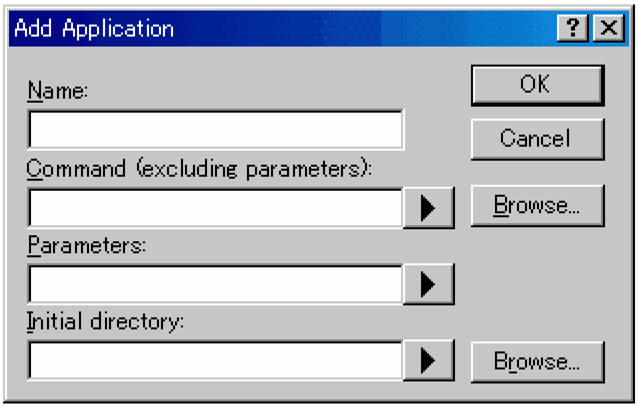
1. Select the [**Project->File Extensions**] menu option. The **File Extensions** dialog box will be displayed.
2. Select the file group that you want to associate from the file extensions list.
3. Click the **Open with** button. The **Modify File Extension** dialog box will be displayed.



File group: Linkage map file

Open "Linkage map file" with: Editor

4. Select **None** to remove any association, **Editor** to open this type of file in the internal/external editor, or select **Other** to open this type of file with a specific application. If you select **Other**, you can either specify a new application, or select any previously defined application from the drop-down list. Click the **Add** button to define a new application. The **Add Application** dialog box will be displayed.



Enter the name of the tool into the **Name** field. Enter the full path to the tool in the **Command** field (do not include any parameters). Enter the parameters that are required to open a file into the **Parameters** field. Be sure to use the $(FULLFILE) placeholder to specify the location of the file (see Reference 5, Placeholders, for more information on placeholders). Enter the initial directory in which you would like the application to run into the **Initial directory** field. Click the **OK** button to finish creating the application. Click the **Modify** button to modify an application. The **Modify Application** dialog box will be displayed. This dialog is the same as the **Add Application** dialog box described above except that the **Name** field is read-only. Modify the settings as desired and then click the OK button.

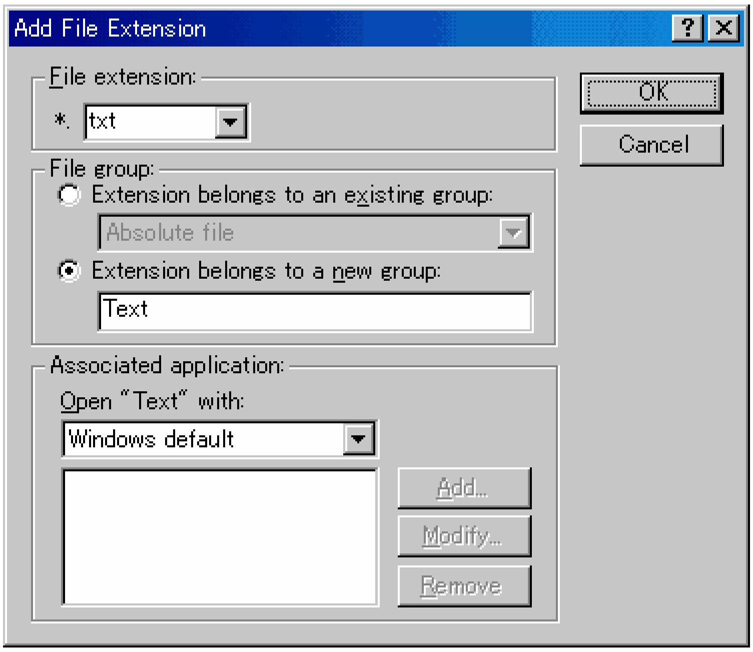
5. Click the OK button to set the application for the selected file group.

2.5.2 Creating a new file extension and file group

If you want to manage files that are not, by default, recognized by the HEW (e.g. documents) then you need to create a new extension and a new file group.

To create a new file extension in a new file group

1. Select the [**Project->File Extensions**] menu option. The **File Extensions** dialog box will be displayed.
2. Click the **Add** button. The **Add File Extension** dialog box will be displayed.



Add File Extension

File extension:

*. txt

File group:

○ Extension belongs to an existing group:

Absolute file

● Extension belongs to a new group:

Text

Associated application:

Open "Text" with:

Windows default

3. Enter the extension that you want to define into the **File extension** field. Use only alphanumeric and an underscore as characters of a file extension string. The drop-down list contains all extensions that are undefined in the current project. Selecting one of these extensions will add the text to the file extension field automatically.
4. Select the **Extension belongs to new group** option and enter a description that defines this new file group.
5. At this stage it is possible to change the associated application. There are four available choices in the **Open "<extension group>" with** drop-down list:
   - Editor
   - None
   - Other
   - Windows default

If **Editor** is selected, the **Open File** function in the workspace window causes the file to be opened in the HEW editor. If **None** is selected then the **Open File** operation is disabled when it is attempted. Selecting **Other** allows you to configure another tool for the **Open File** operation. See section 2.5.1, Associating an application with a file group, for more details. If **Windows default** is selected then the **Open File** function in the **Workspace** window passes the **Open File** operation to the Windows® operating system. This then selects the default behavior for this file extension as defined in Windows® Explorer.

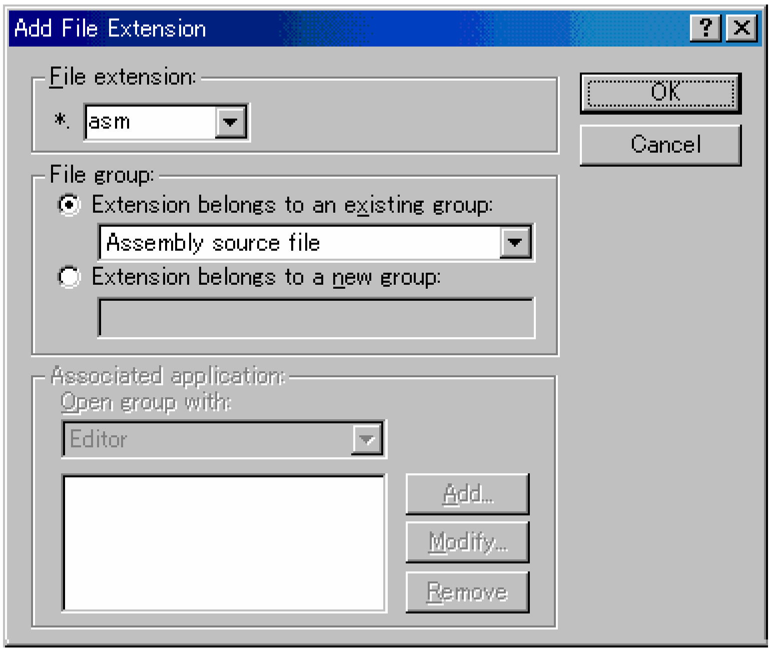
6. Click the OK button to add the extension to the **File Extensions** list.

2.5.3 Creating a new file extension

If your files use a different extension from those accepted by the HEW for a given phase (e.g. your assembler source files are .asm but the HEW only recognizes .src), then you need to create a new extension and add it to an existing file group. This process is described below.

To create a new file extension in an existing file group

1. Select the [**Project->File Extensions**] menu option. The **File Extensions** dialog box will be displayed.
2. Click the **Add** button. The **Add File Extension** dialog box will be displayed.



File extension: *. asm

File group:
● Extension belongs to an existing group: Assembly source file
○ Extension belongs to a new group:

Associated application:
Open group with: Editor

3. Enter the extension that you want to define into the **File extension** field. Use only alphanumeric and an underscore as characters of a file extension string. The drop-down list contains all extensions that are undefined in the current project. Selecting one of these extensions will add the text to the file extension field automatically.
4. Select the **Extension belongs to an existing group** option and select the group to which you would like to add this new extension.
5. Click the OK button to add the extension to the **File Extensions** list.

2.6 Setting build options

Once you have added the necessary files to the project, the next step is to instruct the HEW on how to build each file. To do this, you will need to select a menu option from the **Build** menu. The contents of this menu depend upon which tools you are using.

To set options for a build phase

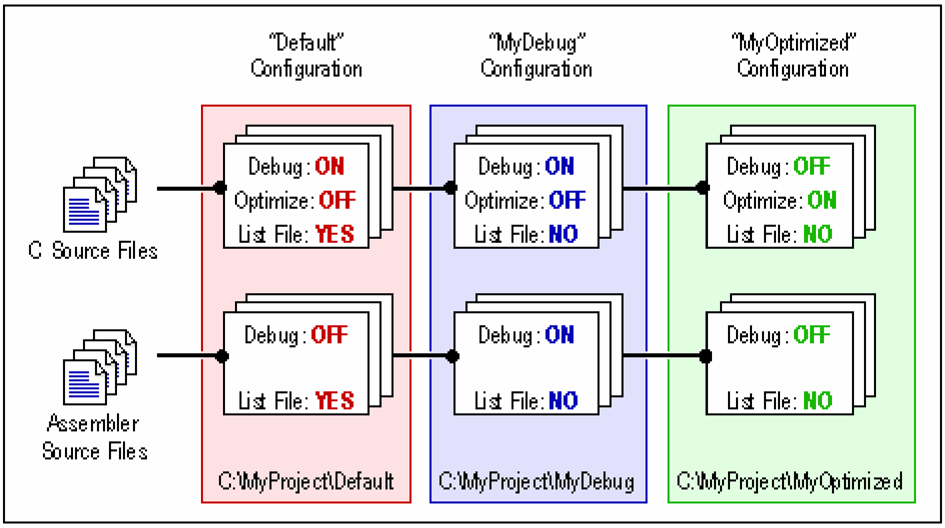
1. Select the **Build** menu and select the phase whose options you would like to modify.
2. A dialog box will be displayed allowing you to specify the options.
3. After making your selections, click the OK button to set them.

To obtain further information, use the context-sensitive help button or select the area in which you need assistance and press F1.

2.7 Build configurations

The HEW allows you to store all of your build options into a build configuration, i.e. you can ‘freeze’ all of the options and give them a name. Later on, if you select that configuration, all options for all of the build phases will be restored. These configurations also allow the user to specify debugger settings for a build configuration. This means that each configuration can be targeted at a different end platform.

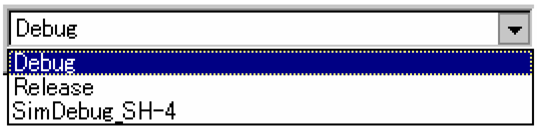
The figure below shows three configurations: **Default**, **MyDebug** and **MyOptimized**. In the first configuration, **Default**, each phase (compile and assemble) is set to its standard settings. In the second configuration, **MyDebug**, each file is being built with debug information switched on. In the third configuration, **MyOptimized**, each file is being built with optimization on full and without any debug information. The developer of this project can select any of those configurations and build them without having to return to the options dialogs to set them again.



2.7.1 Selecting a build configuration

To select the current configuration

1. Select the [**Build->Build Configurations**] menu option. The **Build Configurations** dialog box will be displayed.
2. Select the build configuration that you want to use from the **Current Configuration** drop-down list.
3. Click the OK button.

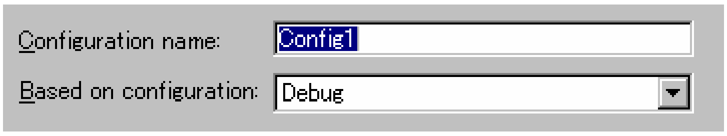


You can also select a different build configuration by selecting it from the Current Configuration drop-down list on the **Standard** toolbar.

2.7.2 Adding a new build configuration

To add a new build configuration

1. Select the [**Build->Build Configurations**] menu option. The **Build Configurations** dialog box will be displayed.
2. Click the **Add** button. The **Add Configuration** dialog box will be invoked.
3. Enter the new build configuration name into the **Configuration name** field. As you enter the new build configuration name, the directory underneath changes to reflect the configuration directory that will be used. Select one of the existing build configurations, on which you want to base the new build configuration, from the **Based on configuration** drop-down list. Click the **OK** buttons on both dialogs to complete the creation of the new build configuration.



Configuration name: Config1

Based on configuration: Debug

2.7.3 Removing a build configuration

To remove a build configuration

1. Select the [**Build->Build Configurations**] menu option. The **Build Configurations** dialog box will be displayed.
2. Select the build configuration to remove and click the **Remove** button.
3. Click the OK button to close the **Build Configurations** dialog box.

2.8 Building a project

2.8.1 Building individual files

The High-performance Embedded Workshop lets you build project files individually.

To build an individual file

1. Select the file to build from the **Projects** tab of the **Workspace** window.
2. Select one of the following operations:
    - Click the **Build File** toolbar button (), **OR**
    - Select the **Build <file>** option from pop-up menu, **OR**
    - Select the [**Build->Build File**] menu option, **OR**
    - Press CTRL+F7.

All output is redirected to the **Build** tab of the Output window.

2.8.2 Building a project

The **Build** option only compiles or assembles those files that have changed since the last build. Additionally, it will rebuild source files if they depend upon a file that has changed since the last build. For instance, if the file ‘TEST.C’ #include’s the file ‘HEADER.H’ and the latter has changed since the last build, the file ‘TEST.C’ will be recompiled.

To perform a build operation

Select one of the following operations:

- Click the **Build** toolbar button (), **OR**
- Press F7, **OR**
- Select the [**Build->Build**] menu option, **OR**
- Right-click on a project in the **Projects** tab of the **Workspace** window and select the [**Build->Build**] option from the pop-up menu.

The **Build All** option compiles and assembles all source files, irrespective of whether they have been modified or not, and links all of the new object files produced.

To perform a build all operation

Select one of the following operations:

- Click the **Build All** toolbar button (), **OR**
- Select the [**Build->Build All**] menu option, **OR**
- Right-click on a project in the **Projects** tab of the **Workspace** window and select the [**Build->Build All**] option from the pop-up menu.

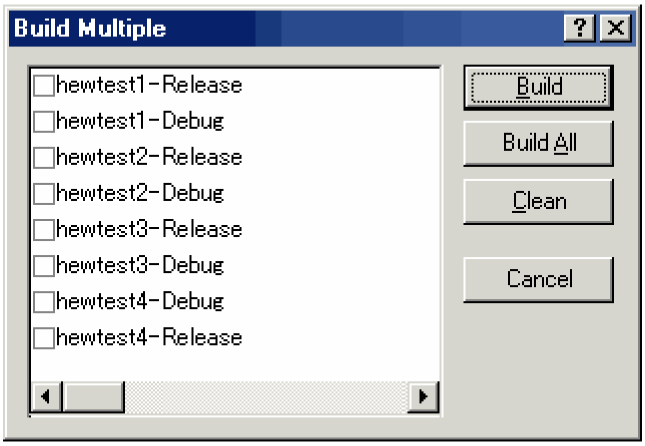
All output from a build or build all operation is redirected to the **Build** tab of the Output window. Both the Build and the Build All operations will terminate if any project files produce errors.

2.8.3 Building multiple projects

The High-performance Embedded Workshop allows you to build multiple projects and configurations at once.

To build multiple projects

1. Select the [**Build->Build Multiple**] menu option. The **Build Multiple** dialog box is displayed.
2. The **Build Multiple** dialog box gives you the choice of which projects and configurations to build. Select the check boxes next to the projects and configurations that you want to build. For example, in figure below if you wanted to build the entire "hewtest2" project you would check the "hewtest2-Debug" and the "hewtest2-Release" selections and leave all other check boxes unchecked.



□hewtest1-Release
□hewtest1-Debug
□hewtest2-Release
□hewtest2-Debug
□hewtest3-Release
□hewtest3-Debug
□hewtest4-Debug
□hewtest4-Release

3. When you are happy with your chosen selection, click the **Build** button and the HEW will build the selected projects and configurations.
4. If you want to build all of the projects, click the **Build All** button. This will automatically select all projects and configurations, and build them all.

5. If you also wish to delete intermediate and output files of the selected project or configuration, click the **Clean** button. This does not execute the build.
6. Results from the build are displayed in the Build tab of the Output window in the same way as in a normal build process.

2.8.4 Stopping tool execution

The High-performance Embedded Workshop allows you to halt the build process once it is under way.

To stop tool execution

1. Select one of the following operations:
   - Click the **Stop Tool Execution** toolbar button (), **OR**
   - Select the [**Build->Stop Tool Execution**] menu option.
2. The HEW will attempt to stop the tool immediately.
3. The 'Build Stopped by User' message appears in the Build tab of the Output window.

Note:

Do NOT assume that any output from the tool you terminated is valid. It is recommended that you delete any output files produced and ensure that the phase is executed again.

2.8.5 Deleting intermediate and output files produced in building

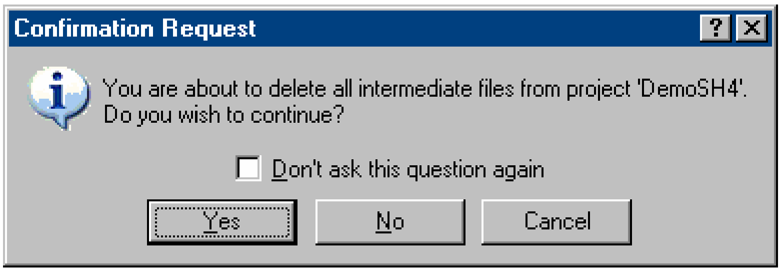
You can delete intermediate and output files (produced in building) included in the current configuration in the project.

To delete intermediate and output files included in the current configuration in the project

Select one of the following operations:

- Select [**Build->Clean Current Project**], **OR**
- Right-click on a project in the Projects tab of the workspace window. Select [**Build->Clean Current Project**] from the pop-up menu.

A confirmation dialog opens for you to select whether or not to delete all intermediate and output files produced in building. To delete all files, select Yes. Otherwise select No or Cancel.



Confirmation Request

You are about to delete all intermediate files from project 'DemoSH4'. Do you wish to continue?

□ Don't ask this question again

If you do not wish to open this confirmation dialog box, select the **Don't ask this question again** checkbox.

To open the confirmation dialog box again

1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.

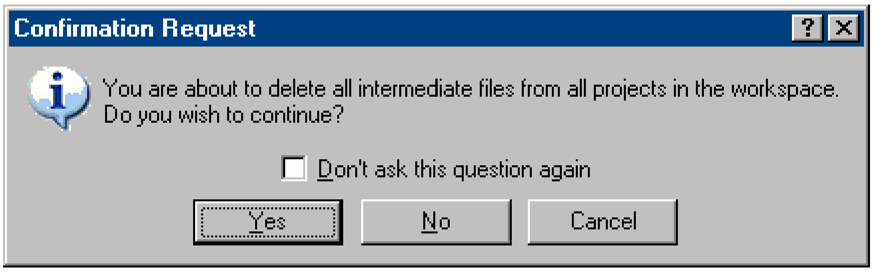
2. Select the **Confirmation** tab.
3. Select the **Clean project** checkbox. This checkbox is selected by default.
4. Click OK.

To delete intermediate and output files in all projects of the workspace

Select one of the following operations:

- Select [**Build->Clean All Projects**], **OR**
- Right-click on a workspace in the **Projects** tab of the workspace window. Select [**Clean All Projects**] from the pop-up menu.

A confirmation dialog opens for you to select whether or not to delete all intermediate and output files produced in building. To delete all files, select Yes. Otherwise select No.



You are about to delete all intermediate files from all projects in the workspace.
Do you wish to continue?

□ Don't ask this question again

If you do not wish to open this confirmation dialog box, select the **Don't ask this question again** checkbox.

To open the confirmation dialog box again

1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Confirmation** tab.
3. Select the **Clean workspace** checkbox. This checkbox is selected by default.
4. Click OK.

The **Build** tab in the Output window shows all projects and configurations in which intermediate and output files have been deleted.

You can select multiple configurations and delete intermediate and output files produced in building. For details, see section 2.8.3, Building multiple projects.

Note:

Linkage list files (".MAP") and assembly-program list files (".LIS") are not deleted. The HEW deletes relocatable and absolute files. Also note that the extension for absolute files differs depending on the selected format.

2.8.6 Configuring the Build tab of the Output window

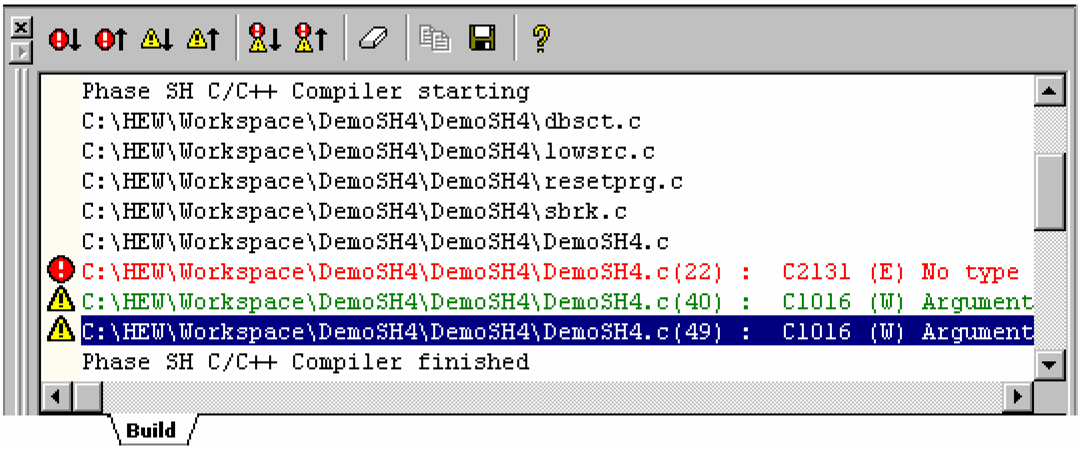
The **Build** tab shows the output from any build process (e.g. compiler, assembler and so on). If an error is encountered in a source file, an icon and the error message will be displayed in the **Build** tab, along with the source file name and line number.

If you click on a toolbar button or pop-up menu item associated with display of error messages, the line of the error message will be highlighted (with text in white and its background in dark blue) and the editor will show the source code of that line (if the line has any source code).

Double-clicking the line of an error message will also show the source code in the editor.

When you attempt an operation to display an error or warning, the status bar shows this error or warning message.

You can also customize the **Build** tab so that the texts in the lines of error messages will be highlighted in a color different from that of the texts in other lines. The figure below shows the error messages in red and warnings in green as an example.



```
  Phase SH C/C++ Compiler starting
  C:\HEW\Workspace\DemoSH4\DemoSH4\dbsct.c
  C:\HEW\Workspace\DemoSH4\DemoSH4\lowsrc.c
  C:\HEW\Workspace\DemoSH4\DemoSH4\resetprg.c
  C:\HEW\Workspace\DemoSH4\DemoSH4\sbrk.c
  C:\HEW\Workspace\DemoSH4\DemoSH4\DemoSH4.c
❗C:\HEW\Workspace\DemoSH4\DemoSH4\DemoSH4.c(22) :  C2131 (E) No type
⚠ C:\HEW\Workspace\DemoSH4\DemoSH4\DemoSH4.c(40) :  C1016 (W) Argument
⚠ C:\HEW\Workspace\DemoSH4\DemoSH4\DemoSH4.c(49) :  C1016 (W) Argument
  Phase SH C/C++ Compiler finished
```

Right-clicking displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up Menu Options	Toolbar button	Function
-	❗↓	Highlights the next occurrence of error and shows the source code in the editor.
-	❗↑	Highlights the previous occurrence of error and shows the source code in the editor.
-	⚠↓	Highlights the next occurrence of warning and shows the source code in the editor.
-	⚠↑	Highlights the previous occurrence of warning and shows the source code in the editor.
Display next Error/Warning/Info	❗⚠↓	Displays the editor that generated the next build error or warning.
Display previous Error/Warning/Info	❗⚠↑	Displays the editor that generated the previous build error or warning.
Help	?	Shows the help information about the line.
Go to Error/Warning/Info	-	Goes to the associated source line.
Clear Window	⌫	Clears the contents of the window.
Save	💾	Saves the contents of the window into a text file.
Copy	📋	Copies the selected contents onto the Windows® clipboard.
Toolbar display	-	Shows or hides the toolbar.
Customize toolbar	-	Customizes toolbar buttons.

The **Build** tab also shows an icon corresponding to the error message output by build execution.

Icon Name	Icon	Error Message Level
Build Error	❗	Error
Build Warning	⚠	Warning
Information	ⓘ	Information

You can also customize the **Build** tab of the Output window so that the texts in the lines of error messages will be highlighted in a color different from that of the texts in other lines.

To customize the current colors

1. Select [**Setup->Format Views**] to open the **Format Views** dialog box.
2. Select the **Output** item from the tree in the left-hand section of the dialog box and expand it.
3. Select the category for which you wish to change the color from the table below.

Category	Foreground Color of the Color Tab (Default)	Background Color of the Color Tab (Default)	Tab(s) to be Applied	Type of Output to be Applied
Text	SYSTEM	SYSTEM	All tabs	All
Build Error Text	Black	White	Build	Error
Build Warning Text	Black	White	Build	Warning
Information Text	Black	White	Build	Information

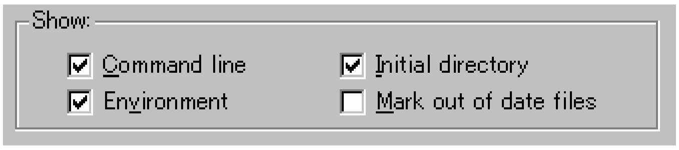
4. Change the selection in the **Foreground** and **Background** lists of the **Color** tab.
5. Click the OK button.

2.8.7 Controlling the content of the Build tab of the Output window

It is often useful to display extra information (such as the command line options that are being applied to a file) during a build. The HEW allows you to specify whether or not you want such options displayed in the **Build** tab of the output window during a Build, Build All or Build File operation via the **Options** dialog box.

To view or hide extra information during a build

1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Build** tab.
3. Set the three check boxes in the **Show** group as follows:
   - **Command line** controls whether the command line is shown as each tool is executed.
   - **Environment** controls whether the environment is shown as each tool is executed.
   - **Initial directory** controls whether the current directory is shown as each tool is executed.



Show:

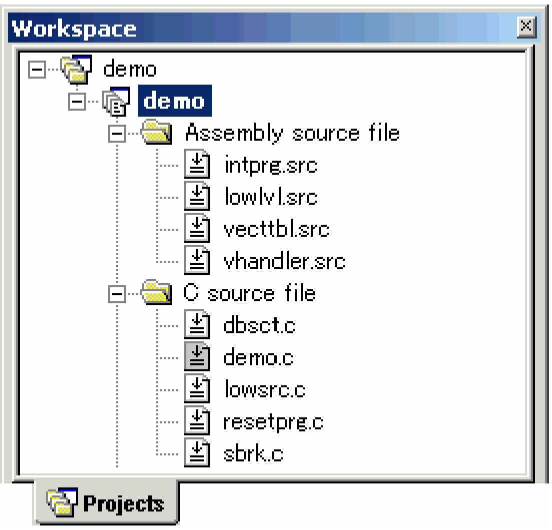
☑ Command line
☑ Initial directory
☑ Environment
□ Mark out of date files

2.8.8 Displaying out of date files in the Workspace window

Files updated later than the file generated by the previous build (i.e. out of date files) are marked in the workspace window. In the window below the file "demo.c" is out of date.

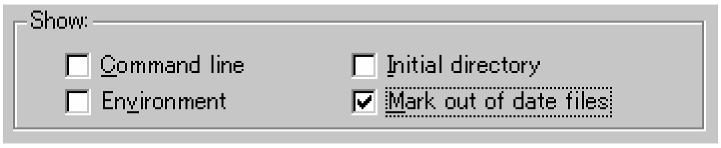
When you click **Build** next time these files will be re-built. This is also displayed for dependent projects of the current project.

The view of these files is updated whenever something that affects the build occurs, e.g. options changing, file addition, dependencies changing, files modified, etc.



To display out of date files in the workspace window

1. Click on the [**Setup->Options**] menu item.
2. Select the **Build** tab.
3. Check the **Mark out of date files** check box.
4. Click OK.



Show:

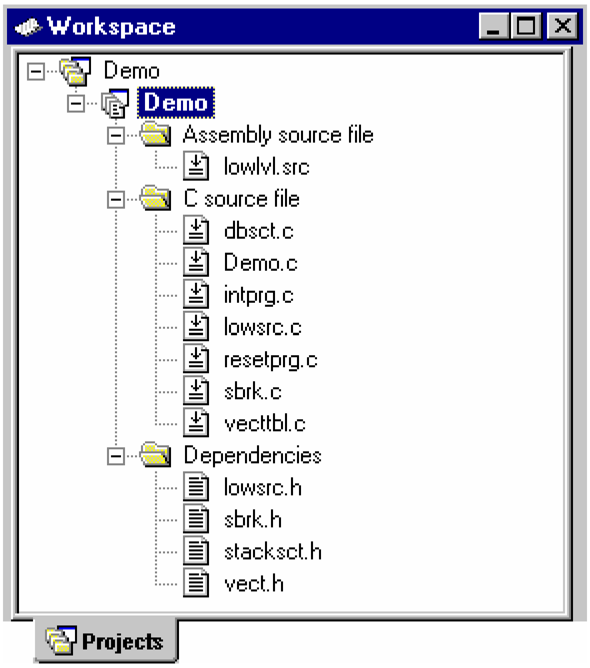
☐ Command line ☐ Initial directory

☐ Environment ☑ Mark out of date files

2.9 File dependencies

A typical project will contain dependencies between files. For example, a C file may '#include' one or more header files. In complex projects, source files will include (or depend upon) others and this can quickly become difficult to manage. However, the HEW provides a dependency scanning mechanism whereby all files in a project are checked for dependencies. Once complete, the **Projects** tab of the Workspace window will display an up-to-date list with all the project file dependencies.

Initially, the dependencies for all files are contained within the **Dependencies** folder (although this can be modified by configuring the projects tab).



File dependencies in a project are automatically updated at the following operation timings:

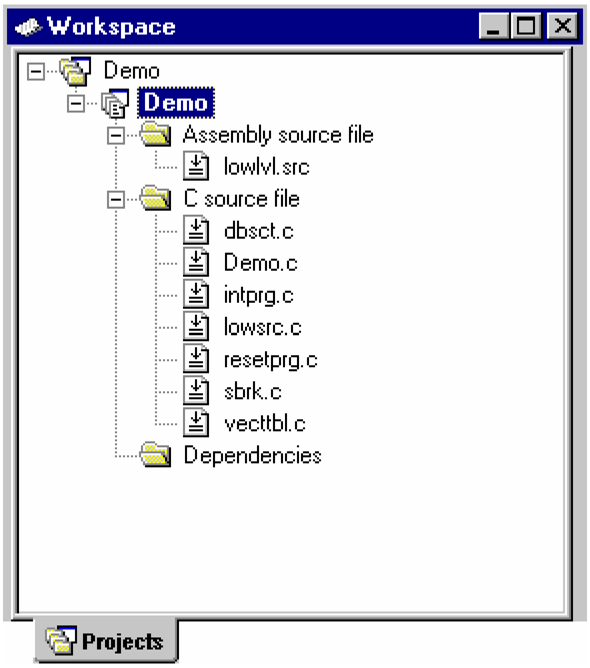
Operation Timing	Specific Operation
Open workspace	Select option in the **Welcome!** dialog box
	Select [**File->New Workspace**]
	Select [**File->Open Workspace**]
Add a project to workspace	Select [**Project->Insert Project**]
Set the current project	Select [**Project->Set Current Project**]
Add files to a project	Select [**File->Add Files**]
Save files	Select [**File->Save**]
Build	Select [**Build->Build**]
Automatic dependency scanning	Deselect **Disable automatic dependency scanning**

To prevent automatic scanning of file dependencies in a project

1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Build** tab.
3. Select the **Disable automatic dependency scanning** checkbox. This checkbox is not selected by default.
4. Click OK.

Note:

Selecting this option clears all file dependencies in a project. To manually update file dependencies in a project, see "To manually update file dependencies in a project".

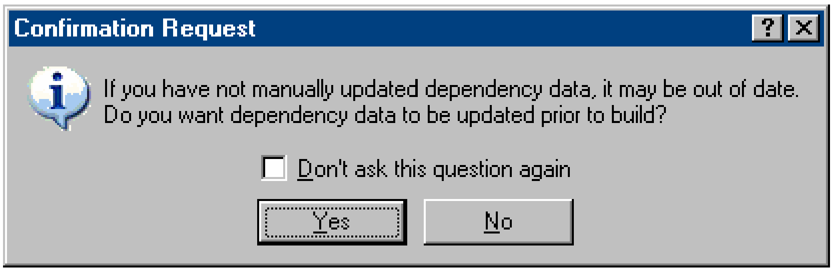


When [**Build->Build**] is selected, a confirmation dialog box appears to ask if you wish to update the file dependencies before executing the build.

If you have not manually updated dependency data, it may be out of date.

When you want dependency data to be updated prior to build, select **Yes**.

If you select **No**, it executes "Build" based on current dependency scanning data.



If you have not manually updated dependency data, it may be out of date.
Do you want dependency data to be updated prior to build?

□ Don't ask this question again

If you do not wish to open this confirmation dialog box, select the **Don't ask this question again** checkbox.

To open the confirmation dialog box again

1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Confirmation** tab.
3. Select the **Scan Dependencies Prior To Build** checkbox. This checkbox is selected by default.
4. Click OK.

To manually update file dependencies in a project

Select one of the following operations:

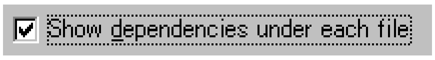
- Select the [**Build->Update All Dependencies**] menu option, **OR**

- Right-click on a project in the Projects tab of the Workspace window and select the [**Build->Update All Dependencies**] option from the pop-up menu.

2.10 Configuring the Workspace window

If you right-click anywhere inside the Projects tab of the Workspace window, a pop-up menu will be invoked. Select the **Configure View** menu option to modify the way in which information is displayed. The following four sections detail the effect of each option on the **Configure View** dialog box.

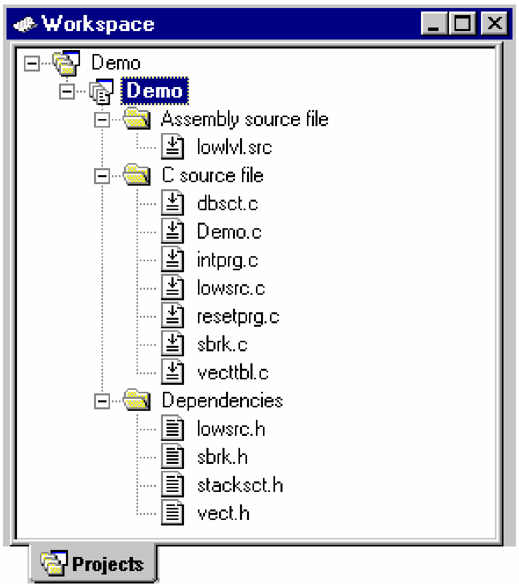
Show Dependencies Under Each File



☑ Show dependencies under each file

If you check the **Show dependencies under each file** checkbox, the dependent files are shown under the including source file as a flat structure, i.e. the files themselves become folders (as in figure (ii) below). If this option is not selected then a separate folder contains all dependencies (as in figure (i) below).

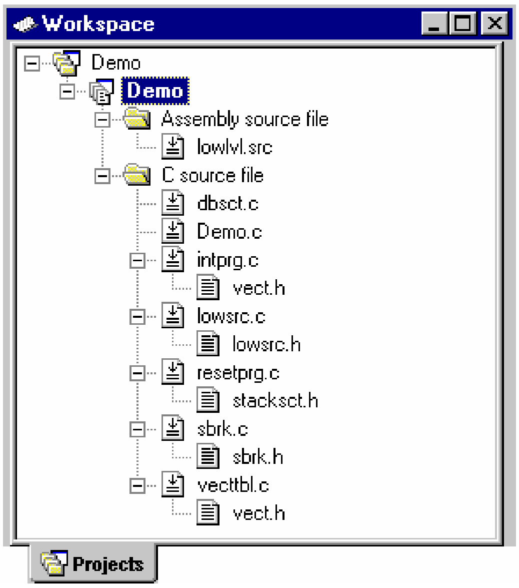
(i)



Workspace

```
Demo
  Demo
    Assembly source file
      lowlvl.src
    C source file
      dbsct.c
      Demo.c
      intprg.c
      lowsrc.c
      resetprg.c
      sbrk.c
      vecttbl.c
    Dependencies
      lowsrc.h
      sbrk.h
      stacksct.h
      vect.h
```

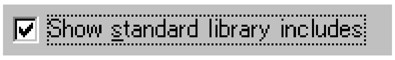
(ii)



Workspace

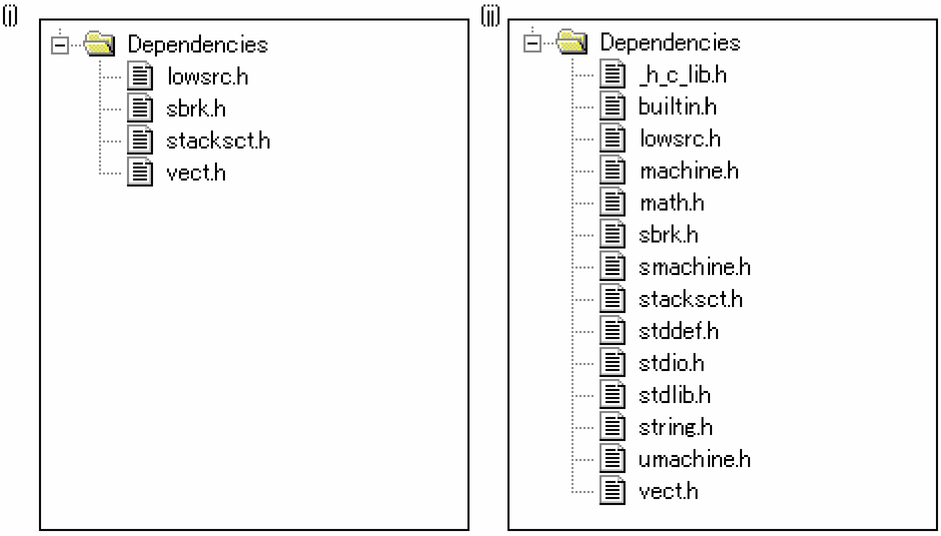
```
Demo
  Demo
    Assembly source file
      lowlvl.src
    C source file
      dbsct.c
      Demo.c
      intprg.c
        vect.h
      lowsrc.c
        lowsrc.h
      resetprg.c
        stacksct.h
      sbrk.c
        sbrk.h
      vecttbl.c
        vect.h
```

Show Standard Library Includes



☑ Show standard library includes

By default, any dependent files found in standard include paths will not be shown (as in figure (j) below). For example, in C code, if you write a '#include' statement, such as '#include <stdio.h>', stdio.h will not be listed as a dependent file. To view such system include files, select the **Show standard library includes** checkbox (as in figure (jj) below).



(i)

- Dependencies
  - lowsrc.h
  - sbrk.h
  - stacksct.h
  - vect.h

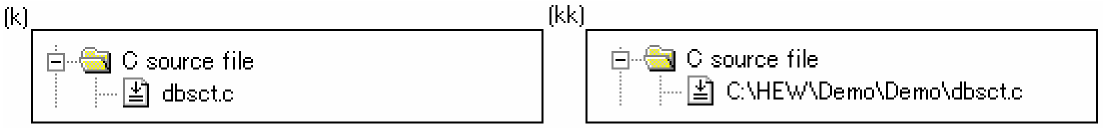
(ii)

- Dependencies
  - _h_c_lib.h
  - builtin.h
  - lowsrc.h
  - machine.h
  - math.h
  - sbrk.h
  - smachine.h
  - stacksct.h
  - stddef.h
  - stdio.h
  - stdlib.h
  - string.h
  - umachine.h
  - vect.h

Show File Paths

☑ Show file paths

By default, any files will not be shown with their full path (as in figure (k) below). If the **Show file paths** checkbox is selected, all of the files in the Projects tab of the Workspace window are shown with their full path, i.e. from a drive letter (as in figure (kk) below).



(k)

- C source file
  - dbsct.c

(kk)

- C source file
  - C:\HEW\Demo\Demo\dbsct.c

Sorting the workspace window into time stamp order

☑ Sort files in time order

When **Sort files in time order** option is selected, the workspace window is then sorted into file time stamp order. The newest files are located at the top of list with the older files towards the bottom.

If files are updated after selecting this option, manually update the order of these files.

To manually update the file order

Select **Refresh Order** in the pop-up menu on the Projects tab of the Workspace window.

2.11 Inserting a project into the workspace

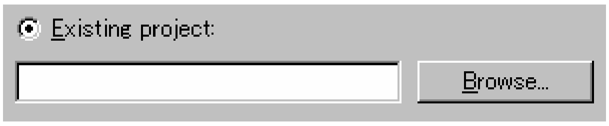
When a workspace is created, it contains only one project but, after it is created, you can insert new or existing projects into the workspace.

To insert a new project into the workspace

● New project

1. Select the [**Project->Insert Project**] menu option. The **Insert Project** dialog box will be displayed.
2. Select the **New project** radio button.
3. Click the OK button. The **Insert New Project** dialog box will be invoked.
4. Enter the name of the new workspace into the **Project Name** field. This can be up to 32 characters in length and contain letters, numbers and the underscore character. Especially, do not use a minus sign, or a space. As you enter the project name the HEW will add a sub-directory for you automatically. This can be deleted if desired.
5. Use the **Browse** button to graphically select the directory in which you would like to create the project. Alternatively, you can type the directory into the **Directory** field manually.
6. The project type list displays all of the available project types (e.g. Application, Library etc.). Select the type of project that you want to create from this list.
7. Click the OK button to create the project and insert it into the workspace.

To insert an existing project into a workspace



● Existing project:

1. Select the [**Project->Insert Project**] menu option. The **Insert Project** dialog box will be displayed.
2. Select the **Existing project** radio button.
3. Enter the full path of the project database file (.HWP file) into the edit field, or click the **Browse** button to search for it graphically.
4. Click the OK button to insert the existing project into the workspace.

2.12 Setting the current project

A project can be in three states – the **Current** project, a **Loaded** project or an **Unloaded** project.

Since a workspace can contain many projects, only one of them can be the **Current** project at any time. This project is the one that build actions and debug operations can be performed on (e.g. clicking the Build toolbar button will build the **Current** project).

To set a project as the current project

Select one of the following operations:

- Select the project that you want to make active from the [**Project->Set Current Project**] sub-menu, **OR**
- Select the project from the Projects tab of the workspace window. Right-click to display the pop-up menu and select the **Set as Current Project** option.

If the project is **Loaded**, it is possible to open the project's directory and view the files. It is also possible to change the builder or debugger options for the project. A **Loaded** project can also have tool executions performed on it from the **Tools** menu.

To unload a project from the workspace

1. Select the **Loaded** project from the Projects tab of the workspace window.
2. Right-click to display the pop-up menu and select the **Unload Project** option. It is possible to select multiple projects in the workspace window to perform this operation.

If the project is **Unloaded**, its icon appears 'grayed' in the Projects tab of the workspace window and no actions can be performed upon it.

Unloaded projects can be loaded by the following operations.

To load all projects in the workspace

1. Select a workspace in the Projects tab of the workspace window.
2. Right-click to display the pop-up menu and select the **Load All Projects** option.

To load a project in the workspace

1. Select the **Unloaded** project from the Projects tab of the workspace window.
2. Right-click to display the pop-up menu and select the **Load Project** option. It is possible to select multiple projects in the workspace window to perform this operation.

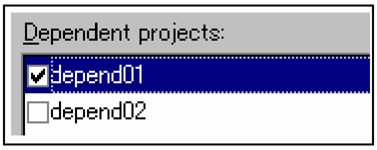
2.13 Specifying dependencies between projects

The projects within a workspace can be dependent upon one another so that when one project is built, all its dependent projects are built first. This is useful if one project uses another in the workspace. For example, imagine that a workspace contains two projects. The first project is a library, which is included by an application project. In this case the library must have been built and up-to-date before the second application can build correctly. To achieve this situation we can specify the library as a dependent (i.e. child) project of the application project. This would then allow the library to be built first if it is out-of-date.

When a dependent project is built, the HEW attempts to match the configuration in the dependent project with that of the current project. This means that if the current configuration is 'Debug' then the HEW will attempt to build the 'Debug' configuration in the dependent project. If this matched configuration does not exist then the HEW will use the configuration that was last used in the dependent project.

To make projects depend upon another

1. Select the [**Project->Dependent Projects**] menu option. The **Dependent Projects** dialog box will be displayed.
2. Select the project to which you would like to add dependents. When you do this, the **Dependent projects** list will display all of the projects in the workspace (excluding the selected project).
3. The **Dependent projects** list has a checkbox for each project listed. Set the associated checkboxes to make those projects depend upon the selected project.
4. Click the OK button.



Dependent projects:

☑ depend01

□ depend02

2.14 Removing a project from the workspace

To remove a project from a workspace

1. Select the project that you wish to remove in the Projects tab of the workspace window.
2. Right-click on the selected project to invoke a pop-up menu.
3. Select the **Remove Project** menu option.

Note:

You cannot remove the Current project from the workspace.

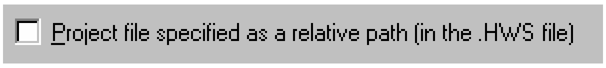
2.15 Relative projects paths in the workspace

In the High-performance Embedded Workshop when you add a project you can choose to add the project to the workspace using a relative path. This allows you to position a file above the workspace directory and it will still be relocated correctly if you relocate the HEW workspace. The project is always relative to the workspace so if the project is one directory above the workspace before it is moved the HEW will try to find the project in the same relative location after the relocation procedure. This is especially useful if you are using a project shared between more than one workspace.

In older versions of HEW this project would not have been relocated and would have still tried to access the original file path. The older version of HEW could only relocate the projects, which were in a sub-directory of the workspace directory. This is still the standard behavior for the High-performance Embedded Workshop.

To add the project to the workspace using a relative path

1. Select the project in the workspace window.
2. Right-click on the selected project to invoke a pop-up menu.
3. Select the **Properties** option.
4. Click the **Project relative file path** checkbox to switch the relative file path feature.
5. Click OK.



☐ Project file specified as a relative path (in the .HWS file)

3. Advanced Build Features

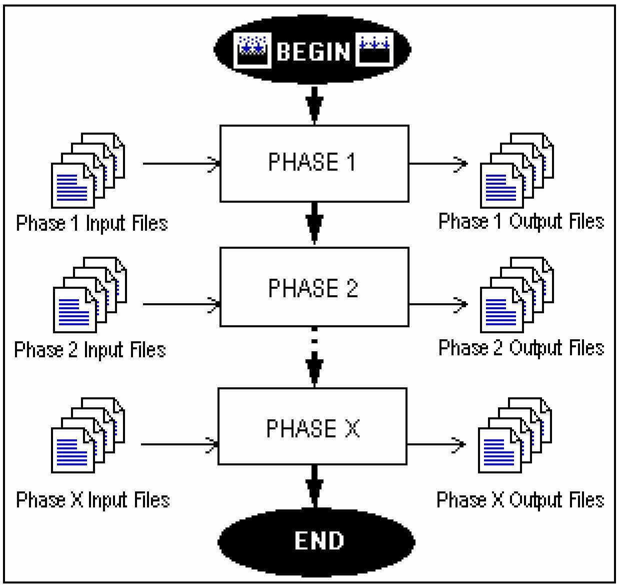
This chapter explains the advanced build concepts.

3.1 The build process revisited

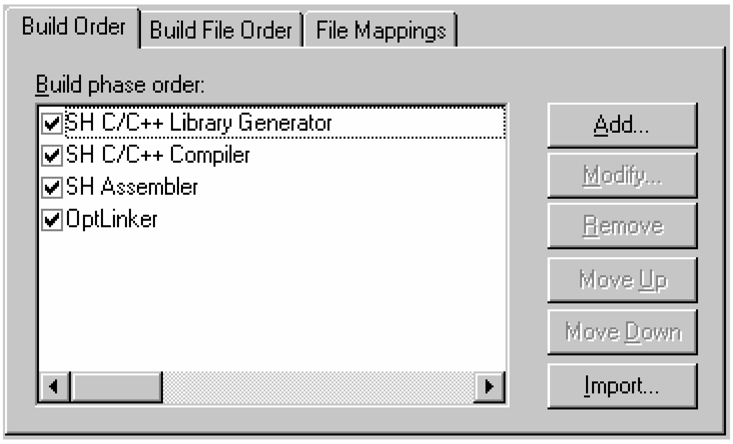
“Chapter 2, Build Basics” began by describing the build process in terms of a compiler, an assembler and a linker. This will be the case for most installations of the High-performance Embedded Workshop. However, if you want to begin changing the build process (e.g. adding and removing phases) then it is important to understand more about the way in which a build functions.

3.1.1 What is a build?

Building a project means applying a set of tools upon certain input files in order to produce the desired output. Thus, we apply a compiler upon C/C++ source files in order to create object files, we apply an assembler upon assembler source files in order to create object files and so forth. At each step or ‘phase’ of the build, we apply a different tool upon a different set of input files. The figure below presents a different view of the build process.



The High-performance Embedded Workshop provides the ability to change this build process via its **Build Phases** dialog box, which can be accessed by selecting the [**Build->Build Phases**] menu option. On the left-hand side are the phases that are defined in the current project.



Build phase order:

- ☑ SH C/C++ Library Generator
- ☑ SH C/C++ Compiler
- ☑ SH Assembler
- ☑ OptLinker

The remainder of this chapter details the various functions that the **Build Phases** dialog box provides.

3.2 Creating a custom build phase

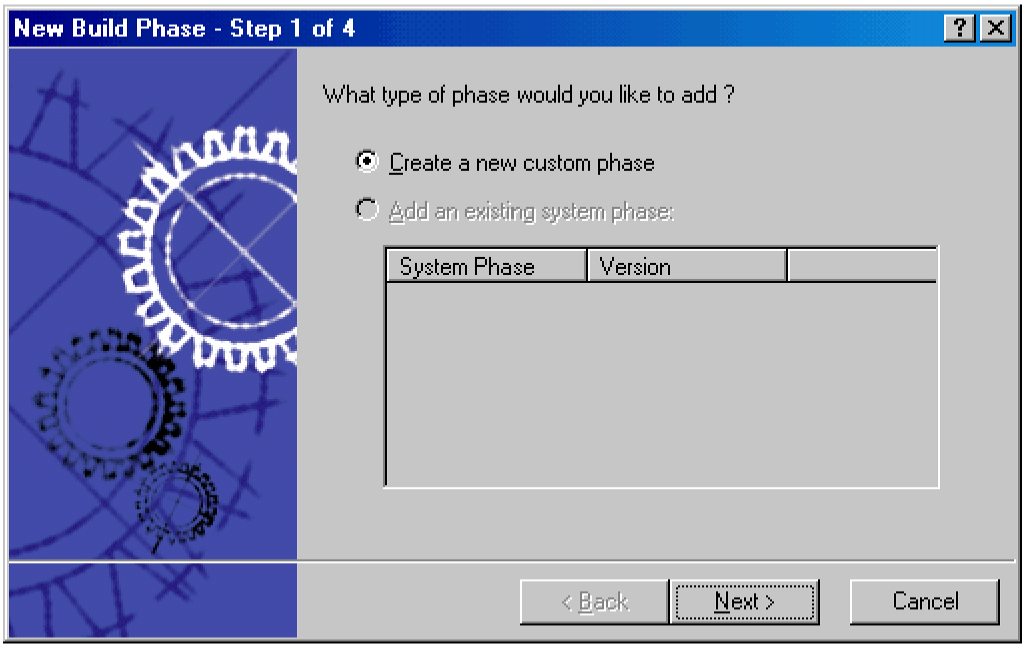
If you want to execute another tool before, during or after a standard build process then this can be achieved by creating your own (i.e. custom) build phase.

To create a new custom build phase

1. Select the [**Build->Build Phases**] menu option to invoke the **Build Phases** dialog box.
2. Click the **Add** button. This will invoke the **New Build Phase** wizard dialog box.
3. Follow the 4 steps below. To move forward and backward between steps click the **Next >** and **< Previous** buttons respectively.

Step 1

The first step asks whether you want to create an entirely new phase or whether you want to add a system phase. A system phase is a ‘ready-made’ phase which is already defined within the toolchain you are using (e.g. compiler, assembler, linker, librarian, etc.) or a utility phase (e.g. file copy, complexity analyzer etc.). The **Add an existing system phase** button is inactive if no more system phases are available. Select the **Create a new custom phase** button to create your own build phase.


New Build Phase - Step 1 of 4

What type of phase would you like to add ?

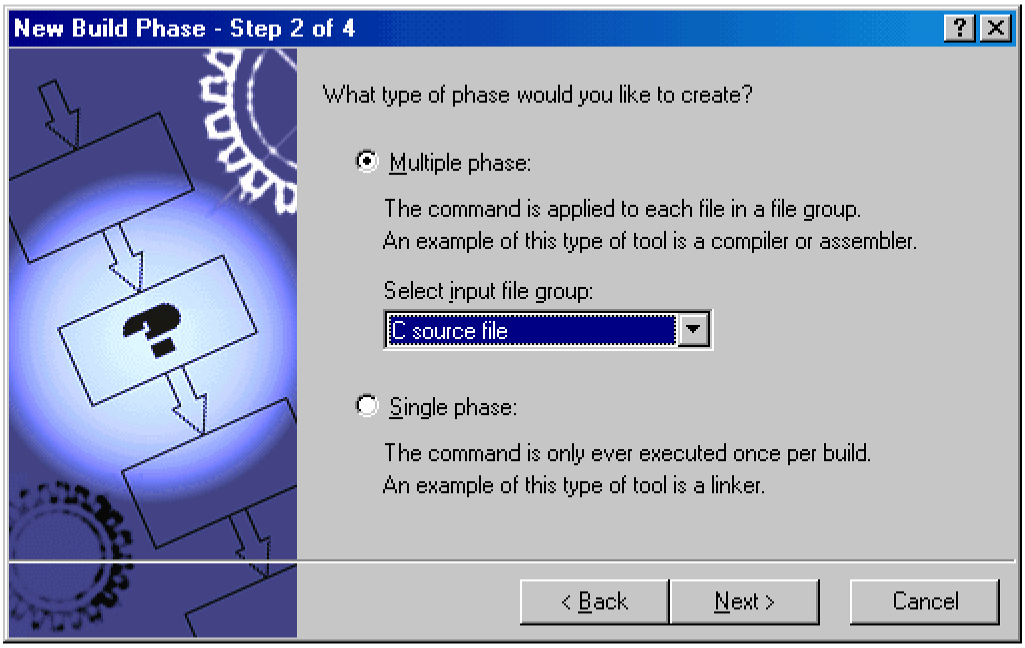
● Create a new custom phase

○ Add an existing system phase:

System Phase	Version	

Step 2

The second step asks what type of phase you would like to create. There are two choices: multiple or single. When a multiple phase is executed, the command is applied to each file in the project of a certain file group. For example, if you set the input file group to be C source files then the command will be executed once for each C source file in the project. A single phase is executed once at most during a build.


New Build Phase - Step 2 of 4

What type of phase would you like to create?

● Multiple phase:

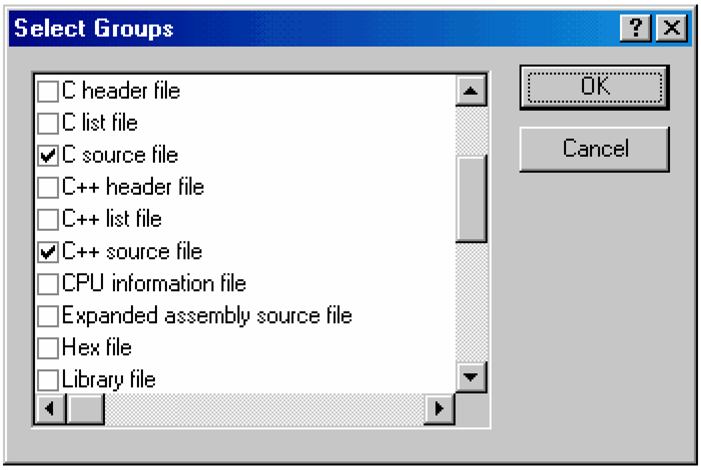
The command is applied to each file in a file group.
An example of this type of tool is a compiler or assembler.

Select input file group: C source file

○ Single phase:

The command is only ever executed once per build.
An example of this type of tool is a linker.

The input file group list contains the current file groups defined for the project. It is possible to define multiple input file groups by selecting the **Multiple Groups** entry in the input file group list.



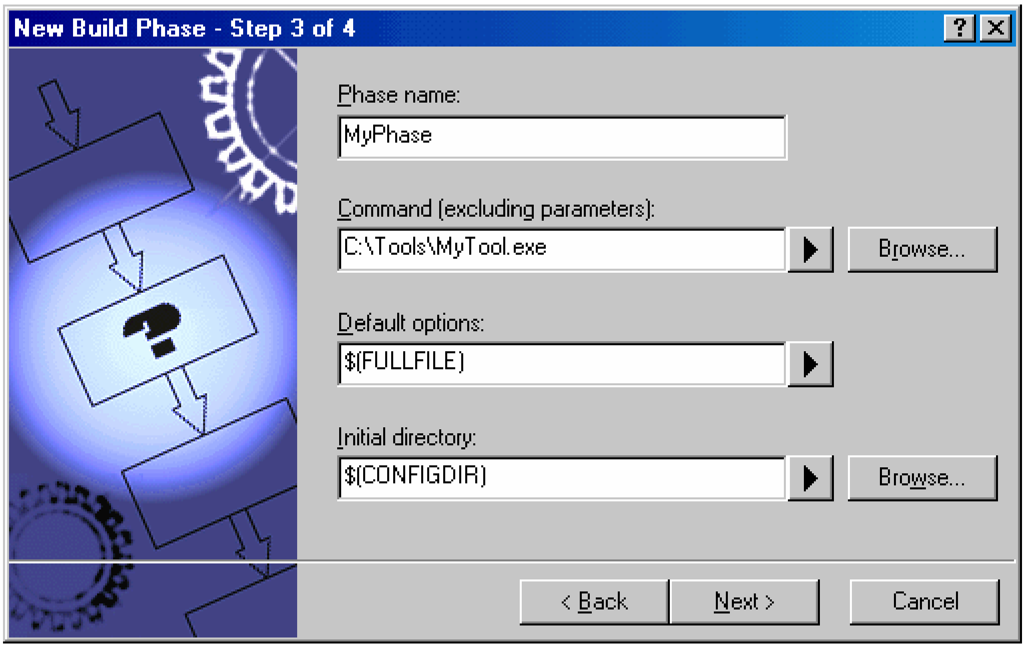
Select Groups

☐ C header file
☐ C list file
☑ C source file
☐ C++ header file
☐ C++ list file
☑ C++ source file
☐ CPU information file
☐ Expanded assembly source file
☐ Hex file
☐ Library file

Once this choice has been made the input file group selection is displayed as **Multiple Groups**. This dialog box allows the user to choose multiple input file groups for the custom phase being added to the project. To select a file group check the box next to the file group's name. One or more file groups can be selected in this dialog box.

Step 3

The third step requests the fundamental information about the new build phase. Enter the name of the phase into the **Phase name** field. Enter the location of the program file into the **Command** field (do not insert any command line options as these options are specified via the **Options** menu of the HEW menu bar). Specify the default options for the phase (i.e. what options you would like new files to take when added to the project) into the **Default Options** field. If you have a preferred directory in which you would like this program to run (i.e. where you want the current working directory to be set to before the tool is executed) then enter it into the **Initial directory** field.



New Build Phase - Step 3 of 4

Phase name: MyPhase

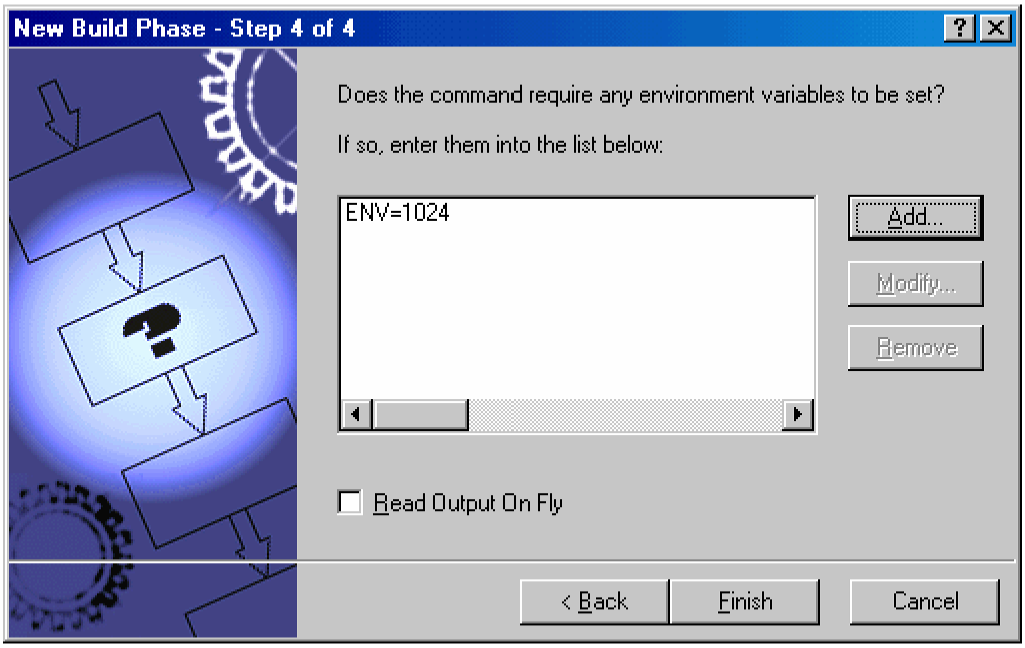
Command (excluding parameters): C:\Tools\MyTool.exe

Default options: $(FULLFILE)

Initial directory: $(CONFIGDIR)

Step 4

The fourth and final step allows you to specify any environment variables that the phase requires.



New Build Phase - Step 4 of 4

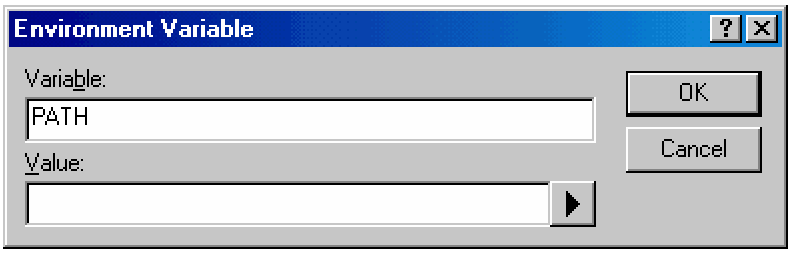
Does the command require any environment variables to be set?

If so, enter them into the list below:

ENV=1024

□ Read Output On Fly

To add a new environment variable click the **Add** button (the **Environment Variable** dialog is displayed). Enter the variable name into the **Variable** field and the variable's value into the **Value** field and then click the OK button to add the new variable to the list. To modify an environment variable select the variable from the list and then click the **Modify** button. Make the required changes to the **Variable** and **Value** fields and then click the OK button to add the modified variable to the list. To remove environment variables select the variable that you want to remove from the list and then click the **Remove** button.



Environment Variable

Variable: PATH

Value:

If the tool you are adding can display its output whilst the tool is running then use the **Read Output on Fly** option. This will display the tool output as each line of output happens. If this option is set to off then the HEW will store all output that is being displayed by the tool, and display it in the **Output** window when the tool has finished its operation. This can be a problem when the tool is running an operation that might take many minutes, as it is difficult to see the progress of the current execution.

Note:

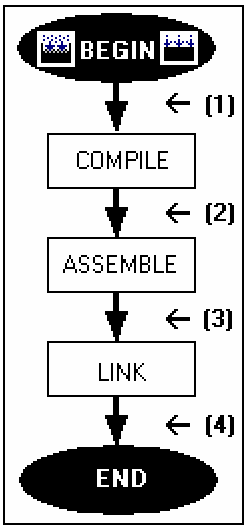
Using **Read Output on Fly** can cause problems when using certain tools on certain operating systems. If you are having problems with tools locking up or freezing in HEW then un-check the **Read Output on Fly** option.

Click the **Finish** button to create the new phase. By default the new phase is added to the bottom of the **Build phase order** list in the **Build Order** tab of the **Build Phases** dialog box.

3.3 Ordering build phases

In a standard build (shown in figure below), you could add a phase at four different positions: before the compiler (1), before the assembler (2), before the linker (3) or after the linker (4).

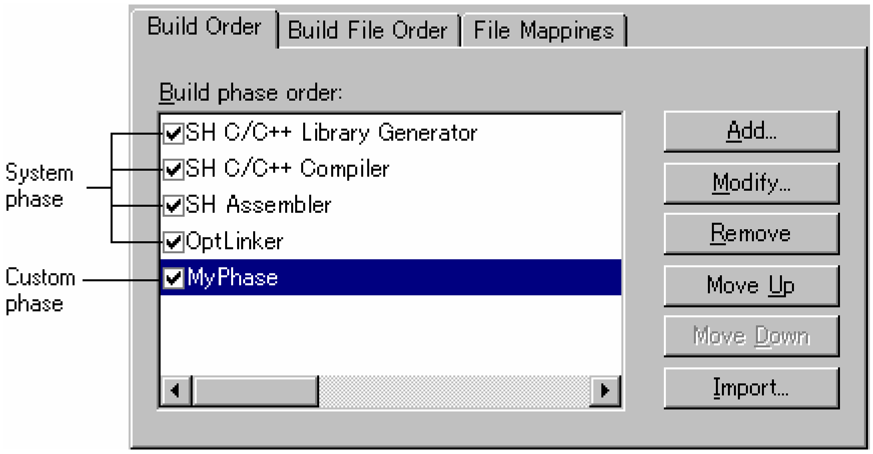
You may place your own custom phases or move system phases to any position in the build order. It is important to remember that if the output of your custom phase can be input into another phase then the phase order must be correct if the build is to behave as intended.



Select the [**Build->Build Phases**] menu option. The **Build Phases** dialog box will be displayed. The build phase dialog box provides facilities for ordering build phases via the **Build Phases** dialog box. It has two tabs, which are concerned with the ordering of phases: **Build Order** and **Build File Order**. And then you can click OK button.

3.3.1 Build Order tab

The **Build Order** tab displays the current order in which phases will be executed when the **Build** (▦) or **Build All** (▦) buttons is selected. The check boxes to the left of each phase indicates whether or not the phase is currently enabled. A phase can be toggled on/off by checking/unchecking its corresponding checkbox respectively.



Build phase order:

☑SH C/C++ Library Generator
☑SH C/C++ Compiler
☑SH Assembler
☑OptLinker
☑MyPhase

To change the order of phases (system/custom) in a build or build all operation

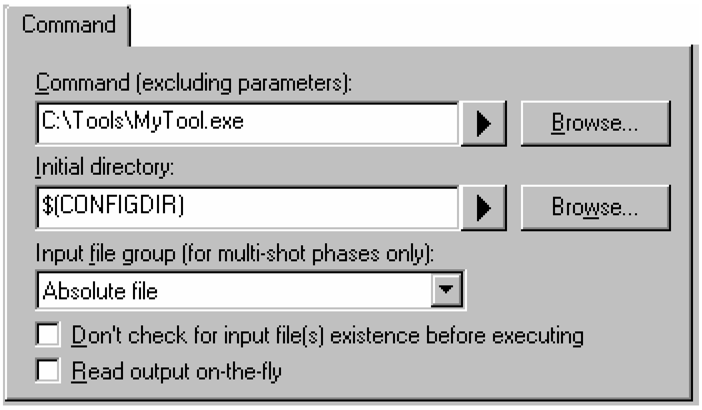
1. Select the phase to be moved and then click the **Move Up** and **Move Down** buttons to move the phase up and down respectively.
2. Click the OK button to set the new ordering.

To view the properties of a system phase

1. Select the system phase that you would like to view.
2. Click the **Modify** button. The **Modify Phase** dialog box will be invoked.
3. The **Command** tab shows general information about the phase. This may include copyright information, enhancements, bug fixes, user notes and so on.
4. Select the **Environment** tab to view the environment settings of the phase.
5. Click the OK button to close the dialog box.

To modify a custom phase

1. Select the custom phase that you would like to modify.
2. Click the **Modify** button. The Modify Phase dialog box will be invoked with the **Command** tab selected.



Command (excluding parameters): C:\Tools\MyTool.exe

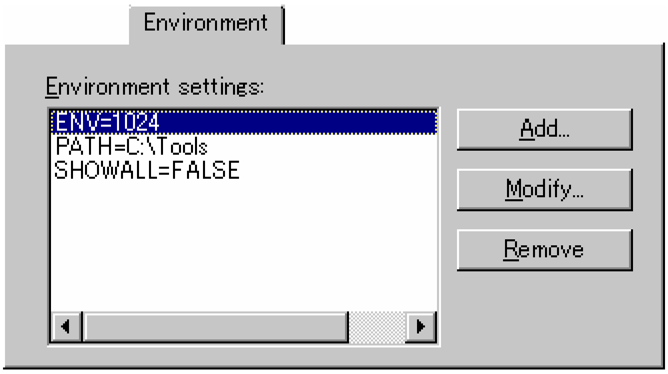
Initial directory: $(CONFIGDIR)

Input file group (for multi-shot phases only): Absolute file

☐ Don't check for input file(s) existence before executing

☐ Read output on-the-fly

3. Change the contents of **Command** and **Initial directory** as appropriate.
4. If there are multiple shot phases, select the file type to execute a phase from the **Input file group (for multi-shot phases only)** drop-down list.
5. Select the **Don’t check for input file(s) existence before executing** checkbox if you don’t want the HEW to abort the execution of the phase if any of the input files do not exist.
6. Select the **Read output on-the-fly** checkbox if you want to display build output as it happens, rather than showing the output at the end of each phase execution.
7. Select the **Environment** tab to edit the environment settings for the phase.

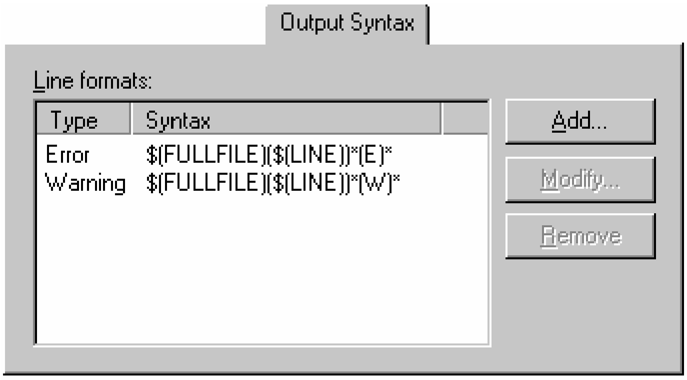


Environment settings:

```
ENV=1024
PATH=C:\Tools
SHOWALL=FALSE
```

8. Use the **Add**, **Modify** and **Remove** buttons to add, modify and remove environment variables respectively.

9. Select the **Output Syntax** tab and define the error, warning, and information for the custom phase.



Type	Syntax
Error	$(FULLFILE)($(LINE))*(E)*
Warning	$(FULLFILE)($(LINE))*(W)*

When the lines showing error messages in the Build tab of the output window match the setting of **Output Syntax**, you can find icons on the left to the error messages and view the source codes in the editor.

If the Renesas SH C/C++ compiler is used, for example, the following error message may be output to the Build tab of the output window.

C:\HEW\Workspace\SH\SH\resetprg.c(70) : C2225 (E) Undeclared name "a"

To define this error, type the following syntax.

$(FULLFILE)($(LINE))*(E)*

The correspondence between an error message and its syntax is as shown below.

Error Message (Example)	Output Syntax (Example)	Description
C:\HEW\Workspace\SH\SH\resetprg.c	$(FULLFILE)	Placeholder of the file name with its full path
(70)	($(LINE))	Placeholder of the line number enclosed with parentheses
: C2225	*	Wild-card for ":" and error message number
(E)	(E)	(E) indicating an error
Undeclared name "a"	*	Wild-card for error message

There are some restrictions on these regular expressions. For example, it is not possible to define two unclear strings next to each other, such as "**", "*$(FILEDIR), or "$(FULLFILENAME)*".

- When "Error" is selected as the syntax type:

Double-clicking on an output message line in the Build tab of the output window shows the file name with its full path and line number. The source code of this line is shown in the editor. The Build tab of the output window also shows an error icon (❗).

- When "Warning" is selected as the syntax type:

Double-clicking on an output message line in the Build tab of the output window shows the file name with its full path and line number. The source code of this line is shown in the editor. The Build tab of the output window also shows a warning icon (⚠).

- When "Other" is selected as the syntax type:

The Build tab of the output window shows an information icon (ⓘ). The total number of errors and warnings will also be displayed in the Build tab of the output window at the end of a build phase.

10. Use the **Add**, **Modify** and **Remove** buttons to add, modify and remove output syntax line formats respectively.

11. Click the OK button when all modifications have been made.

Note:

You can only change the environment of a system phase via the **Tools Administration** dialog box.

To remove a custom phase

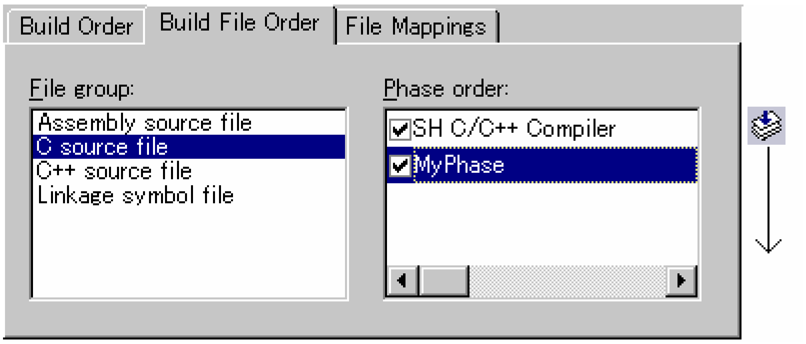
1. Select the phase to be removed and click the **Remove** button.
2. Click the OK button to confirm the new settings.

To import a custom phase

1. Click the **Import** button. The **Import Custom Phase** dialog box is displayed, which allows you to browse to an existing project, from which you want to import a custom phase.
2. Choose the location of the project, from which you want to import a custom phase. Once selected, the **Import Phase** dialog box is displayed, which lists the custom phases in the imported project.
3. Once you have decided which phase to import, highlight it in the list and click the OK button. The phase will then be added to the **Build Phases** dialog box, at the bottom of the build order.

3.3.2 Build File Order tab

If you were to select a C source file from the Workspace window and then activate [**Build->Build File**] (or click the **Build File** toolbar button) you would expect the file to be compiled. Likewise, if you were to select an assembly source file from the workspace window and then activate [**Build->Build File**] you would expect the file to be assembled. The connection between file group and which phase(s) to execute is managed by the **Build File Order** tab of the **Build Phases** dialog box. The list displays all of the current phases that will be executed when the build file operation is selected upon the file group shown in the **File group** list box. In figure below the "C source file" file group is selected and the "Compiler" and "MyPhase" phases are associated with it. Entries in the **Phase order** list, of the **Build File Order** tab, are added automatically as new entries are added to the **Build Order** tab.



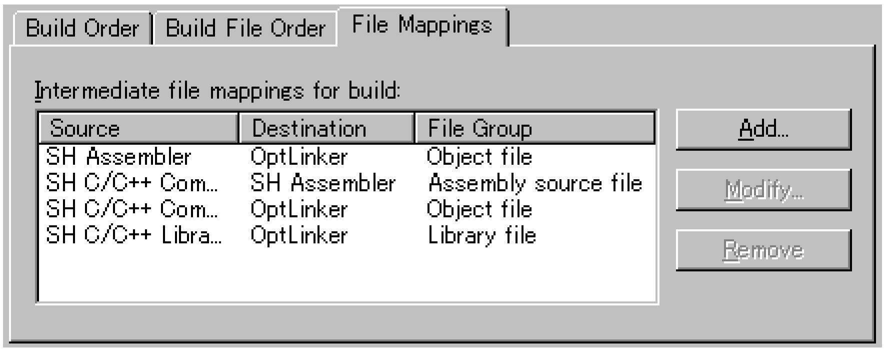
File group:
- Assembly source file
- C source file
- C++ source file
- Linkage symbol file

Phase order:
- ☑SH C/C++ Compiler
- ☑MyPhase

For example, if you were to add a phase which takes C source files as input then this phase will be automatically added to the list of phases to execute when a build file operation is applied to a C source file. If you don't want a certain phase to execute when [**Build->Build File**] is selected then clear the check box to the left of the phase name in the **Phase order** list.

3.3.3 File Mappings tab

By default, the files input to a build phase are only taken from the project, i.e. all project files of the type specified in the **Select input file group** drop-down list on the **New Build Phase** dialog box. Select the [**Build->Build Phases**] menu option. The **Build Phases** dialog box will be displayed. If you would like a build phase to take files output from a previous build phase (these files are called intermediate files), then you must define this in the **File Mappings** tab of the **Build Phases** dialog box.



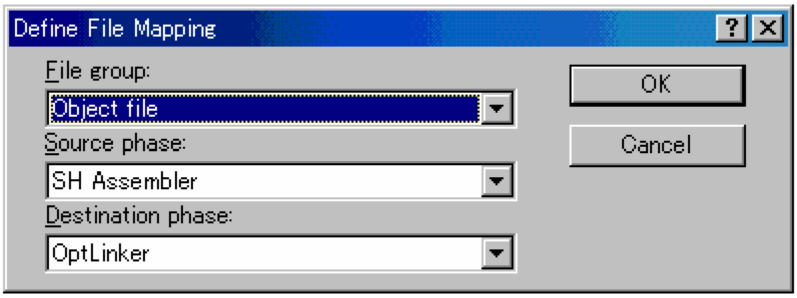
Intermediate file mappings for build:

Source	Destination	File Group
SH Assembler	OptLinker	Object file
SH C/C++ Com...	SH Assembler	Assembly source file
SH C/C++ Com...	OptLinker	Object file
SH C/C++ Libra...	OptLinker	Library file

A **File Mapping** means that you would like the output files of a certain type produced by one build phase (referred to as the **Source** phase) to another build phase (referred to as the **Destination** phase). Such intermediate files are then passed in addition to the project files.

To add a file mapping

1. Click the **Add** button. The **Define File Mapping** dialog box will be invoked.
2. Select an existing file group to use for the new mapping from the **File group** drop-down list.
3. Select the source phase (i.e. the phase that generates the files) from the **Source phase** drop-down list.
4. Select the destination phase (i.e. the phase that takes these files) from the **Destination phase** drop-down list.
5. Click the OK button to create the new mapping.



File group: Object file

Source phase: SH Assembler

Destination phase: OptLinker

To modify a file mapping

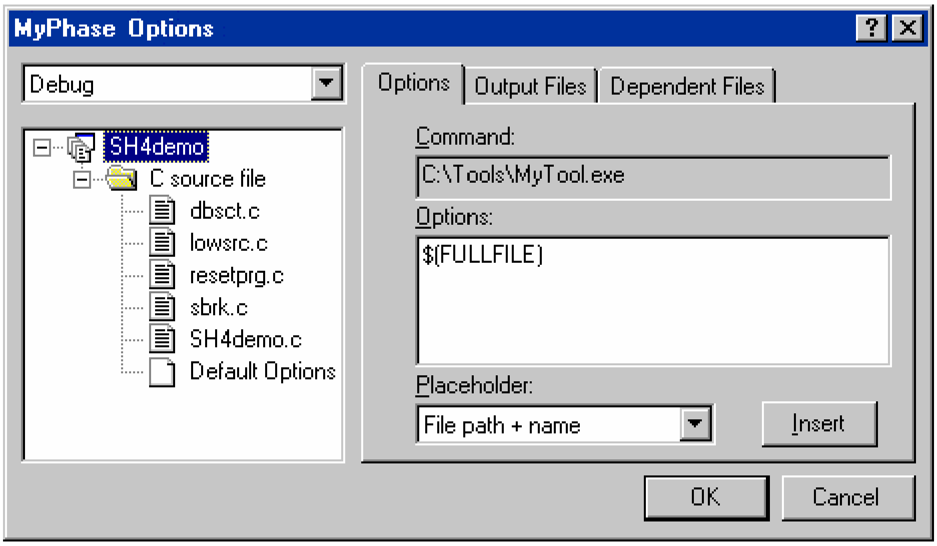
1. Select the mapping to be modified.
2. Click the **Modify** button. The **Define File Mapping** dialog box will be invoked.
3. Modify the options as necessary.
4. Click the OK button to commit the changes.

To remove a file mapping

1. Select the mapping to be removed.
2. Click the **Remove** button.
3. Click the OK button to commit the changes.

3.4 Setting custom build phase options

Once you have defined a custom phase, you will want to specify the command line options that should be used when it is executed. Each defined phase has a menu option in the **Build** menu. To specify options for that phase select it. The dialog that will be displayed depends on whether the custom phase selected was multiple or single (according to the selection of phase type when it was created in the New Build Phase wizard).



Command: C:\Tools\MyTool.exe

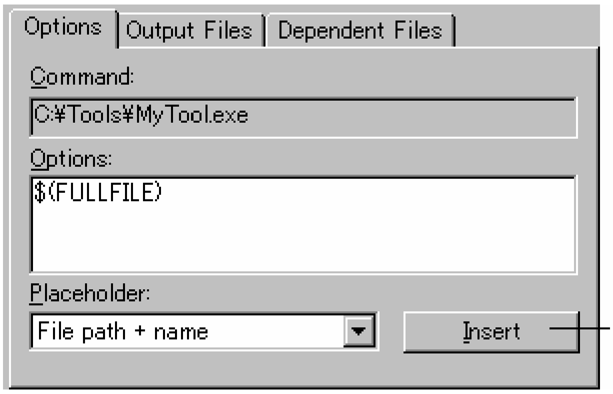
Options: $(FULLFILE)

Placeholder: File path + name

If the phase selected was multiple then a list of project files is displayed on the left-hand side of the dialog to enable you to specify the build options on a file by file basis. If the phase selected was single then there is no project file list displayed. In either case, the three tabs below are available. This is where you can set the options that you want to apply to the selected file(s). You can also choose which configurations are being viewed. In the configuration list, each configuration is listed along with a special entry named **Multiple configurations**. If you select **Multiple configurations** then the **Select project configurations to modify** dialog box is displayed which allows you to select more than one configuration. This method is used throughout HEW for modifying multiple configurations at once.

3.4.1 Options tab

This tab allows you to define the command line options that will be passed to the phase. The **Command** field displays the command that was entered when you defined the phase. Enter into the **Options** field the command line arguments that you would like to pass to the command. If you want to insert a placeholder, select the relevant placeholder from the **Placeholder** drop-down list and then click the **Insert** button. See Reference 5, Placeholders, for more information on placeholders.



Command: C:¥Tools¥MyTool.exe

Options: $(FULLFILE)

Placeholder: File path + name

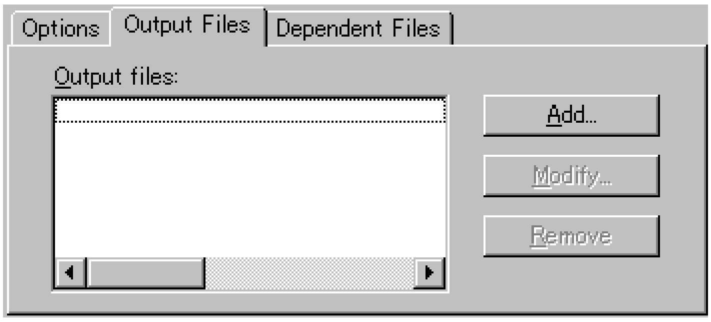
Insert button
for
placeholder

3.4.2 Output Files tab

This tab is where you can specify the output files that will be produced by the phase. Before each file is passed into this phase, the HEW checks that the output files are of a less recent date than the input file. If so, the phase will be executed for that file (i.e. input files have been modified since the output file or files were last produced). If the files are up-to-date then the phase will not be executed.

Note:

If no output files are specified, the phase will execute regardless.



To add an output file

1. Click the **Add** button. The **Add Output File** dialog box will be invoked.
2. Enter the file path or browse to it using the **Browse** button.
3. Click the OK button to add this output file to the list.

To modify an output file

1. Select the output file that you would like to modify.
2. Click the **Modify** button. The **Modify Output File** dialog box will be invoked.
3. Modify the fields as required.
4. Click the OK button to add the modified entry back to the list.

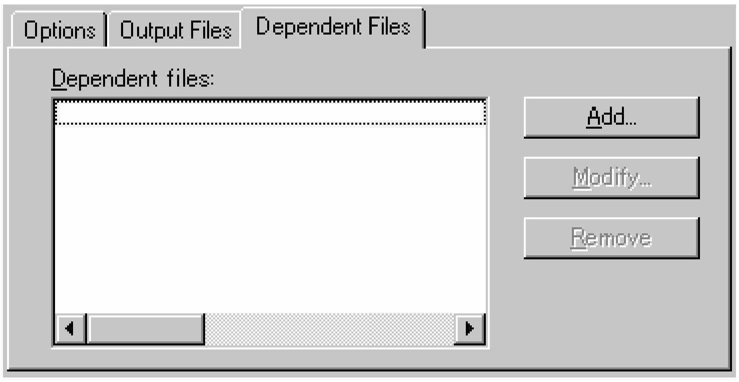
To remove an output file

1. Select the output file that you would like to remove.
2. Click the **Remove** button.

An output file's path can include placeholders.

3.4.3 Dependent Files tab

This tab is where you can specify the dependent files that are needed by the phase. Before each file is passed into this phase, the HEW checks that the dependent files are of a more recent date than the input file. If so, the phase will be executed for that file (i.e. dependent files have been modified since the input files was last modified). If not, the phase will not be executed.



To add a dependent file

1. Click the **Add** button. The **Add Dependent File** dialog box will be invoked.
2. Enter the file path or browse to it using the **Browse** button.
3. Click the OK button to add this output file to the list.

To modify a dependent file

1. Select the dependent file that you would like to modify.
2. Click the **Modify** button. The **Modify Dependent File** dialog box will be invoked.
3. Modify the fields as required.
4. Click the OK button to add the modified entry back to the list.

To remove a dependent file

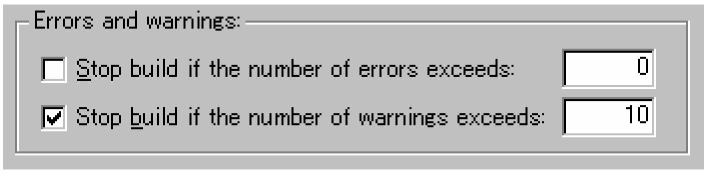
1. Select the dependent file that you would like to remove.
2. Click the **Remove** button.

A dependent file's path can include placeholders.

3.5 Controlling the build

By default, the High-performance Embedded Workshop will execute all of the phases in a build and only stop if a fatal error is encountered. You can change this behavior by setting the controls on the **Build** tab of the **Options** dialog box.

Select the [**Setup->Options**] menu option to display the **Options** dialog box and then select the **Build** tab. If you want to stop the build when a certain number of errors are exceeded then set the **Stop build if the number of errors exceeds** checkbox and specify the error count limit in the field to the right. If you want to stop the build when a certain number of warnings are exceeded then set the **Stop build if the number of warnings exceeds** checkbox and specify the warning count limit in the field to the right.


Errors and warnings:

□ Stop build if the number of errors exceeds: 0

☑ Stop build if the number of warnings exceeds: 10

In addition to specifying error and warning count limits, the **Build** tab also allows you to request that the **Command line**, **Environment** and **Initial directory** of each execution should be displayed. Check the appropriate check boxes as necessary.

Note:

- Irrespective of what these controls are set to, the build will always halt if a fatal error is encountered.
- Note the following descriptions when you check the **Stop build if the number of errors exceeds** check box and specify a number to the controls.

    i. "number of errors" in the statement means the number of errors output from an execution of a tool. The total number of errors in more than one execution of a tool is irrelevant to **Stop build if the number of errors exceeds**. For example, the build stops when one execution of the C/C++ Compiler outputs more than the number specified. If the number of errors output from each compilation is no more than the specified number, the build does not stop even though the total number of errors in more than one compilation exceeds the specified number.

    ii. The build stops after the whole phase is done. For example, suppose you compile more than one file in a build. Even though the number of errors in the first compilation exceeds the specified number, the HEW continues compilation of the other files and stops the build after finishing the whole compiler phase.

    iii. When the number of errors in one execution of a tool exceeds the specified number, the number of error messages displayed on the "Output" window is the specified number plus one. A message saying that the number of error has exceeded the specified number is NOT displayed on the **Output** window.

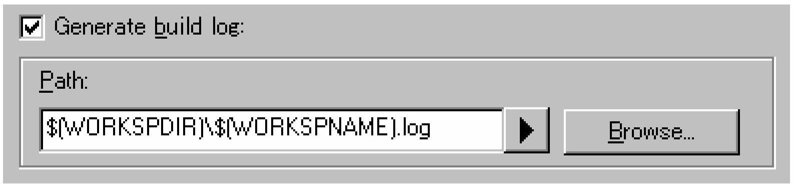
    The same things as **Stop build if the number of errors exceeds** shown above hold also in **Stop build if the number of warnings exceeds**. There is no correlation between **Stop build if the number of errors exceeds** and **Stop build if the number of warnings exceeds**. They are independent.

3.6 Logging build output

The HEW allows you to write the results of each build to file.

To specify a log file

1. Select the [**Setup->Customize**] menu option. The **Customize** dialog box will be displayed.
2. Select the **Log** tab.
3. Select the **Generate build log** checkbox.
4. Enter the full path of the log file into the **Path** field, or browse to it graphically by clicking the **Browse** button, or click the placeholder button (▶) and select the placeholder from the pop-up menu.
5. Click the OK button to confirm the new log file settings.

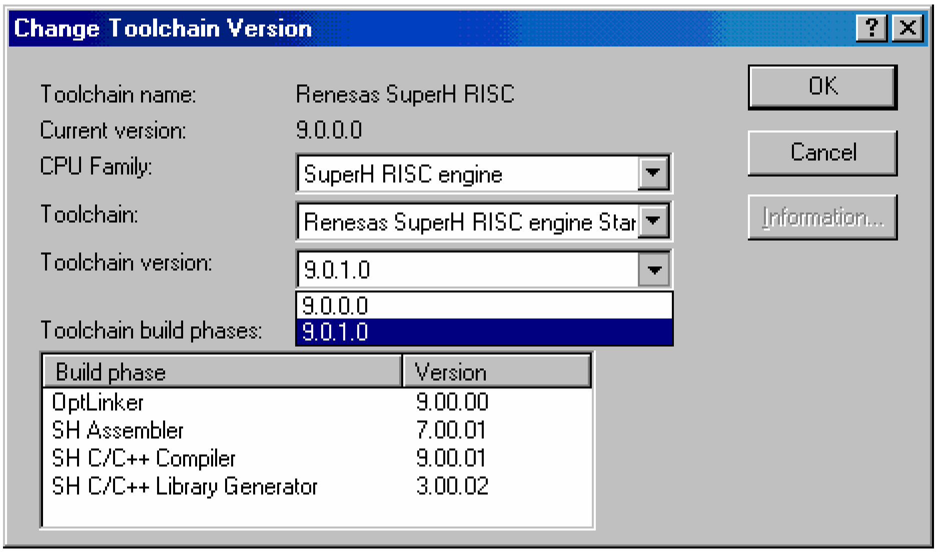


☑ Generate build log:

Path: $(WORKSPDIR)\$(WORKSPNAME).log

3.7 Changing toolchain version

If two or more versions of the same toolchain are registered in the HEW, you can choose a version of the toolchain on the **Change Toolchain Version** dialog box. To invoke the dialog, select the [**Tools->Change Toolchain Version**] menu option. Choose one of the versions from the **Toolchain version** drop-down list and click the OK button to enforce your choice.



Toolchain name: Renesas SuperH RISC

Current version: 9.0.0.0

CPU Family: SuperH RISC engine

Toolchain: Renesas SuperH RISC engine Star

Toolchain version: 9.0.1.0

9.0.0.0
9.0.1.0

Toolchain build phases:

Build phase	Version
OptLinker	9.00.00
SH Assembler	7.00.01
SH C/C++ Compiler	9.00.01
SH C/C++ Library Generator	3.00.02

To show information about toolchain components, select a tool from the **Toolchain build phase** drop-down list on the **Change Toolchain Version** dialog box, and click the **Information** button. A tool information dialog box will show you information about the tool. Click the **Close** button to close the dialog box.

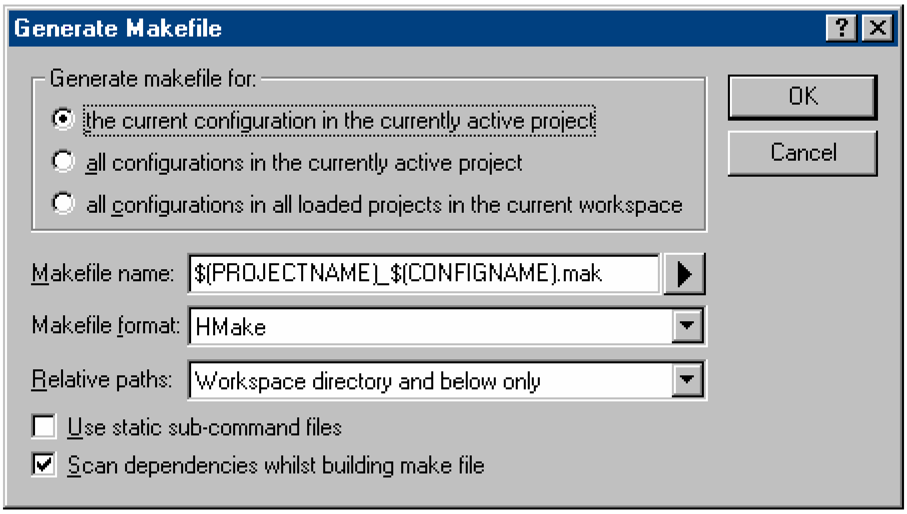
3.8 Generating a makefile

The HEW allows you to generate a makefile associated with your workspace. HEW is capable of generating hmake, nmake, and gmake compatible files. The make tool to execute makefiles is operated in the DOS command prompt and thus the use of makefiles allows you to build projects without HEW.

The generated makefiles can be executed in the HEW. For details, see section 3.9, Using a makefile inside the HEW system.

To generate a makefile

1. Open a workspace where you wish to generate a makefile (if the workspace includes two or more projects, the active project must be one in which you wish to generate a makefile).
2. Select a configuration for generating a makefile.
3. Select the [**Build->Generate Makefile**] menu option.
4. The **Generate Makefile** dialog box is displayed.



Generate Makefile

Generate makefile for:

● the current configuration in the currently active project

○ all configurations in the currently active project

○ all configurations in all loaded projects in the current workspace

Makefile name: $(PROJECTNAME)_$(CONFIGNAME).mak

Makefile format: HMake

Relative paths: Workspace directory and below only

□ Use static sub-command files

☑ Scan dependencies whilst building make file

5. Select a makefile type in **Generate makefile for**.

- the current configuration in the currently active project
- all configurations in the currently active project
- all configurations in all loaded projects in the current workspace

6. Enter the file name in the **MakeFile name** edit box. To insert a placeholder, place the cursor at the position where you wish to insert the placeholder, click the placeholder button (▶), and select Placeholder from the pop-up menu. The default makefile names are as follows.

- When "the current configuration in the currently active project" is selected: $(PROJECTNAME)_$(CONFIGNAME).mak
- When "all configurations in the currently active project" is selected: $(PROJECTNAME).mak
- When "all configurations in all loaded projects in the current workspace" is selected: $(WORKSPNAME).mak

7. Select a makefile format in the **Makefile format** drop-down list. HEW is capable of generating GNUMake, HMake and NMake compatible files.

8. In the **Relative paths** drop-down list, select how directories within the makefile should be expressed. “Workspace directory and below only” is the default option. See the table below for details.

Option	Workspace Directory and below	Outside the Workspace Directory
None	Absolute path	Absolute path
Workspace directory and below only	Relative path	Absolute path
All	Relative path	Relative path

Note:

While a SuperH or H8 family toolchain is in use, this option is fixed to “Workspace directory and below only”.

9. Selecting the **Use static sub-command files** checkbox will generate separate command files in the make destination directory. By default, this checkbox is not selected.

Note:

While a SuperH or H8 family toolchain is in use, the **Use static sub-command files** checkbox has a fixed tick if you have selected GNUMake in **Makefile format**.

10. Selecting the **Scan dependencies whilst building make file** checkbox will execute a dependency scan before creation of a makefile. This checkbox is selected by default.

11. Click OK.

The HEW will create a subdirectory called ’make’ within the current workspace directory and then generate the makefile into it. The executable HMAKE.EXE, located in the HEW installation directory, is provided for you to execute the makefiles generated by HMake selected in **Makefile format**.

If you wish to modify makefiles, see Reference 13, HMAKE User Guide.

Note:

If the name of the HEW installation directory includes a space, the GNU Make command may not work correctly when GNU Make is selected as the makefile format with the makefile generating function.

To execute a makefile (HMake)

1. Open a DOS Command Prompt window and move to the ‘make’ directory where the makefile was generated.
2. Execute HMAKE. Its command line is HMAKE.EXE <makefile>.

Note:

The degree of portability of a generated makefile is entirely dependent upon how portable the project itself is. For example, any compiler options that include full paths to an output directory or include file directory will mean that, when given to another user with a different installation, the build will probably fail. In general use placeholders wherever possible – using a full, specific path should be avoided when possible.

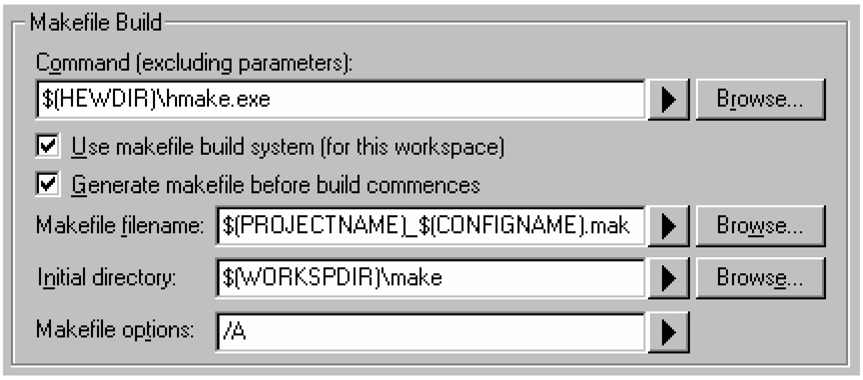
3.9 Using a makefile inside the HEW system

The HEW also allows you to configure the internal build to use a particular make tool (hmake, nmake, or gmake) as well as the internal HEW system. Only the makefiles created in workspaces are executable.

The build procedures using makefiles are listed below. nmake and gmake tools must be prepared by the user, while hmake is installed in the HEW installation directory.

To set-up the internal makefile execution

1. Create a HEW workspace.
2. Select [**Build->Generate Makefile**] to generate a makefile. For details, see section 3.8, Generating a makefile.
3. Select [**Setup->Options**]. The **Options** dialog box is displayed.
4. Click the **Build** tab.



Makefile Build

Command (excluding parameters): $(HEWDIR)\hmake.exe

☑ Use makefile build system (for this workspace)

☑ Generate makefile before build commences

Makefile filename: $(PROJECTNAME)_$(CONFIGNAME).mak

Initial directory: $(WORKSPDIR)\make

Makefile options: /A

5. Enter the make tool name to execute the makefile in the **Command (excluding parameters)** edit box. The default is hmake ($(HEWDIR)\hmake.exe).

6. Select the **Use makefile build system (for this workspace)** checkbox. This means that the makefile should be executed rather than the internal build.
7. While the **Generate makefile before build commences** checkbox is selected, the HEW will generate a makefile before executing the build. This means the makefile will always be up to date with the HEW project system.
8. Enter the file name in the **MakeFile name** edit box. The default file name is $(PROJECTNAME)_$(CONFIGNAME).mak.
9. In the **Initial directory** edit box, enter the current directory where the makefile is to be executed. When this edit box is empty, the directory where the makefile specified in the **MakeFile name** edit box is stored is assumed as the current directory.
10. Enter the options for building the makefile in the **Options** edit box.
11. Click OK.

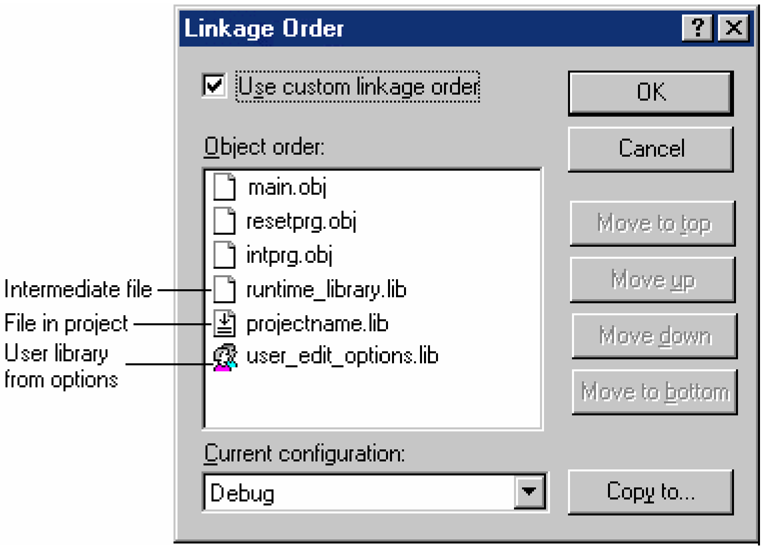
Once this system is setup the build button, menu and keyboard shortcuts are linked to the makefile tool execution. All output is directed to the Build tab of the output window as in the case of the normal build. If you are using a toolchain supported by HEW the line of an error message can be double-clicked to jump to the source files. The help link should also be supported. Note when using the HMAKE.exe system the **Build All** toolbar button () will pass a command to HMAKE to force a re-build all operation.

3.10 Customizing the HEW linkage order

Object modules are linked as alphabetical order in HEW default. You can specify the linkage order, if you wish to.

To switch on manual linkage order facilities

1. Click the [**Build->Linkage Order**] menu item. The **Linkage Order** dialog box is displayed.
2. Select the **Use custom linkage order** check box.
3. Then you can move the objects into which ever order you need. Simply select the module and click the **Move to top**, **Move up**, **Move down** and **Move to bottom** buttons to position it in the desired location.
4. Each module has a different icon depending on where it originated from. This is shown below:



Linkage Order

☑ Use custom linkage order

Object order:

main.obj
resetprg.obj
intprg.obj
runtime_library.lib
projectname.lib
user_edit_options.lib

Current configuration: Debug

5. Click OK.

When you are using multiple configurations it is likely that the linkage order will be very similar. To do this effectively you can copy the current settings in the dialog to other configurations. This is described below:

To copy the linkage order from one configuration to another

1. Click the [**Build->Linkage Order**] menu item. The **Linkage Order** dialog box is displayed.
2. Select the configuration you wish to copy in the **Current configuration** drop-down list. This defaults to the currently loaded configuration.
3. Click the **Copy to** button this displays the **Select Configuration To Copy To** dialog box and asks you which of the configurations in the current project you wish to copy the current linkage order to. Select a configuration and click OK.
4. Click OK.

4. Using the Editor

This chapter describes how to use the editor that is provided with the High-performance Embedded Workshop (HEW).

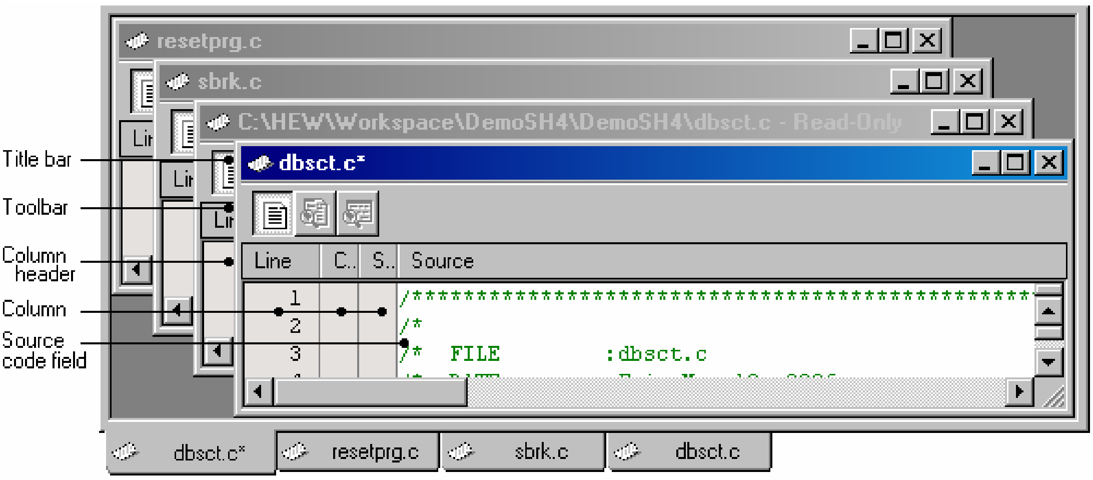
4.1 Editor window

The editor window contains the file windows that are being viewed or edited.

The title bar of the active window will appear a different color from that of the others ("dbsct.c" is the active window in figure bellow). All text operations such as typing, pasting text and so forth only affect the active window.

To switch to another source file window (i.e. to make some other window the active window) there are a number of methods:

- Click on it if it is visible, **OR**
- Press CTRL+TAB or CTRL+SHIFT+TAB to cycle through the windows one after another, **OR**
- Select the window by name from the **Window** menu, **OR**
- Select its tab at the bottom of the editor window.



```
/***************************************************
/*
/*  FILE        :dbsct.c
```

Title bar
Shows information on files (file name, asterisk * while editing, and Read-Only attribute). If a file is included in a directory of the current project, the file name is displayed on the title bar of the editor window. Otherwise the title bar of the editor window displays the file name including its absolute path.

Toolbar
The toolbar is only shown while the debugging platform is connected with the session. When one display mode is active, clicking another toolbar button switches the display mode. If you place the mouse pointer on a toolbar button, the name of the button pops up.

Column header
You can adjust the column width by dragging the mouse on a column header. Moreover, right-clicking the column header displays a pop-up menu. A tick mark right next to an entry indicates that this column is displayed. Clicking an entry will switch showing/hiding the column. If you place the mouse pointer on a column header, the name of the column pops up. You can also select showing/hiding the column header itself.

Column
There are several spaces (columns) on the left of the source code field. If you place the mouse pointer on a column, the name of the column pops up.

Source code field — This field includes codes highlighting the syntax. The maximum number of characters per line shown in the editor is 8,192. Right-clicking within this field displays a pop-up menu containing available options.

4.2 Working with multiple files

The file area is where you will work with the files of your project. The editor allows you to have many files open at one time, to switch between them, to arrange them in different configurations and to edit them in whichever order you want to. The operations that you can perform upon the windows are typical of most Windows® applications and they can be found under the **Window** menu:

Menu Option	Operation
Window->Cascade	Arrange all open windows so that they overlap, with the top left of each Editor window visible.
Window->Tile Horizontally	Arrange all open windows horizontally so that they occupy the entire Editor window, without any overlaps.
Window->Tile Vertically	Arrange all open windows vertically so that they occupy the entire Editor window, without any overlaps.
Window->Arrange Icons	Line up all minimized windows at the bottom of the Editor window.
Window->Close All	Close all open Editor windows.

The files within the editor can be displayed in a 'notebook' style. This means that each file has a separate tab associated with it to aid in navigating between files.

To show files in a notebook style

☑ Show files in notebook

1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Editor** tab.
3. Select the **Show files in notebook** checkbox as appropriate.
4. Click the OK button.

4.3 Standard file operations

4.3.1 Creating a new file

To create a new editing window

Select one of the following operations:

- Click the **New File** toolbar button (🗋), **OR**
- Press CTRL+N, **OR**
- Select the [**File**->**New**] menu option.

The new window will be given an arbitrary name by default. You can provide a new name when you save the file.

4.3.2 Editing a file

The HEW editor's standard editing functionality is available through the usual methods (i.e. the menu, toolbar and keyboard shortcuts) and is additionally supported via a pop-up menu that is local to each Editor window. Right-click in an open Editor window to invoke the pop-up menu.

The table below outlines the basic operations that are provided by the editor.

Operation	Effect	Action
Undo	Reverses the last editing operation	Select [**Edit->Undo**] Press CTRL+Z
Redo	Repeats the last undone editing operation	Select [**Edit->Redo**] Press CTRL+Y
Cut	Removes highlighted text and places it on the Windows® clipboard	Click the **Cut** toolbar button (✂) Select [**Edit->Cut**] Select **Cut** from the pop-up menu Press CTRL+X
Copy	Places a copy of the highlighted text into the Windows® clipboard	Click the **Copy** toolbar button (⎘) Select [**Edit->Copy**] Select **Copy** from the pop-up menu Press CTRL+C
Paste	Copies the contents of the Windows® clipboard into the active window at the position of the insertion cursor	Click the Paste toolbar button (📋) Select [**Edit->Paste**] Select **Paste** from the pop-up menu Press CTRL+V
Clear	Removes highlighted text (it is not copied to the Windows® clipboard)	Select [**Edit->Clear**] Press Delete
Select All	Selects (i.e. highlights) the entire contents of the active window	Select [**Edit->Select All**] Press CTRL+A

If you edit the file, the title bar of the editor window shows an asterisk (*). This asterisk remains until you save the file. If you undo all the changes made in the file, the asterisk disappears.

Selecting text in the editor

It is possible to select text in the same manner as all editors. However to access column selection hold down the ALT key while you are selecting the text with the mouse. This changes the selection technique from line to column selection.

4.3.3 Saving a file

To save the contents of an editing window

1. Ensure that the window, whose contents you want to save, is the active window.
2. Select one of the following operations:

- Click the **Save File** toolbar button (💾), **OR**
- Press CTRL+S, **OR**
- Select the [**File**->**Save**] menu option.

3. If the file has not been saved before, a **File Save** dialog will be displayed. Enter a filename, specify a directory and then click the OK button to create the file with the name given in the directory specified. If the file has been saved before then the file will be updated (no dialog box will be displayed).

To save the contents of an editing window under a new name

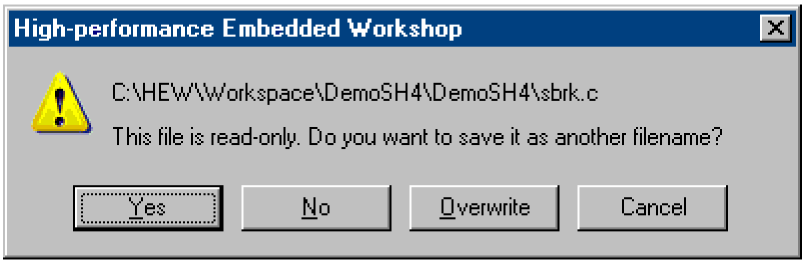
1. Ensure that the window, whose contents you want to save, is the active window.
2. Select the [**File**->**Save As**] menu option.

3. A **File Save** dialog box will be displayed. Enter a filename, specify a directory and then click the OK button to create the file with the name given, in the directory specified.

To save the contents of every open editor window

1. Select one of the following operations:
   - Click the **Save All** toolbar button (), **OR**
   - Select the [**File->Save All**] menu option.
2. If any of the files has not been saved before, a **File Save** dialog will be displayed. Enter a filename, specify a directory and then click the OK button to create the file with the name given in the directory specified. If any of the files have been saved before, then that file will be updated (no dialog will be displayed).

When you attempt to save a read-only file, a confirmation dialog box appears asking if you wish to save the file as another name.



High-performance Embedded Workshop

C:\HEW\Workspace\DemoSH4\DemoSH4\sbrk.c

This file is read-only. Do you want to save it as another filename?

- Clicking **Yes** opens the **Save file** dialog box. Change the file name before saving.
- Clicking **No** closes the file without saving.
- Clicking **Overwrite** saves the file by overwriting the contents. The file is no longer read-only.
- Clicking **Cancel** cancels the attempt to save the file.

4.3.4 Opening a file

To open a file

1. Select one of the following operations:
   - Click the **File Open** toolbar button (), **OR**
   - Press CTRL+O, **OR**
   - Select the [**File->Open**] menu option.
2. A **File Open** dialog box will be displayed. Use the directory browser to navigate to the directory in which the file you want to open is located. Use the **Files of Type** combo box to select the type of file you want to open (or set it to **All Files (*.*)** to see every file in a directory).
3. Once you have located the file, select it and click **Open**.

To open a file, drag the file from the Windows® Explorer and drop it into the HEW main window.

To open a source file on the **Projects** tab of the workspace window, select one of the following ways:

- Double-click the file, **OR**
- Select the file and click the right-hand mouse button. Select **Open <file name>** from the pop-up menu, **OR**
- If the file has already been selected (focused), press **Enter**.

You can also use another editor (external editor) to open files. For details, see section 6.9,Using an external editor.

The High-performance Embedded Workshop keeps track of the last four files that you have opened and adds them to the File menu under the [**File->Recent Files**] sub-menu.

To open a recently used file

Select the [**File->Recent Files**] menu option and from this sub-menu select the desired file.

4.3.5 Closing files

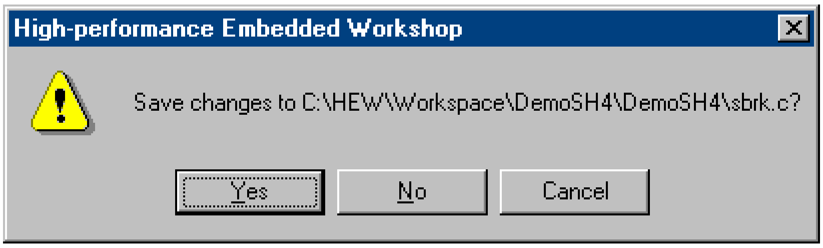
To close an individual file

- Double-click the **Editor** window's system menu (located at the top left of each window), **OR**
- Click on the **Editor** window's system menu (located at the top left of each window) and select the **Close** menu option, **OR**
- Ensure that the window that you want to close is the active window and then press CTRL+F4, **OR**
- Ensure that the window that you want to close is the active window and then select the [**File**->**Close**] menu option.

To close all files

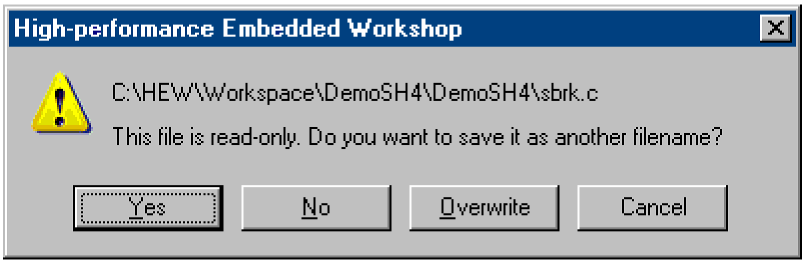
Select the [**Window**->**Close All**] menu option.

If you attempt to close an edited file that is not saved, a confirmation dialog box appears asking if you wish to save the file.



Save changes to C:\HEW\Workspace\DemoSH4\DemoSH4\sbrk.c?

- Clicking **No** closes the file without saving.
- Clicking **Cancel** returns to the editor window.
- Clicking **Yes** closes the file by overwriting the contents. If the file has the read-only attribute, a confirmation dialog box appears asking if you wish to save the file as another name.



C:\HEW\Workspace\DemoSH4\DemoSH4\sbrk.c

This file is read-only. Do you want to save it as another filename?

- Clicking **Yes** opens the **Save file** dialog box. Change the file name before saving.
- Clicking **No** closes the file without saving.

- Clicking **Overwrite** closes the file by overwriting the contents.
- Clicking **Cancel** returns to the editor window.

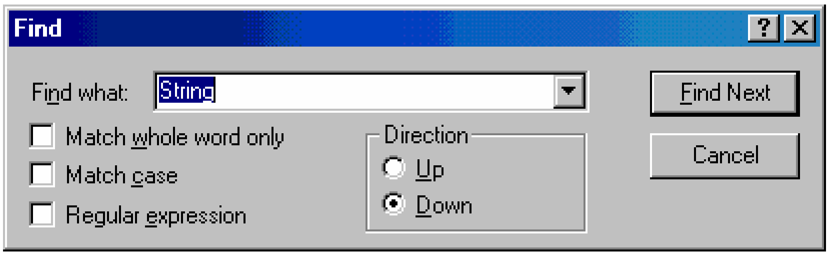
4.4 Searching and navigating through files

4.4.1 Finding text

To search for text in the current file

1. Ensure that the window, whose contents you want to search, is the active window.
2. Position the insertion cursor at the point from which you want to start your search.
3. Select one of the following operations to display the **Find** dialog box:

   - Click the **Find** toolbar button (), **OR**
   - Press the CTRL+F key, **OR**
   - Select the [**Edit->Find**] menu option, **OR**
   - Select the **Find** from the pop-up menu in the Editor window.



Find what: String

□ Match whole word only

□ Match case

□ Regular expression

Direction

○ Up

● Down

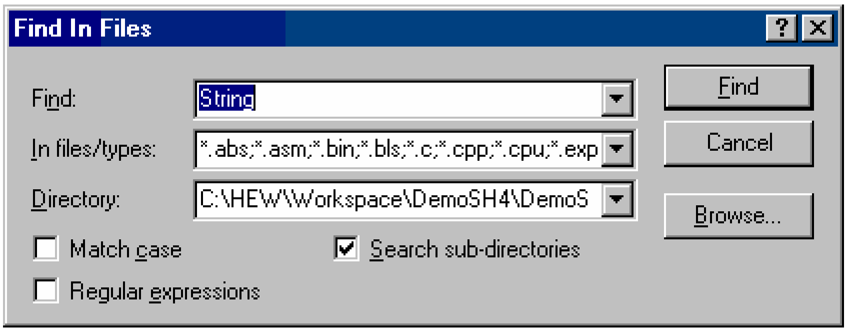
4. Enter the text that you want to search for into the **Find what** field, or select a previous search string from the drop-down list box. If you select text before invoking the find operation, the selected text will be automatically placed into the **Find what** field.
5. If you would like to search for character string as a whole word then click the **Match whole word only** checkbox. When this option is not selected, the search will be for any string that is matched by the search string.
6. If you would like your search to be case-sensitive (i.e. to distinguish between upper and lower case letters) then check the **Match case** checkbox.
7. If your search string uses regular expressions then check the **Regular expressions** checkbox. See Reference 4, Regular Expressions, for further information.
8. The **Direction** radio buttons allow you to select the direction of the search. Selecting **Down** means that the search will be performed from the insertion cursor towards the bottom of the file. Selecting **Up** means that the search will be performed from the insertion cursor towards the top of the file.
9. Click the **Find Next** button to begin the search. Click the Cancel button to stop the **Find** action.

4.4.2 Finding text in multiple files

To search for text in multiple files

1. Select one of the following operations to display the **Find In Files** dialog box:

   - Click the **Find In Files** toolbar button (), **OR**
   - Press the F4 key, **OR**
   - Select the [**Edit->Find in Files**] menu option.



Find: String

In files/types: *.abs;*.asm;*.bin;*.bls;*.c;*.cpp;*.cpu;*.exp

Directory: C:\HEW\Workspace\DemoSH4\DemoS

□ Match case ☑ Search sub-directories

□ Regular expressions

2. Enter the text that you want to search for into the **Find** field, or select a previous search string from the drop-down list box. If you selected text before invoking the find operation, the selected text will be automatically placed into the **Find** field.
3. Enter the file extensions of the files you would like to search into the **In files/types** field. If several extensions are specified, be sure to separate them with a comma (e.g. *.C, *.H).
4. Enter the directory that contains the files to search, into the **Directory** field. Alternatively you may browse to the desired directory graphically if you click the **Browse** button.
5. If you would like your search to be case-sensitive (i.e. to distinguish between upper and lower case letters) then check the **Match case** checkbox.
6. If you would like to search the directory specified and all directories below it then check the **Search sub directories** checkbox. If you just want to search the single directory specified in **Directory** field then ensure that this checkbox is not checked.
7. If your search string uses regular expressions then check the **Regular expressions** checkbox. See Reference 4, Regular Expressions, for further information.
8. Click **Find** to begin the search. Any matches found will be displayed in the **Find In Files** tab of the Output window. To stop a **Find In Files** action once it is under way, select the [**Edit->Stop Find in Files**] menu option. Once the **Find In Files** operation is complete, you may jump to an instance of the search string by double-clicking on the desired entry in the Output window.

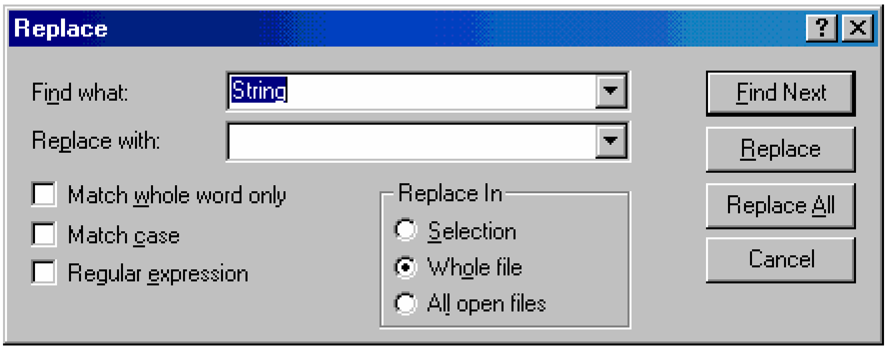
4.4.3 Replacing text

Replacing text is similar to finding text, as discussed in the previous section. The difference is that when the text is found you have the option to replace it with other text.

To replace text in a file

1. Ensure that the window, whose contents you want to replace, is the active window.
2. Position the insertion cursor at the point from which you want to start your search.
3. Select one of the following operations to display the **Replace** dialog box:

- Press the CTRL+H key, **OR**
- Select the [**Edit->Replace**] menu option, **OR**
- Select **Replace** from the pop-up menu in the Editor window.



Find what: String

Replace with:

☐ Match whole word only

☐ Match case

☐ Regular expression

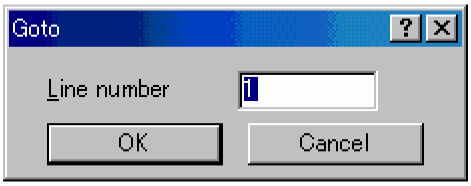
Replace In
○ Selection
● Whole file
○ All open files

4. Enter the text that you want to search for into the **Find what** field, or select a previous search string from the drop-down list box. If you select text before invoking the replace operation, the selected text will be automatically placed into the **Find what** field.
5. Enter the text that you want to replace the search string with, or select a previous replace string from the drop-down list box.
6. If you would like to search for character string as a whole word then click the **Match whole word only** checkbox. When this option is not selected, the search will be for any string that is matched by the search string.
7. If you would like your search to be case-sensitive (i.e. to distinguish between upper and lower case letters) then check the **Match case** checkbox.
8. If your search string uses regular expressions then check the **Regular expressions** checkbox. See Reference 4, Regular Expressions, for further information.
9. If you clicked the **Find Next** button, the editor will search for the first occurrence of the search string. Click the **Replace** button if you want to replace it. Click **Replace All** button to replace all occurrences or click the Cancel button to stop the replace action. If you select **Selection** in the **Replace In** field, the replace action will be performed in the range of the selected text. If you select **Whole file**, the replace action will be performed on the whole file. If you select **All open files**, all files that are currently open in the editor will have the replace operation carried out on them.

4.4.4 Jumping to a specified line

To jump to a line in a file

1. Ensure that the window, whose contents you want to replace, is the active window.
2. Select one of the following operations to display the **Goto** dialog box:
   - Press CTRL+G, **OR**
   - Select the [**Edit->Goto Line**] menu option, **OR**
   - Select **Goto Line** from the pop-up menu in the Editor window.



Line number: 1

3. Enter the number of the line that you want to jump to into the **Line number** box and then click the OK button.
4. The insertion cursor will be placed at the start of the line number specified.

4.5 Bookmarks

When working with many large files at a time, it can become difficult to locate specific lines or areas of interest. Bookmarks enable you to specify lines that you want to jump back to at a subsequent time. One example of its use is in a large C file where you may want to set a bookmark on each function definition. Once a bookmark has been set, it exists until it is removed or the file is closed.

To set a bookmark

1. Place the insertion cursor on the line to mark.
2. Select one of the following operations:
   - Click the **Toggle Bookmark** toolbar button (📖), **OR**
   - Press CTRL+F2, **OR**
   - Select the [**Edit**->**Bookmarks**->**Toggle Bookmark**] menu option, **OR**
   - Right-click and select the [**Bookmarks**->**Toggle Bookmark**] menu option from the pop-up menu.
3. A bookmark icon (🟦) will be placed on the bookmarked line, to indicate that it is an active bookmark.

To remove a bookmark

1. Place the insertion cursor on the marked line.
2. Select one of the following operations:
   - Click the **Toggle Bookmark** toolbar button (📖), **OR**
   - Press CTRL+F2, **OR**
   - Select the [**Edit**->**Bookmarks**->**Toggle Bookmark**] menu option, **OR**
   - Right-click and select the [**Bookmarks**->**Toggle Bookmark**] menu option from the pop-up menu.
3. The bookmark icon will be removed from the line.

To jump to the next bookmark in a file

1. Ensure that the insertion cursor is somewhere within the file to be searched.
2. Select one of the following operations:
   - Click the Next Bookmark toolbar button (📖), **OR**
   - Press F2, **OR**
   - Select the [**Edit**->**Bookmarks**->**Next Bookmark**] menu option, **OR**
   - Right-click and select the [**Bookmarks**->**Next Bookmark**] menu option from the pop-up menu.

To jump to the previous bookmark in a file

1. Ensure that the insertion cursor is somewhere within the file to be searched.
2. Select one of the following operations:
   - Click the **Previous Bookmark** toolbar button (📖), **OR**
   - Press SHIFT+F2, **OR**
   - Select the [**Edit**->**Bookmarks**->**Previous Bookmark**] menu option, **OR**

- Right-click and select the [**Bookmarks**->**Previous Bookmark**] menu option from the pop-up menu.

To remove all bookmarks

1. Ensure that the window, whose bookmarks you want to remove is the active window.
2. Select one of the following operations:

- Click the **Clear All Bookmarks** toolbar button (), **OR**
- Select the [**Edit**->**Bookmarks**->**Clear All Bookmarks**] menu option, **OR**
- Right-click and select the [**Bookmarks**->**Clear All Bookmarks**] menu option from the pop-up menu.

4.6 Printing a file

To print a file

1. Ensure that the window, whose contents you want to print, is the active window.
2. Select one of the following operations:

- Click the **Print** toolbar button (🖨), **OR**
- Press CTRL+P, **OR**
- Select the [**File**->**Print**] menu option.

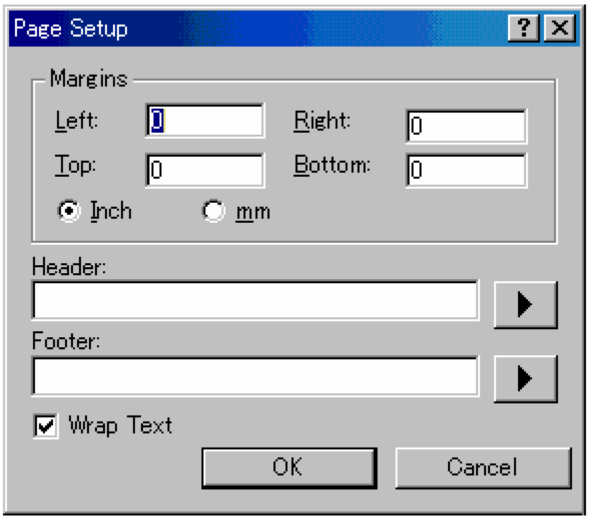
4.7 Configuring text layout

4.7.1 Page set-up

When you print a file from the HEW editor, the settings in the print dialog affect the way the file is printed (e.g. double or single sided). Control over how the text is formatted on the page can also be controlled via the **Page Setup** option. This allows you to specify the margins (top, bottom, left and right) of your printouts. It is often necessary to set this because some printers cannot print to the edges of an A4 page. Furthermore, some users have their own layout requirements (e.g. a large left-hand margin so that code can be placed in an A4 binder).

To set up the page margins

1. Select the [**File->Page Setup**] menu option. The **Page Setup** dialog box will be invoked.
2. Enter the width of the margins required into the **Left**, **Right**, **Top** and **Bottom** fields.
3. Set the **Inch** or **mm** option accordingly.
4. Click the OK button for the new settings to take effect.



Page Setup

Margins

Left: 0 Right: 0

Top: 0 Bottom: 0

● Inch ○ mm

Header:

Footer:

☑ Wrap Text

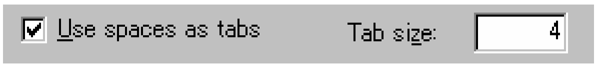
To set up the header and footer information

1. Select the [**File->Page Setup**] menu option. The **Page Setup** dialog box will be invoked.
2. Enter into the **Header** and **Footer** edit fields the text required to be displayed. All normal placeholders are available along with page numbering, text justification and date fields. These are all expanded before the page is to be printed.
3. Click the OK button for the new settings to take effect.

To set up print wrapping

1. Select [**File->Page Setup**]. The **Page Setup** dialog box will be invoked.
2. Click the **Wrap Text** check box. This switches on the wrap text facility when printing so no text is truncated and everything is visible.
3. Click OK for the new settings to take effect.

4.7.2 Changing tabs



☑ Use spaces as tabs   Tab size: 4

To change tab size

1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Editor** tab.
3. Enter into **Tab Size** the number of desired tabs.
4. Click the OK button for the new tab settings to take effect.

When the TAB key is pressed in the editor a tab character is usually stored in the file. However, sometimes it is preferable to store spaces instead. The representation of tab characters can be controlled via the **Options** dialog box.

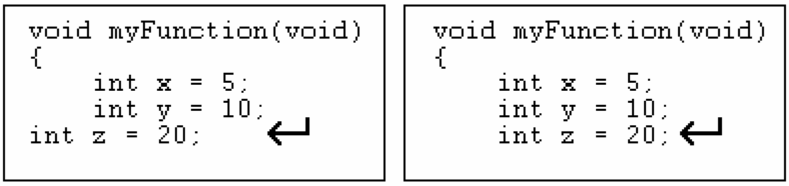
To use spaces as tabs

1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Editor** tab.
3. Set the **Use spaces as tabs** checkbox as appropriate.
4. Click the OK button for the new tab settings to take effect.

4.7.3 Auto indentation

When you press **Enter** the editor, the insertion cursor will move to the next line down, at the first column (i.e. against the left-hand side of the window). **Auto Indentation** is a feature which, when RETURN is pressed, places the insertion cursor on the next line (as before) but under the first non-whitespace character of the previous line. This enables you to type neat C/C++ or assembler code faster as you don't have to type leading spaces or tabs yourself.

The figure below illustrates two examples. The first shows the effect of pressing Enter when the **Auto Indentation** feature is disabled – the insertion cursor returns to the left-hand side of the window on the next line. When the 'int z = 20' line is typed, it is not aligned with the previous two lines. The second example shows the effect of pressing Enter when **Auto Indentation** is enabled – the insertion cursor drops underneath the 'i' of the 'int' word on the previous line. Now, when the 'int z = 20' line is typed, it is automatically aligned (i.e. automatically indented).



```
void myFunction(void)
{
    int x = 5;
    int y = 10;
int z = 20;     ↵
```

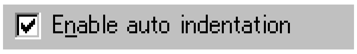
(i) Auto Indentation Disabled

```
void myFunction(void)
{
    int x = 5;
    int y = 10;
    int z = 20; ↵
```

(ii) Auto Indentation Enabled

To enable/disable Auto Indentation

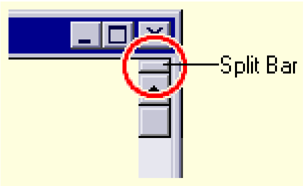
1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Editor** tab.
3. Set the **Enable auto indentation** checkbox. By default, this checkbox is selected.
4. Click the OK button for the new settings to take effect.



☑ Enable auto indentation

4.8 Splitting a window

The HEW editor allows you to split an editor window into two. The split bar button is located just underneath the maximize button at the top right-hand corner of any editor window (as shown below).



To split a window

Double-click the split bar button to split the window in half, or click on the split bar button, keep the button pressed, drag the mouse down and then release the mouse button at the point you want to split the window.

To adjust the position of the split bar

Click on the split bar itself, keep the button pressed then move the bar to the new position and then release the button.

To remove the split bar

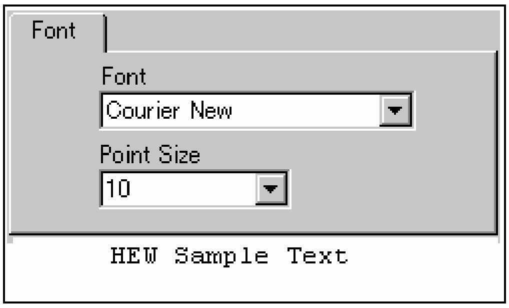
Double-click on the split bar or move the split bar to the top or bottom of the window.

4.9 Changing the editor font

The High-performance Embedded Workshop allows you to specify the font to be used in its internal editor. All editor windows, regardless of the file type, use the same font.

To change the editor font

1. Select [**Setup->Format Views**]. The **Format Views** dialog box will be displayed.
2. Select the **Source** icon in the tree. The **Font** tab should be available on the right of the **Format Views** dialog box.
3. Select the font type from the **Font** list.
4. Select the font size from the **Point Size** list.
5. When this is being modified the sample text below shows what the font will look like.
6. Click the OK button.



Font: Courier New

Point Size: 10

HEW Sample Text

4.10 Syntax coloring

To enhance code readability, the HEW editor can display specific strings (i.e. keywords) in different colors. For instance, C source code comments could be shown in green, and C types (e.g. int) could be shown in blue.

The coloring method used can be specified on a file group by file group basis. For example, you can define different color schemes for C source file, C header file, Assembly source file, or Assembly include file.

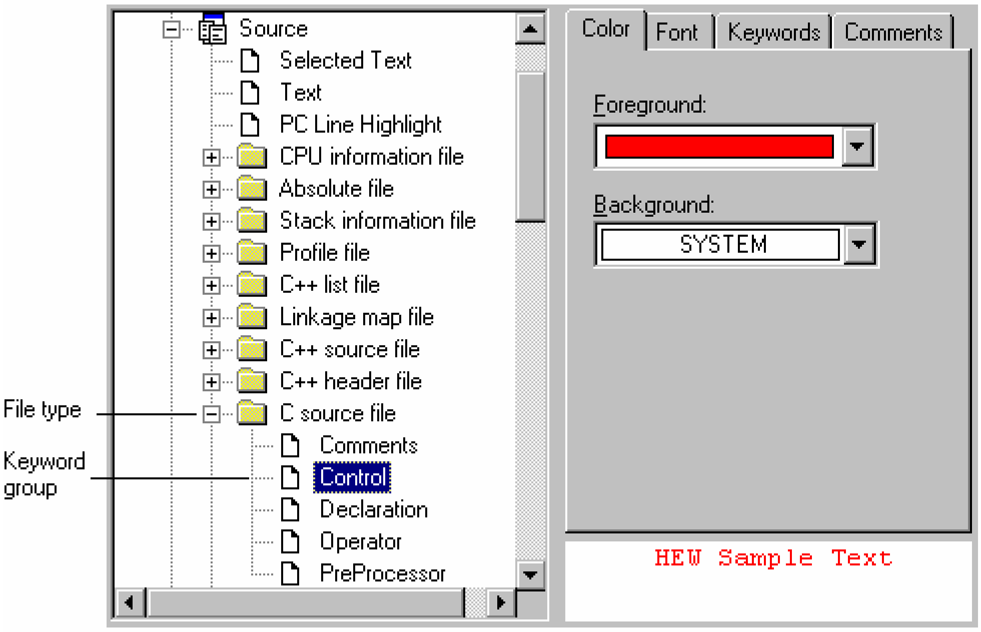
Note:

When you create a new file, syntax coloring will not be active, as a new file does not initially have an extension (new files are named arbitrarily by the editor without an extension). In order to activate syntax coloring, you must save the new file with a name and extension that the HEW recognizes. See section 2.5, File extensions and file groups, for information on file extensions.

4.10.1 Changing text colors

To change existing colors

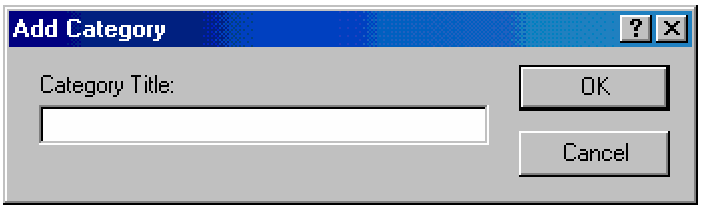
1. Select the [**Setup->Format Views**] menu option. The **Format Views** dialog box will be displayed.
2. Select the view you are interested in changing the font for. If it is the editor expand the "Source" in the tree on the left hand side of the dialog box.
3. Select the file type for which you want to edit syntax coloring from the File group list and then expand and select it.
4. Select the category are interested in. The tabs on the right side of the dialog box change depending on the selection.
5. Select the **Color** tab.
6. Modify the **Foreground** and **Background** color lists as desired. The **System** color refers to the current window foreground and background settings in Control Panel.
7. Click the OK button for the new color settings to take effect.



4.10.2 Creating new keywords

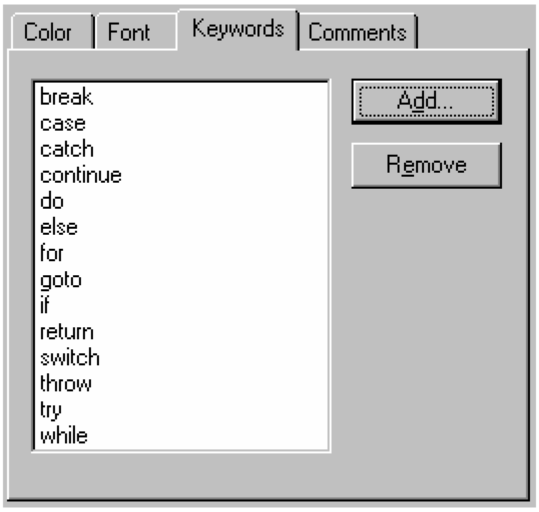
To create new keyword groups

1. Select the [**Setup->Format Views**] menu option. The **Format Views** dialog box will be displayed. Expand the **Source** view icon in the tree.
2. Select the file type for which you want to create a new keyword group from the tree on this dialog box.
3. Click the **Add** button underneath the tree. The **Add Category** dialog box will then be displayed. Enter the name of the group into the **Category Title** field. Click the OK button to add a keyword group. To modify the name of the group, select the keyword group and click the **Modify** button underneath the tree. **Modify Category** dialog box will then be displayed. Enter the name of the group into the **Category Title** field. To remove a keyword group from the tree, select the keyword group and click the **Remove** button underneath the tree.



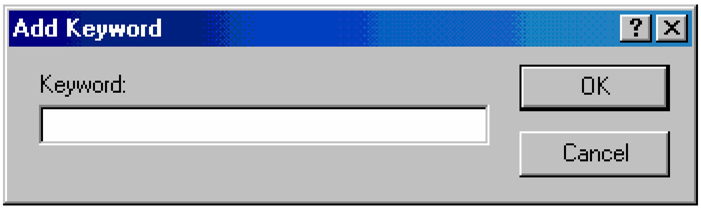
To create new keywords

1. Select the [**Setup->Format Views**] menu option. The **Format Views** dialog box will be displayed.
2. Select the desired keyword group to be modified and click the **Keywords** tab.



break
case
catch
continue
do
else
for
goto
if
return
switch
throw
try
while

3. Click the **Add** button to add a keyword. Then the **Add Keyword** dialog box will be launched. Specify a keyword in the **Keyword** field and click the OK button to close the dialog box. To remove a keyword from the Keywords list, select the keyword and click the **Remove** button.



4. Click the OK button.

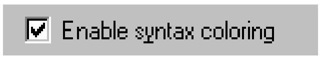
Note:

On the **Keyword** field of the **Add Keyword** dialog box, specify a keyword which consists of only alphanumeric, an underscore, and the # character.

4.10.3 Enabling/disabling syntax coloring

To enable/disable syntax coloring

1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Editor** tab.
3. Set the **Enable syntax coloring** checkbox as necessary and then click the OK button.

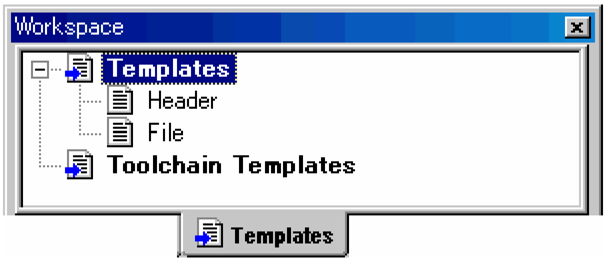


☑ Enable syntax coloring

4.11 Templates

When developing software it is often necessary to enter the same text repeatedly, for instance, when typing a function definition, for loop or a comment block for a function. The HEW editor allows you to specify a block of text (or template) which can be inserted into the currently active Editor window. Thus, once a template has been defined, it can be automatically inserted without the need to re-enter it manually.

Figure below shows a list of templates, which is located on the **Templates** tab of the Workspace window.



Any new templates, which have been added to the HEW, are displayed under the **Templates** folder. The **Toolchain Templates** folder is for templates, which are read only and have been provided for use in the HEW system by the current toolchain.

Templates in this view can be dragged for insertion into an editor file. It is also possible to drag an area of text from the editor into the templates folder for quick template creation.

Right-clicking on the Templates folder opens a pop-up menu containing the available options.

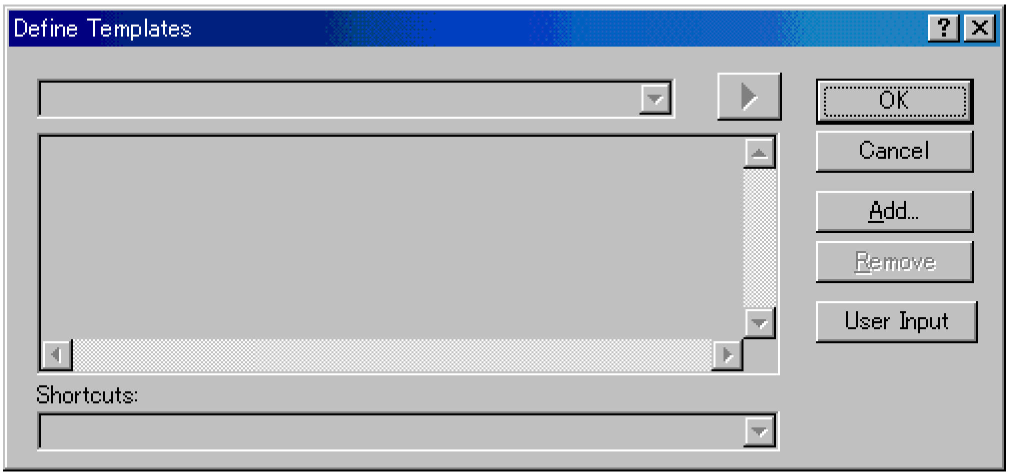
Pop-up Menu Options	Function
Add	Adds a new template.
Remove	Removes the current selection.
Edit	Edits the current selection.

4.11.1 Defining a template

To define a template

1. Select one of the following operations to display the **Define Templates** dialog box:

- Click the **Define Template** toolbar button (), **OR**
- Select the [**Edit->Templates->Define Templates**] menu option, **OR**
- Right-click on the **Templates** in the Templates tab of the workspace window and select **Edit** from the pop-up menu, **OR**
- Right-click in the editor window and select [**Templates->Define Templates**] from the pop-up menu.



2. Click the **Add** button. The **Add Template** dialog box is displayed, which asks you to enter your chosen template name. This name must be unique, otherwise a duplicated template name message will be displayed and the template will not be added.
3. If you want to modify an existing template use the **Template Name** drop-down menu to select which template you want to modify.
4. Enter the desired text into the **Template Text** area. You can copy text from another Editor window and then paste it into this dialog box using CTRL+V.
5. Enter the following placeholder to insert special information when the template is inserted:

Menu Entry	Placeholder	Replaced With
File path + name	$(FULLFILE)	Filename (including full path)
File name	$(FILENAME)	Filename (excluding path, including extension)
File leaf	$(FILELEAF)	Filename (excluding path and extension)
Workspace name	$(WORKSPNAME)	Workspace name
Project name	$(PROJECTNAME)	Current project name
Line number	$(LINE)	First line number of template insertion
Time	$(TIME)	Current time
Date, text	$(DATE_TEXT)	Current date in text form
Date, day/month/year	$(DATE_DMY)	Current date in dd/mm/yy form
Date, month/day/year	$(DATE_MDY)	Current date in mm/dd/yy form
Date, year/month/day	$(DATE_YMD)	Current date in yy/mm/dd form
User name	$(USER)	Current Windows® user
Cursor position	$(CURSOR)	Insertion cursor – positions the cursor in this position after template has been inserted

6. Enter ‘$(CURSOR)’ to specify where the insertion cursor is to be placed after the template has been inserted. If this is not specified then the insertion cursor will be placed after the last character in the template, as in a normal paste operation.
7. There are 10 shortcut keys reserved for templates. If you want to designate one of these select the key in the drop-down list at the bottom of the edit template dialog box. These range from ALT+0 to ALT+9.

User input

When defining a template it is possible to define a user input field. Using the following placeholder specifies this $(USERINPUT<n:1-10>|”<some text>”). The 'n' is a number, which identifies the user input identifier.

These placeholders can be added manually but the **User Input** button on the **Define Templates** dialog box adds these placeholders in an automated manner. When the template is inserted into a file a dialog is displayed which allows you to enter some custom text for each of these fields. This text is then inserted instead of the placeholder. You can define ten of these user input fields.

4.11.2 Deleting a template

To delete a template

1. Select one of the following operations to display the **Define Templates** dialog box:
   - Click on the **Define Template** toolbar button (), **OR**
   - Select the [**Edit->Templates->Define Templates**] menu option, **OR**
   - Right-click on the **Templates** in the Templates tab of the workspace window and select **Edit** from the pop-up menu, **OR**
   - Right-click in the editor window and select [**Templates->Define Templates**] from the pop-up menu.
2. Use the **Template Name** drop-down list to select the name of the template you wish to remove and then click the **Remove** button.
3. Clicking the OK button saves the changes and closes the dialog box.

To remove selected templates using the Templates tab of the Workspace window

1. Select the templates that you want to remove in the **Templates** tab of the Workspace window. Multiple templates can be selected by holding down the SHIFT or CTRL key.
2. Right-click on the **Templates** within the Templates tab of the workspace window and select **Remove** from the pop-up menu.

4.11.3 Inserting a template

To insert a template

1. Select one of the following operations to display the **Insert Template** dialog box:
   - Click the **Insert Template** toolbar button (), **OR**
   - Select the [**Edit->Templates->Insert Template**] menu option, **OR**
   - Right-click in the editor window and select [**Templates->Insert Templates**] from the pop-up menu.
2. Use the **Template Name** drop-down list to select the name of the template to be inserted, and then click the OK button. The dialog box is closed and the chosen template is added to the current Editor window.

Alternatively, you can press ALT along with the number of the template to be inserted (e.g. ALT+4 to insert template 4). You can define these shortcuts on the **Defile Templates** dialog box. A drop-down list is available at the bottom of the **Defile Templates** dialog box.

4.12 Brace matching

Complicated source code can often become unwieldy, especially when blocks of C /C++ code are deeply nested within each other, or when complex logic statements are expressed within a large ‘`if`’ clause. To help in such situations, the High-performance Embedded Workshop editor provides a **Brace Matching** feature, which highlights text between braces of type { }, ( ) and [ ].

To find a matching brace

1. Either highlight the open brace to match from or place the cursor before it.
2. Select one of the following operations:
   - Click the **Match Braces** toolbar button ({→}), **OR**
   - Press CTRL+SHIFT+M, **OR**
   - Select the [**Edit->Match Braces**] menu option, **OR**
   - Select the **Match Braces** menu option from the pop-up menu.

To check the structure of an entire file, place the cursor at its start and then repeatedly invoke the match brace operation. The editor will successively highlight each pair of braces in turn until there are no more left to match.

4.13 Setting the read-only attribute for a file

While using the HEW, you can set a file to be read-only. When you attempt to save a read-only file, a confirmation dialog box appears asking if you wish to save the file as another name.

To set a file to be read-only

1. Select one of the following operations:

- Right-click on a file in the **Projects** tab of the workspace window to open a pop-up menu. You can even select multiple files, **OR**
- Right-click within the editor window to open a pop-up menu.

2. Selecting **Properties** opens the **Properties** dialog box.
3. Select the **Read-only** checkbox. If you have selected two or more files including both read-only and writable files, the checkbox is gray (intermediate state). By default, this checkbox is not selected.
4. The title bar of the editor window shows “Read-Only”.

When a file is open within the editor window and if you switch on/off the read-only attribute of the file via the Windows Explorer, it does not match the attribute shown on the title bar of the editor window (because the title bar is not updated). The attribute shown on the title bar is not updated until you start modifying the contents or re-open the file.

4.14 Preventing modification of files while debugging

Use this option if you wish to prevent modification of files while debugging with HEW and a debugging platform connected. This prevents modification of all files that are open in the HEW editor.

To prevent modification of files while debugging

1. Select [**Setup->Options**] to open the **Options** dialog box.
2. Select the **Editor** tab.
3. Selecting the **Do not allow file editing when target is connected** checkbox prevents modification of files while the HEW is connected to a debugging platform. In other words, if the HEW is not connected to a debugging platform, files can be modified. By default, this checkbox is not selected.
4. Click OK.

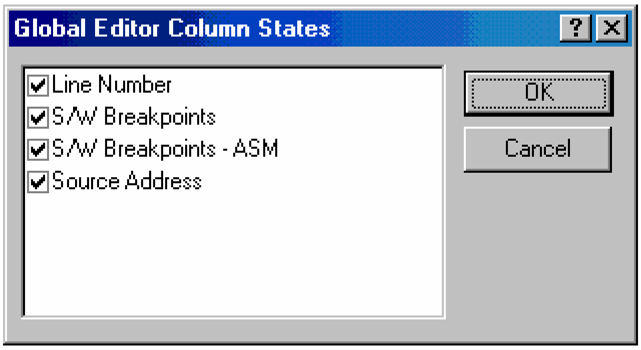
4.15 Managing the editor columns

The editor in HEW has the ability to manage columns (apart from the main editor column). These can be added and used by any component in the HEW system. Examples of this functionality might include a hardware breakpoint column added by the target, or possibly an address information column added by the debugger. The global column states feature is also accessible from the main edit menu.

To switch off a column in all source files

1. Select one of the following operations to display the **Global Editor Column States** dialog box:

- Select the [**Edit->Define Column Format**] menu option, **OR**
- Right-click in the editor window and select **Define Column Format** from the pop-up menu.



Global Editor Column States

☑ Line Number
☑ S/W Breakpoints
☑ S/W Breakpoints - ASM
☑ Source Address

2. If the column's checkbox is checked then the column is enabled; if the column's checkbox is gray then this means that the column is enabled in some files, and disabled in others.
3. Click the OK button for the new column settings to take effect.

To switch off a column in one source file

1. Open the editor window of the file which you wish to remove a column from.
2. Select one of the following operations:

- Right-clicking the column header displays a pop-up menu. A tick mark right next to an entry indicates that this column is displayed. Clicking an entry will switch showing/hiding the column, **OR**
- Right-click in the Editor window and select the **Columns** menu item. The cascaded menu item appears. Each column is displayed in this pop-up menu. If the column is enabled it will have a tick next to its name. Clicking on the entry will toggle whether the column is displayed or not.

You can adjust the column width by dragging the mouse on a column header.

4.16 Showing/hiding the column header

The editor window has a column header. You can select to show or hide the column header.

To switch showing/hiding the column header

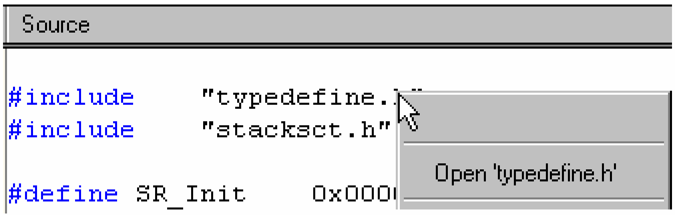
1. Right-click within a field in the editor window showing source or disassembly codes.
2. A pop-up menu opens. Select **Turn Header On/Off**.

4.17 Opening a file within the editor

The HEW allows you to open a file within the editor window.

To open a file shown in the editor window

1. Right-click on the file name in the **Source** field in the editor window.



```
#include     "typedefine.h"
#include     "stacksct.h"

#define SR_Init     0x000
```

2. Select **Open '<file name>'** from the pop-up menu.

4.18 Tooltip watch

Use this function to know the value of a variable defined in the source program. Open the editor window or disassembly window (in source mode) to view the source program and rest the mouse cursor over the variable name that you want to examine. A tooltip (pop-up window) will appear showing the watch information.

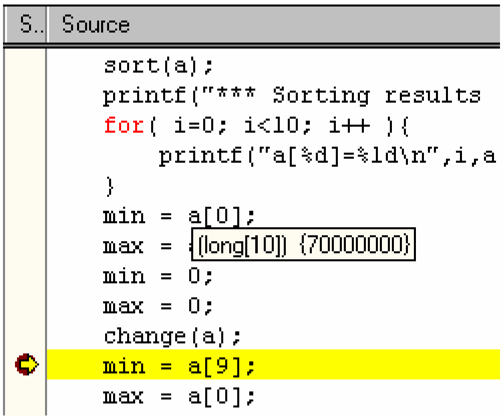
To use Tooltip watch

1. Select [**Setup->Options**] menu option. The **Options** dialog box is displayed.
2. Select the **Editor** tab.
3. Check the **Enable tooltip watch** check box.
4. Click OK.

☑ Enable tooltip watch

To view a tooltip watch on the editor window or disassembly window (in source mode)

1. Open the editor window or disassembly window (in source mode) showing the variable that you want to examine.
2. Rest the mouse cursor over the variable name that you want to examine. A tooltip will appear near the variable containing basic watch information for that variable.



```
    sort(a);
    printf("*** Sorting results
    for( i=0; i<10; i++ ){
        printf("a[%d]=%ld\n",i,a
    }
    min = a[0];
    max = (long[10]) {70000000}
    min = 0;
    max = 0;
    change(a);
    min = a[9];
    max = a[0];
```

4.19 Evaluate an expression

Launches the **Evaluate** dialog box allowing the user to enter a numeric expression, e.g. "205*2", and display the result in all currently supported radices.

To evaluate an expressions

1. Select the [**Edit->Evaluate**] menu option.
2. Launches the **Evaluate** dialog box.
3. Enter the expression that you wish to evaluate and click the **Evaluate** button.

Provides a calculator function, evaluating simple and complex expressions, with parentheses and symbols. All operators have the same precedence but parentheses may be used to change the order of evaluation. The operators have the same meaning as in C/C++. Expressions can also be used in any command where a number is required. The result is displayed in all supported radix types.

Valid operators:

Addition (+)	Subtraction (-)	Multiplication (*)	Division (/)
Logical AND (&&) ***1**	Logical OR (||) ***1**	Logical NOT (!) ***1**	Equal to (==)
Bitwise AND (&)	Bitwise OR (|)	Bitwise NOT (~)	Unequal to (!=)
Left arithmetic shift (<<)	Right arithmetic shift (>>)	Less than (<)	Greater than (>)
Modulo (%) ***1**	Bitwise exclusive OR (^)	Less than or equal to (<=)	Greater than or equal to (>=)

***1.** Support for this function depends on the debugging platform.

Register names:

- **SuperH/H8/R8C (E7/E8) family**

  Register names may be used, but must always be prefixed by the '#' character.

  Enter a numeric expression, e.g. "(#pc+205)*2", and display the result in all currently supported radices.

- **M32C/M32R/M16C/R8C (excluding E7/E8) family**

  Register names may be used, but must always be prefixed by the '%' character.

  Enter a numeric expression, e.g. "(%pc+205)*2", and display the result in all currently supported radices.

Character Constants:

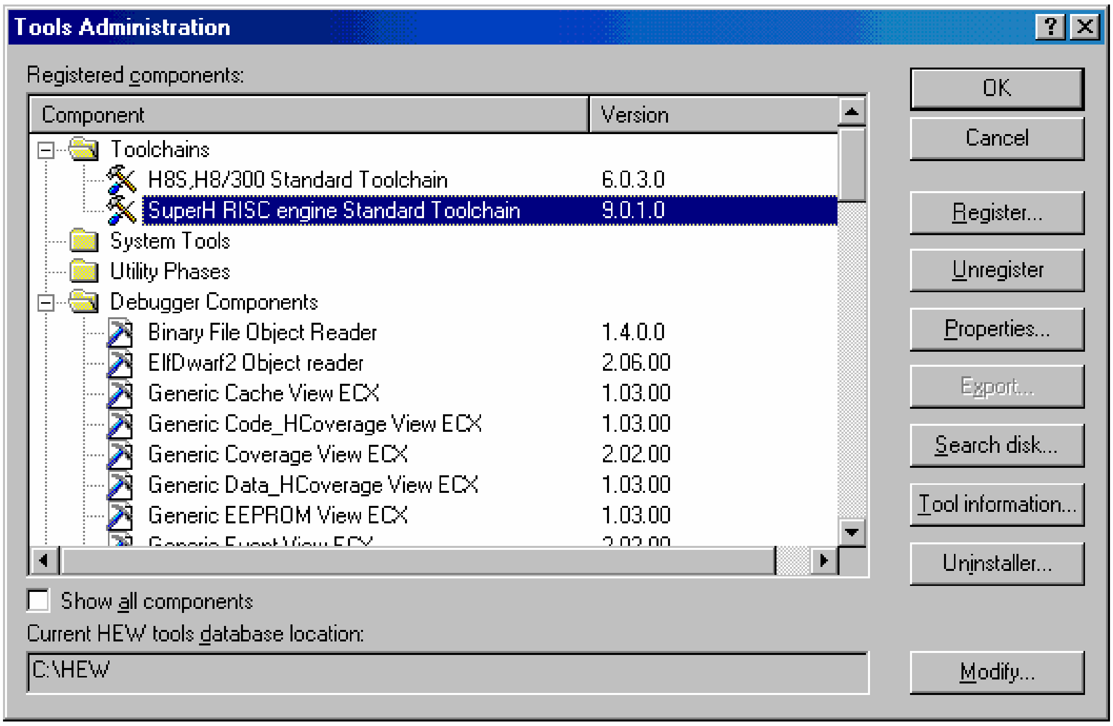
Characters enclosed in single quote marks (') may be use as character constants. For example, 'A', etc. These character constants are converted to ASCII code and used as 1-byte immediate values.

Character String Literals:

Character strings enclosed in double quote marks (") may be use as character string literals. Examples are "abc", etc.

5. Tools Administration

You can control the components that are used by the High-performance Embedded Workshop via the **Tools Administration** dialog box, which is invoked by selecting the [**Tools->Administration**] menu option. The **Tools Administration** dialog box is only accessible when no workspace is open.



Component	Version
Toolchains	
H8S,H8/300 Standard Toolchain	6.0.3.0
SuperH RISC engine Standard Toolchain	9.0.1.0
System Tools	
Utility Phases	
Debugger Components	
Binary File Object Reader	1.4.0.0
ElfDwarf2 Object reader	2.06.00
Generic Cache View ECX	1.03.00
Generic Code_HCoverage View ECX	1.03.00
Generic Coverage View ECX	2.02.00
Generic Data_HCoverage View ECX	1.03.00
Generic EEPROM View ECX	1.03.00
Generic Event View ECX	2.02.00

☐ Show all components

Current HEW tools database location:

C:\HEW

There are five standard types of component:

Toolchain
A Toolchain is a set of build phases (e.g. compiler, assembler, linker and librarian). These components provide the build capability.

System Tool
A System Tool is an application (.EXE) that can be launched from the **Tools** menu. These are often provided as extra applications, which support the toolchain (e.g. an external debugger like the Hitachi Debugging Interface or an interactive graphical librarian).

Utility Phase
A Utility Phase is a ‘ready made’ build phase, which supports some specific build functionality (e.g. analyze complexity of source code, count lines of source code, etc.). These components provide added functionality to the build that is not toolchain-specific.

Debugger Component
A Debugger Component is a component that supports some specific debugger functionality (e.g. CPU DLL, Target platform, Object reader, etc.).

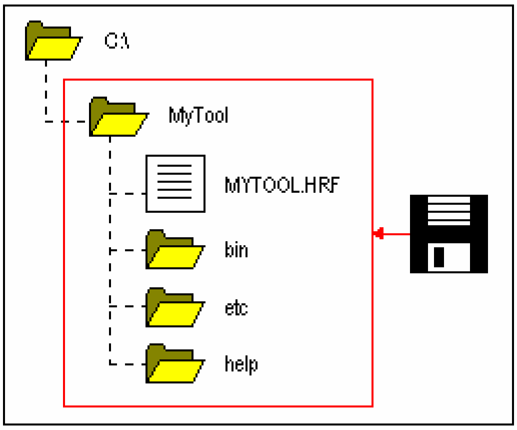
Extension Component
An Extension Component is a component that provides key functionality in a certain area of the HEW system. These components cannot be unregistered when installed (e.g. The HEW builder, debugger and flash support).

5.1 Tool locations

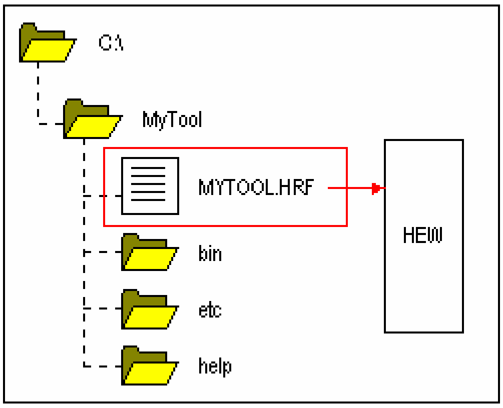
The HEW maintains the locations of HEW compatible components automatically as each new tool is installed. After installation, the HEW stores information about the component (including its location). This is referred to as **Registration**. Although initial registration is automatic, during the course of development or if you want to manage the tools being used in your projects more effectively, you may need to register components yourself.

5.2 HEW registration files

When a HEW compatible component (i.e. toolchain, system tool or utility phase) is installed, part of its installation will include a file with the extension .HRF. This file, called a **HEW Registration File**, describes the component to the HEW (see the figure below).



The process of registration refers to loading a component's .HRF file into the **Tools Administration** dialog box (see the figure below).



In order to use a component with HEW it must first be registered (see section 5.3, Registering a component, for further information). The **Tools Administration** dialog box shows all currently registered components. To access it, ensure no workspaces are open and then select the [**Tools->Administration**] menu option. If you attempt to access the **Tools Administration** dialog box when a workspace is open, the **Tools Administration** dialog box is opened but cannot be modified. When HEW is installed, any new tools are automatically registered. Day to day usage of the HEW though, may mean you need to know more about the tools registration process.

HEW stores tool information in a tool database file, which is stored in the root of the tool installation directory. By default this is set to the HEW application directory, however if you are working in a network environment this directory may be set to another location. It is possible to change the tool directory location and this causes a re-scan of the tools that are registered in HEW.

To change the tools directory location

1. Select the [**Tools->Administration**] menu option.
2. Click the **Modify** button, which is next to the **Current HEW Tools Database Location** field.
3. Browse to the root directory of the new tool location and click the OK button.

4. This will switch the directory and change the tool location to the new directory. It will be necessary to scan for any new tools that may be in this location. This can be achieved by using the **Search Disk** or **Register Tool** functionality.

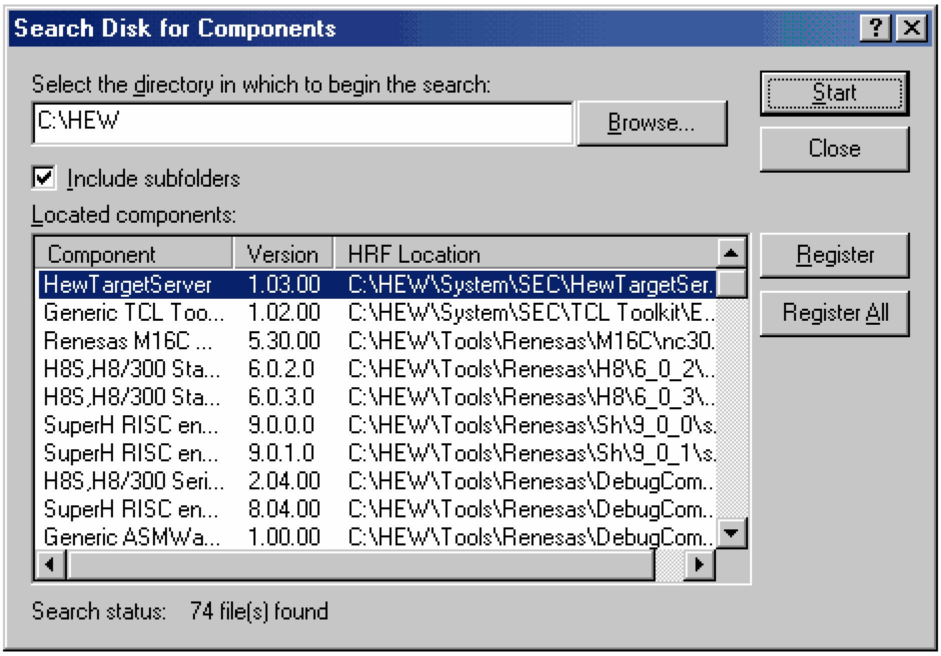
5.3 Registering a component

The HEW will automatically attempt to register any new components installed since the last time it was invoked. However, in some circumstances you may need to register components yourself.

In some cases it is useful to search a drive for HEW-compatible components. This is especially useful if the HEW installation was deleted or corrupted, as it can recreate your tool information instantly.

To search for components and register them

1. Click the **Search Disk** button on the **Tools Administration** dialog box. The **Search Disk for Components** dialog box will be displayed.
2. Enter the directory in which you would like to search into the top field, or browse to it graphically by clicking the **Browse** button.
3. Check the **Include Subfolders** checkbox if you would like to search the directory specified and all directories below it.
4. Click the **Start** button to begin the search. During the search, the **Start** button will change to a **Stop** button. Click the **Stop** button to halt the search at any time.
5. The results of the search are shown in the **Located Components** list. Select a component and click the **Register** button to register an individual component, or click the **Register All** button to register all located components.
6. Click the **Close** button to exit the dialog box.



Search Disk for Components

Select the directory in which to begin the search: C:\HEW

☑ Include subfolders

Located components:

Component	Version	HRF Location
HewTargetServer	1.03.00	C:\HEW\System\SEC\HewTargetSer.
Generic TCL Too...	1.02.00	C:\HEW\System\SEC\TCL Toolkit\E..
Renesas M16C ...	5.30.00	C:\HEW\Tools\Renesas\M16C\nc30.
H8S,H8/300 Sta...	6.0.2.0	C:\HEW\Tools\Renesas\H8\6_0_2\..
H8S,H8/300 Sta...	6.0.3.0	C:\HEW\Tools\Renesas\H8\6_0_3\..
SuperH RISC en...	9.0.0.0	C:\HEW\Tools\Renesas\Sh\9_0_0\s.
SuperH RISC en...	9.0.1.0	C:\HEW\Tools\Renesas\Sh\9_0_1\s.
H8S,H8/300 Seri...	2.04.00	C:\HEW\Tools\Renesas\DebugCom..
SuperH RISC en...	8.04.00	C:\HEW\Tools\Renesas\DebugCom..
Generic ASMWa...	1.00.00	C:\HEW\Tools\Renesas\DebugCom..

Search status: 74 file(s) found

To register a single component

1. Click the **Register** button on the **Tools Administration** dialog box.
2. Browse to the component's .HRF file and click the **OK** button to register that component.

Note:

The HEW registration file is located in the root directory of a component's installation.

5.4 Unregistering a component

The components that are registered with the HEW affect the way it behaves. For example, every compatible system tool that is registered will be added to the **Tools** menu when a new project is created. Sometimes this may not be desirable. If this is not required, open the **Tools Administration** dialog box, select the undesired component from the **Registered Components** list and click the **Unregister** button. A dialog box will be invoked, which asks you to confirm this action. Click **Yes** to unregister the component.

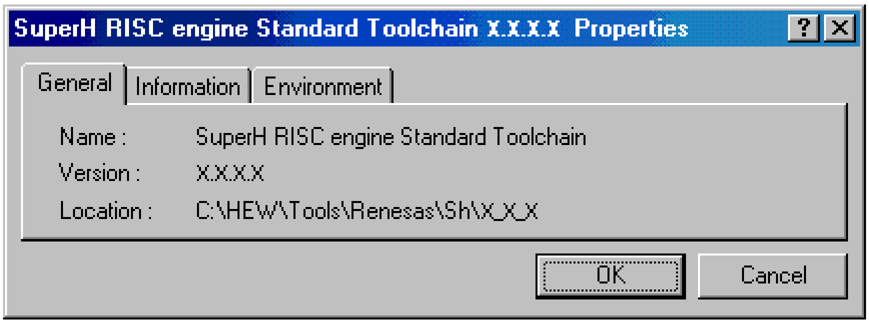
Note:

Unregistering a component does not remove its installation from the hard disk. It simply removes the information that the HEW was storing about that component (i.e. it ‘disconnects’ it from the HEW). The action can be easily reversed at any time by registering the tool manually (see section 5.3, Registering a component). If you want to remove a component from the hard disk (i.e. uninstall a component) then refer to section 5.6, Uninstalling a component.

5.5 Viewing and editing component properties

To view information regarding a component, select it from the **Registered Components** list on the **Tools Administration** dialog box and click the **Properties** button. The **Properties** dialog box will be displayed.

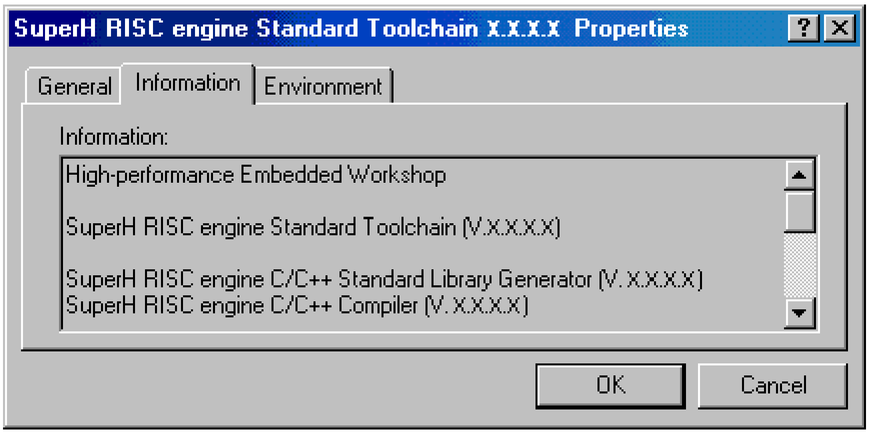
The **General** tab displays the name, version and location of the selected component.



SuperH RISC engine Standard Toolchain X.X.X.X Properties

Name : SuperH RISC engine Standard Toolchain
Version : X.X.X.X
Location : C:\HEW\Tools\Renesas\Sh\X_X_X

Select the **Information** tab to view any information about the component. This may include copyright information, enhancements and so on.



SuperH RISC engine Standard Toolchain X.X.X.X Properties

Information:

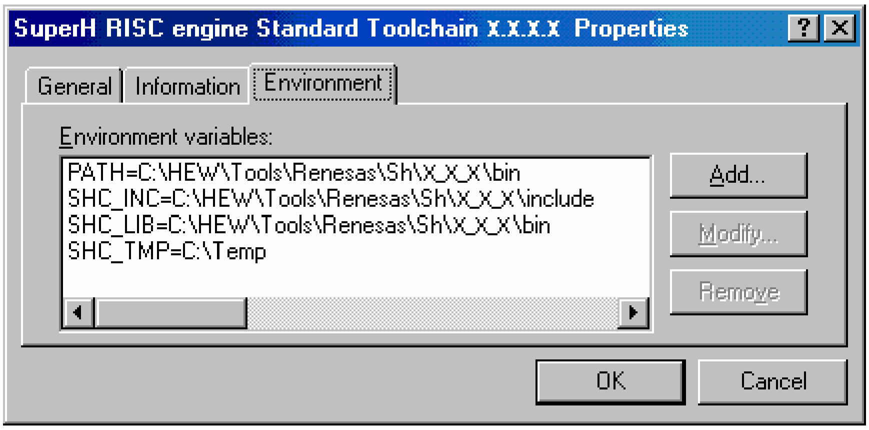
High-performance Embedded Workshop

SuperH RISC engine Standard Toolchain (V.X.X.X.X)

SuperH RISC engine C/C++ Standard Library Generator (V. X.X.X.X)
SuperH RISC engine C/C++ Compiler (V. X.X.X.X)

If there is an issue with the component and it is working incorrectly additional information is displayed here.

Select the **Environment** tab, if it exists, to view and edit a component's environment settings. This tab is most commonly used to modify the environment of a toolchain.



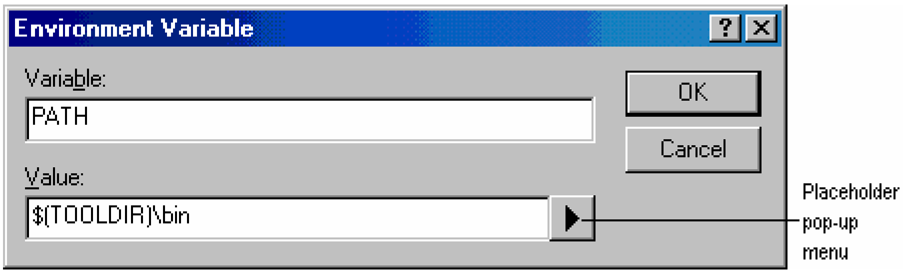
```
PATH=C:\HEW\Tools\Renesas\Sh\X_X_X\bin
SHC_INC=C:\HEW\Tools\Renesas\Sh\X_X_X\include
SHC_LIB=C:\HEW\Tools\Renesas\Sh\X_X_X\bin
SHC_TMP=C:\Temp
```

To add a new environment variable

1. Click the **Add** button. The **Environment Variable** dialog will be invoked.
2. Enter the variable name into the **Variable** field.
3. Enter the variable's value into the **Value** field
4. Click the OK button to add the new variable to the **Environment** tab.

To modify an environment variable

1. Select the variable that you want to modify from the **Environment** tab.
2. Click the **Modify** button. The **Environment Variable** dialog box will be invoked.
3. Make the required changes to the **Variable** and **Value** fields.
4. Click the OK button to modify the environment variable.



Variable: PATH

Value: $(TOOLDIR)\bin

To remove an environment variable

1. Select the variable that you want to remove from the **Environment** tab.
2. Click the **Remove** button.

Note:

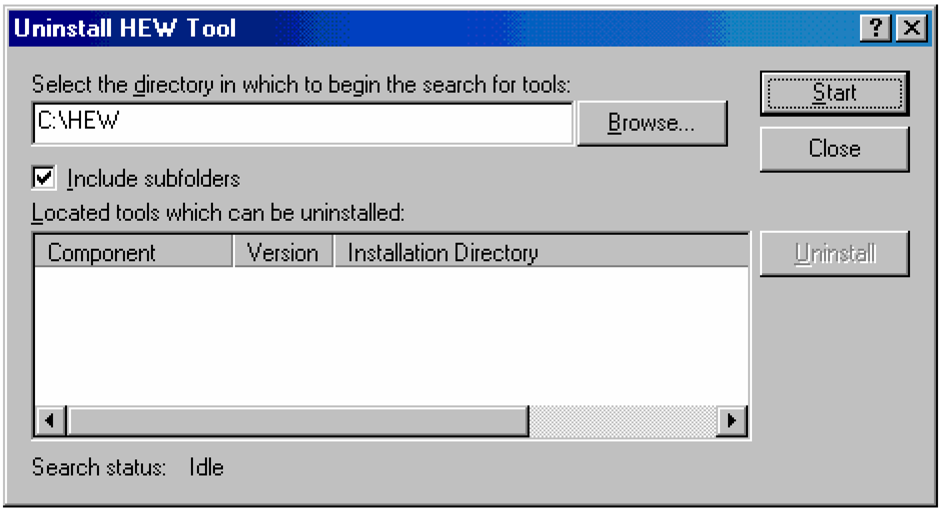
Placeholder pop-up menus are included to ensure that the environment can be specified as flexibly as possible. For further information about using placeholders, see Reference 5, Placeholders.

5.6 Uninstalling a component

The HEW provides a built-in Tools Uninstaller, which can remove unregistered components.

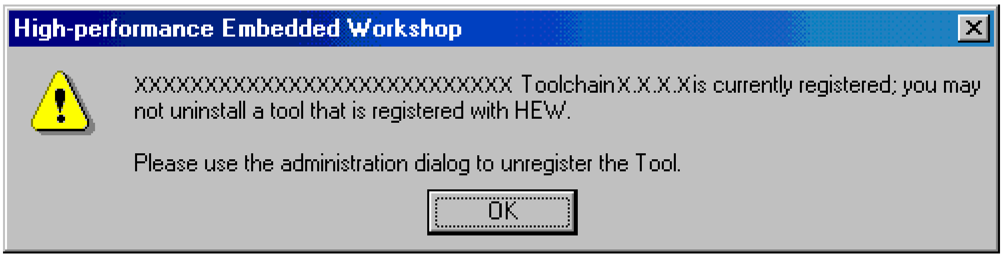
To run the Tools Uninstaller

1. Select the [**Tools->Administration**] menu option.
2. Click the **Uninstaller** button. The **Uninstall HEW Tool** dialog box is displayed.



3. Enter the directory in which you would like to search into the search field, or browse to it graphically by clicking the **Browse** button.
4. Check the **Include subfolders** checkbox if you would like to search the directory specified and all directories below it.
5. Click the **Start** button to begin the search. During the search, the **Start** button will change to a **Stop** button. Click the **Stop** button to halt the search at any time.
6. The results of the search are shown in the **Located tools which can be uninstalled** list. Select a component and click the **Uninstall** button to uninstall that component.
7. Click the **Close** button to exit the dialog box.

A component may only be uninstalled if it is not currently registered with the HEW. If you attempt to uninstall a tool that is registered, you will be asked to uninstall the tool first.



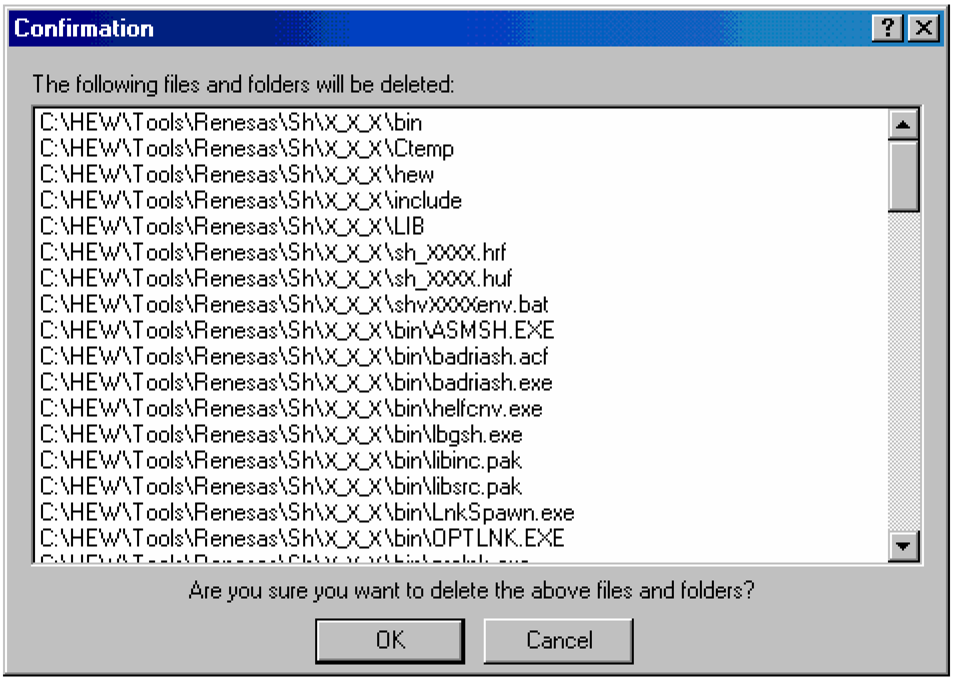
High-performance Embedded Workshop

XXXXXXXXXXXXXXXXXXXXXXXXXXX ToolchainX.X.X.Xis currently registered; you may not uninstall a tool that is registered with HEW.

Please use the administration dialog to unregister the Tool.

You must then return to the **Tools Administration** dialog box (accessed by selecting the [**Tools->Administration**] menu option), unregister the tool and then invoke the **Tool Uninstaller** again.

If the selected tool is not registered with the HEW, the **Confirmation** dialog box will be displayed when the **Uninstall** button is clicked. This dialog displays all of the files and folders that will be deleted. If you are certain that these files and folders can be deleted then click the **OK** button. To abort the uninstallation, click the **Cancel** button.



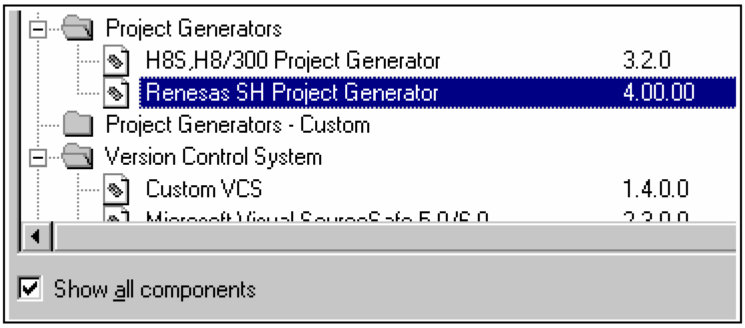
The following files and folders will be deleted:

```
C:\HEW\Tools\Renesas\Sh\X_X_X\bin
C:\HEW\Tools\Renesas\Sh\X_X_X\Ctemp
C:\HEW\Tools\Renesas\Sh\X_X_X\hew
C:\HEW\Tools\Renesas\Sh\X_X_X\include
C:\HEW\Tools\Renesas\Sh\X_X_X\LIB
C:\HEW\Tools\Renesas\Sh\X_X_X\sh_XXXX.hrf
C:\HEW\Tools\Renesas\Sh\X_X_X\sh_XXXX.huf
C:\HEW\Tools\Renesas\Sh\X_X_X\shvXXXXenv.bat
C:\HEW\Tools\Renesas\Sh\X_X_X\bin\ASMSH.EXE
C:\HEW\Tools\Renesas\Sh\X_X_X\bin\badriash.acf
C:\HEW\Tools\Renesas\Sh\X_X_X\bin\badriash.exe
C:\HEW\Tools\Renesas\Sh\X_X_X\bin\helfcnv.exe
C:\HEW\Tools\Renesas\Sh\X_X_X\bin\lbgsh.exe
C:\HEW\Tools\Renesas\Sh\X_X_X\bin\libinc.pak
C:\HEW\Tools\Renesas\Sh\X_X_X\bin\libsrc.pak
C:\HEW\Tools\Renesas\Sh\X_X_X\bin\LnkSpawn.exe
C:\HEW\Tools\Renesas\Sh\X_X_X\bin\OPTLNK.EXE
```

Are you sure you want to delete the above files and folders?

5.7 Technical support

The **Tools Administration** dialog box is capable of displaying information regarding 'hidden' system components. These are part of the HEW itself, and cannot be registered/unregistered manually. If you check the **Show all components** checkbox on the **Tools Administration** dialog, extra component folders are displayed (see the figure below).



Project Generators	
H8S,H8/300 Project Generator	3.2.0
Renesas SH Project Generator	4.00.00
Project Generators - Custom	
Version Control System	
Custom VCS	1.4.0.0

☑ Show all components

When seeking technical support, you may be asked to give details about some or all of these components. To do so, open the respective folder, select a component and click the **Properties** button. The **Properties** dialog box will be invoked.

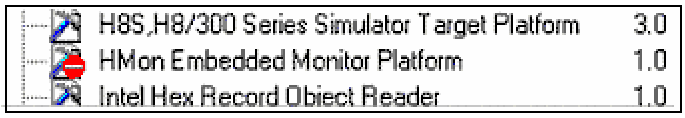
The HEW also has a feature that outputs tool information regarding the registered components to a file. This allows you to retrieve information about the entire HEW system. This information can then be sent to your technical support contact if you are experiencing problems with the HEW.

To output tool information

1. Click the [**Tools->Administration**] menu item.
2. Click the **Tool information** button. A **Save tool information file** dialog box is displayed.
3. Choose the location of the output file and click the **OK** button.
4. A file is created in the chosen location with the current registered tool setup of the HEW.

If any of the components have problems these can be seen in the tools administration dialog. If the icon has an additional icon this explains the problem. There are two additional icons that can be displayed.

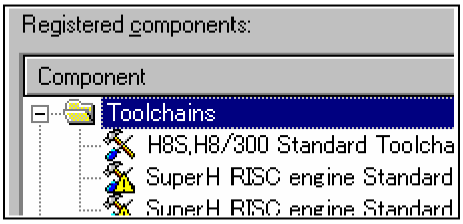
If a component is found but cannot be used due to it being an old version or another dependent component is not available then the icon in figure below is used to show this.



H8S,H8/300 Series Simulator Target Platform 3.0
HMon Embedded Monitor Platform 1.0
Intel Hex Record Object Reader 1.0

Component not found icon

If the component is not located where the registration file says it is then the icon in figure below is used to show this.



Registered components:

Component
Toolchains
H8S,H8/300 Standard Toolcha
SuperH RISC engine Standard
SuperH RISC engine Standard

Incompatible component found icon

Note:

If the tool has one of these errors then it is possible to get more information by the following method:

To get tool error feedback

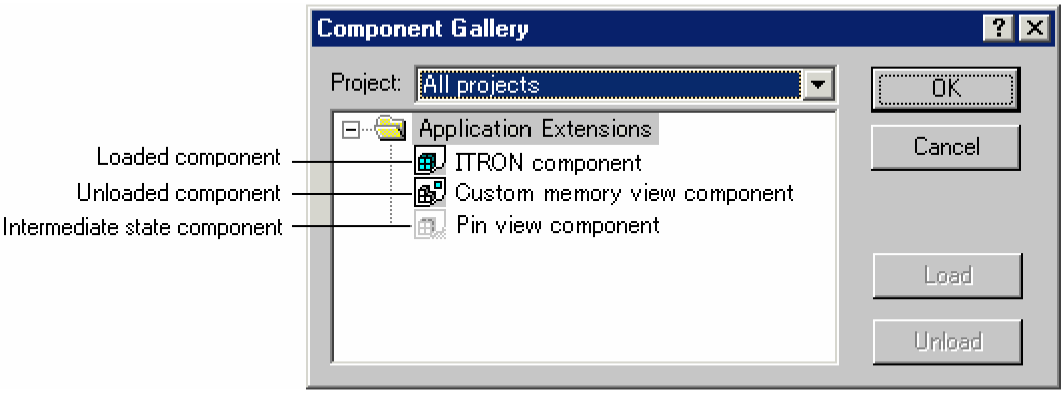
1. Click the [**Tools->Administration**] menu item.
2. Select the tool in the list that has an issue.
3. Click **Properties**.
4. Select the information tab and scroll the edit field to the bottom.
5. The reason for the problem will be displayed in this area.

5.8 Using On-Demand components

The HEW version 3.0 onwards has the concept of on-demand components. These components are not automatically loaded by the application or the debugger component. These components can be loaded by the user or as part of the project generation process.

To load or unload an on-demand component manually

1. Click the [**Project->Components**] menu item.
2. The component gallery dialog box is displayed.
3. Select the component you wish to load. Click the **Load** button. The components image should change to the loaded state.
4. If you wish to unload a component. Select the component. Click the **Unload** button. The components image should change to the unloaded state.
5. Click OK to verify the changes.



Component Gallery

Project: All projects

- Application Extensions
  - ITRON component
  - Custom memory view component
  - Pin view component

Note:

Each project in your workspace can have different components loaded and unloaded. If you have multiple projects you can use the “Multiple projects” and “All projects” items to change a components load status over more than one project. If you select a combination which means the component is loaded in one project and not anther then the intermediate state icon is displayed.

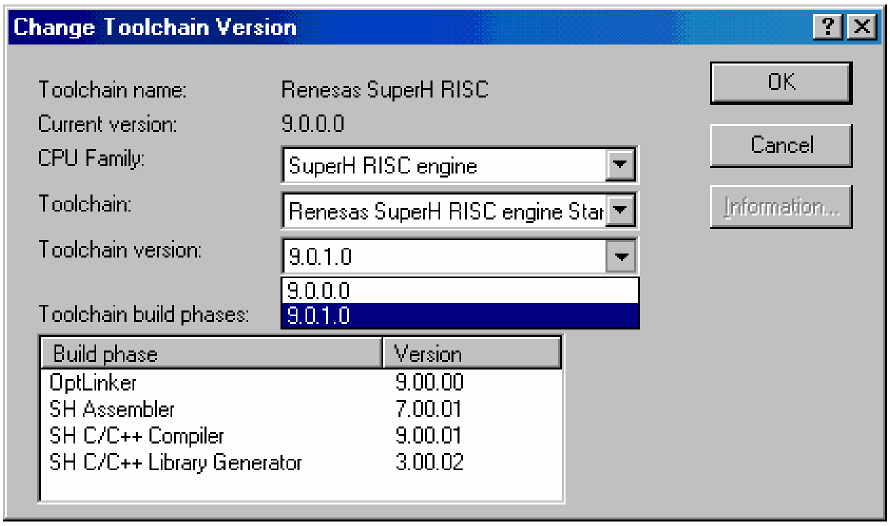
5.9 Custom project types

The [**Project->Create Project Type**] menu item in HEW and allows you to create a template for your project. This menu item takes the settings of the current project and then creates a project type for you. The user can specify the name of the new type and the style of the Project Generation Wizard. Once created, these project types appear in the **Tools Administration** dialog box and are initially hidden in the System Components part of the **Tools Administration** tree.

To export one of the Custom Project Generators, select the **Export** button on the **Tools Administration** dialog box. The export functionality packages the Custom Project Generator into a binary file, which includes an executable. This can be given to another user who then runs the executable. This installs the Project Generator into the correct location on the target user’s machine.

Note:

A project template can be created by selecting [**Project–>Create Project Type**] based on the project in use. This template includes the version information of the toolchain. When a project is created by using this template after the toolchain version has been updated in your HEW system, check that the toolchain version of the created project matches the using environment. When the registered toolchain can be updated, the toolchain version can be changed in the dialog box that is displayed by selecting [**Tools–>Change Toolchain Version**].



Change Toolchain Version

Toolchain name: Renesas SuperH RISC
Current version: 9.0.0.0
CPU Family: SuperH RISC engine
Toolchain: Renesas SuperH RISC engine Star
Toolchain version: 9.0.1.0
9.0.0.0
9.0.1.0

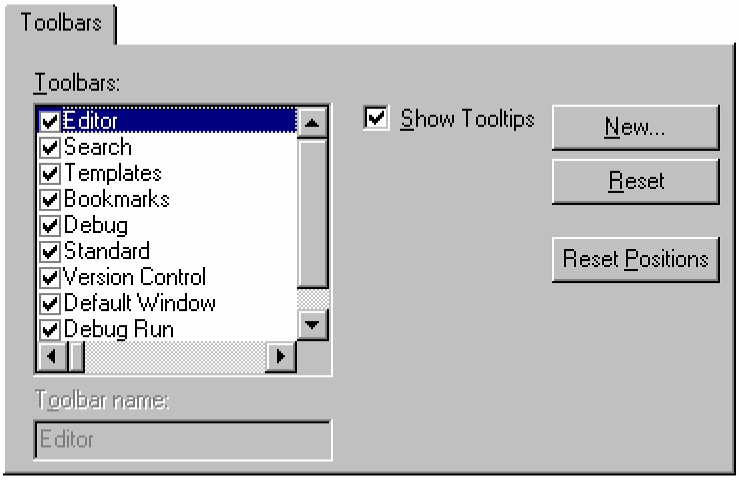
Toolchain build phases:

Build phase	Version
OptLinker	9.00.00
SH Assembler	7.00.01
SH C/C++ Compiler	9.00.01
SH C/C++ Library Generator	3.00.02

6. Customizing the Environment

6.1 Customizing the toolbars

The High-performance Embedded Workshop provides standard toolbars as detailed in the Toolbars topic. In addition to these, you may also construct your own toolbars.


Toolbars

Toolbars:

☑ Editor
☑ Search
☑ Templates
☑ Bookmarks
☑ Debug
☑ Standard
☑ Version Control
☑ Default Window
☑ Debug Run

☑ Show Tooltips

Toolbar name: Editor

To create a new toolbar

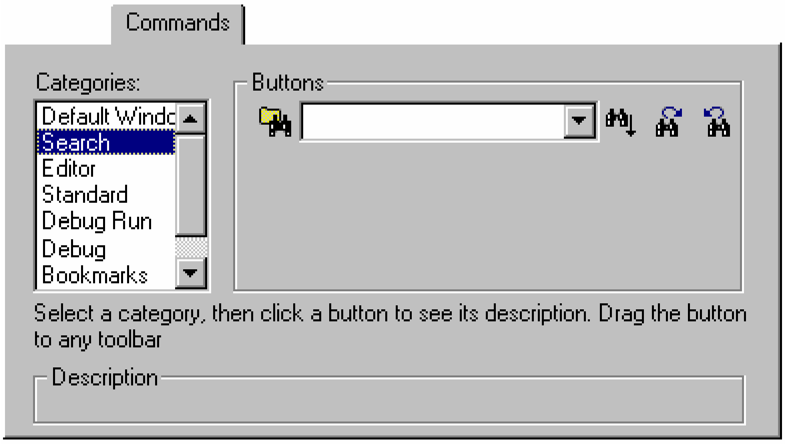
1. Select the [**Setup->Customize**] menu option. The **Setup Customize** dialog box will be displayed.
2. Click the **New** button. The **New Toolbar** dialog box will be displayed.
3. Enter the name of the new toolbar into the **Toolbar Name** field.
4. Click the OK button to create the new toolbar.

Note:

When a new toolbar is created it will appear undocked (i.e. ‘floating’) and empty.

To add buttons to a toolbar

1. Select the [**Setup->Customize**] menu option. The **Setup Customize** dialog box will be displayed. Select the **Commands** tab.
2. Browse the available buttons by selecting the button categories from the **Categories** list. Select a button from the **Buttons** area to display information on its operation.
3. Click and drag a button from the dialog box onto the toolbar.



Select a category, then click a button to see its description. Drag the button to any toolbar

To remove buttons from a toolbar

1. Select the [**Setup->Customize**] menu option. The **Setup Customize** dialog box will be displayed. Select the **Commands** tab.
2. Click and drag a button from the toolbar onto the **Buttons** area.

To modify the name of a user-defined toolbar

1. Select the [**Setup->Customize**] menu option. The **Setup Customize** dialog box will be displayed.
2. In the **Toolbars** list, select the user-defined toolbar and whose name you want to modify.
3. Modify the name of the toolbar in the **Toolbar Name** field.
4. Click the OK button to save the toolbar’s new name.

To remove a user-defined toolbar

1. Select the [**Setup->Customize**] menu option. The **Setup Customize** dialog box will be displayed.
2. Select the user-defined toolbar from the **Toolbars** list and the **Reset** button will change to a **Delete** button. Click the **Delete** button.

To reset a standard toolbar back to its original state

1. Select the [**Setup->Customize**] menu option. The **Setup Customize** dialog box will be displayed.
2. Select the standard toolbar from the **Toolbars** list and then click the **Reset** button.

To reset the toolbar position back to its original state

1. Select the [**Setup->Customize**] menu option. The **Setup Customize** dialog box will be displayed.
2. Click the **Reset Positions** button.

The **Reset Positions** button reverts all toolbars back to their original 'factory' default positions. Be careful as this will reset any custom positions that you may have setup during your session.

To show or hide toolbar tooltips

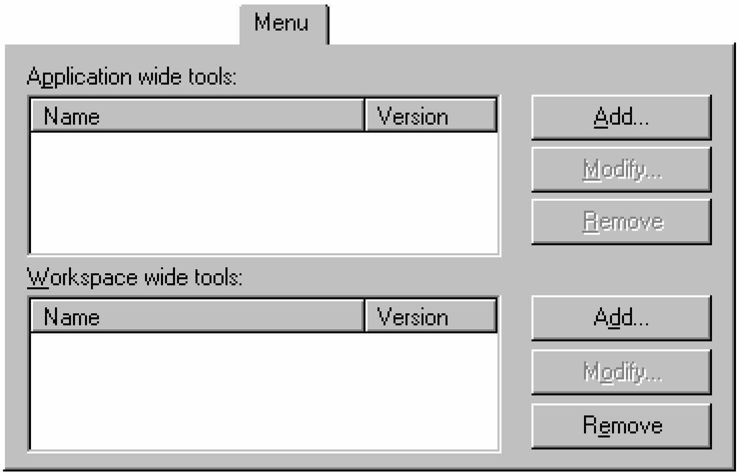
1. Select the [**Setup->Customize**] menu option. The **Setup Customize** dialog box will be displayed.
2. Set the **Show Tooltips** checkbox as desired.

6.2 Customizing the Tools menu

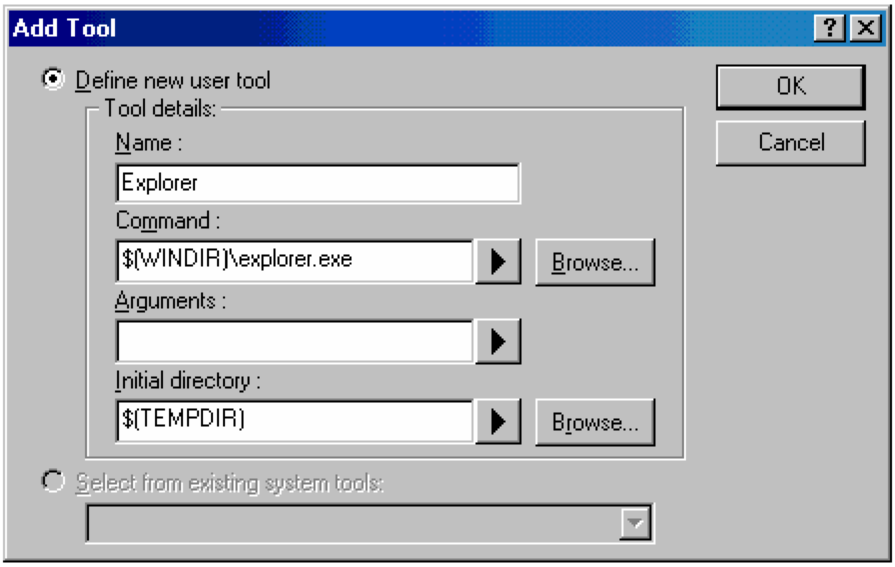
The **Tools** menu can be customized to include your own menu options.

To add a new menu option

1. Select the [**Setup->Customize**] menu option. The **Setup Customize** dialog box will be displayed. Select the **Menu** tab. The first thing you need to decide is whether you are adding a global application-wide tool (which will be available to all of your workspaces), or whether you wish to add a workspace-wide tool (which is only valid for the current workspace). Once you have chosen, choose the relevant section of the dialog.



2. Click the **Add** button. The **Add Tool** dialog box will be invoked. If you would like to add an existing system tool to the menu then select the **Select from existing tools** radio button, choose the tool from the drop-down list and then click the OK button. Alternatively, if you would like to add a tool of your own then follow the remaining steps.
3. Enter the name of the tool into the **Name** field.
4. Enter the command, excluding arguments, into the **Command** field.
5. Enter any arguments that you would like to pass to the command into the **Arguments** field.
6. Enter the initial directory, in which you would like the tool to run, into the **Initial Directory** field.
7. Click the OK button to add the menu option to the **Setup** menu.



● Define new user tool

Name : Explorer

Command : $(WINDIR)\explorer.exe

Arguments :

Initial directory : $(TEMPDIR)

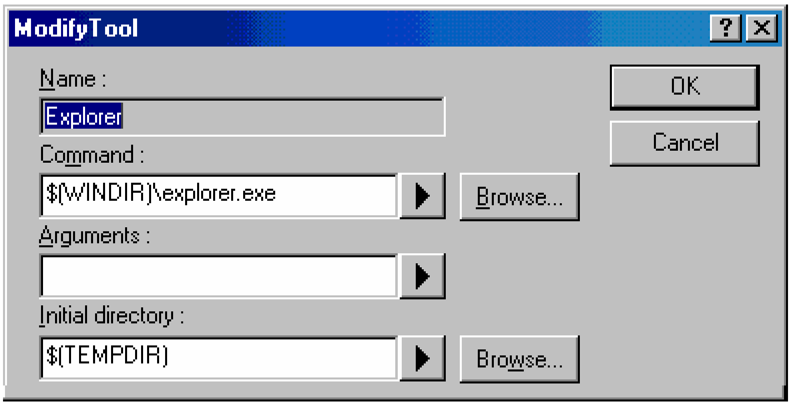
○ Select from existing system tools:

Note:

New menu options are added to the bottom of the list (i.e. bottom of the **Tools** menu) by default. However, the order of menu options in the **Setup** menu can be modified.

To modify a menu option

1. Select the [**Setup->Customize**] menu option. The **Customize** dialog box will be displayed. Select the **Menu** tab.
2. Select the menu option that you would like to modify and then click the **Modify** button.
3. Make the desired changes on the **Modify Tool** dialog box and then click the OK button.



ModifyTool

Name : Explorer

Command : $(WINDIR)\explorer.exe

Arguments :

Initial directory : $(TEMPDIR)

To remove a menu option

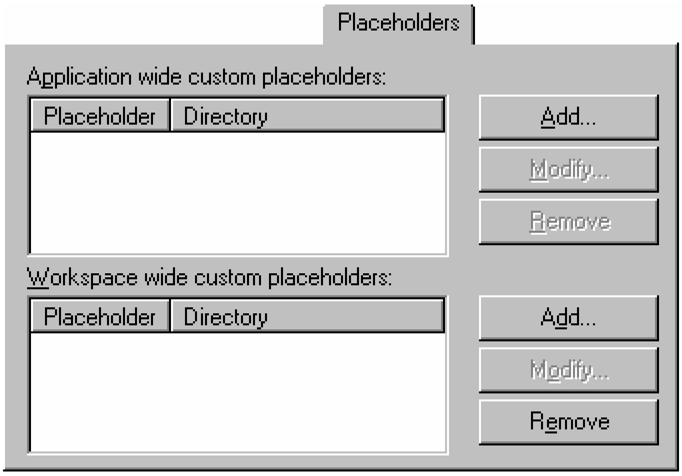
1. Select the [**Setup->Customize**] menu option. The **Customize** dialog box will be displayed. Select the **Menu** tab.
2. Select the menu option that you would like to remove and then click the **Remove** button.

6.3 Using custom placeholders

Throughout the High-performance Embedded Workshop the user can use a number of pre-defined placeholders for directory definitions. For example, the user can use the “$(PROJDIR)” variable to signify the current HEW project directory. This makes it much easier to relocate projects and keep all of the paths correct.

The High-performance Embedded Workshop also has the ability to define custom placeholders. This means you can enter your own custom placeholder definition and decide upon its directory value. Once defined this placeholder becomes available throughout the rest of the HEW system.

The placeholders can be defined on an application-wide level so the placeholders are available to all workspaces and projects that use the HEW. The other method of defining the placeholders is using the workspace-wide custom placeholders. This means the placeholders can only be used in the current workspace. This list is only available when you have a workspace open.

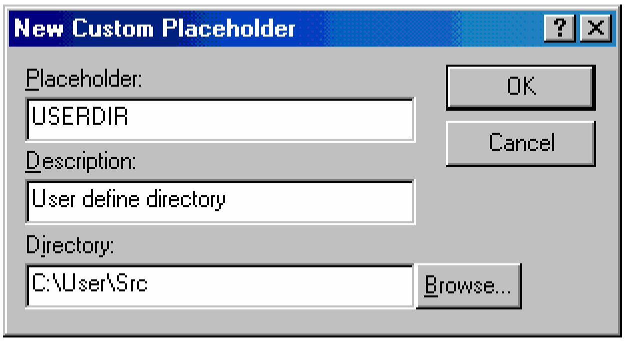


To add a custom placeholder

1. Choose the [**Setup->Customize**]. The **Tools Customize** dialog box will be displayed. Select the **Placeholders** tab.
2. Choose whether you need to use an application-wide or workspace-wide placeholder. Click the **Add** button which is adjacent to the list you require.
3. The **Add New Placeholder** dialog box is displayed.
4. In the fields provided choose a suitable name for the placeholder and a description of what the placeholder means.
5. Choose a directory, which relates to this placeholder. It is possible to use placeholders that are already defined in this field, such as $(PROJDIR).

Note:

In HEW V.2.01 or later version, the user-defined directory can be set as the custom placeholder, which can be used for setting the toolchain option. When the directory path is specified, specify an absolute path in ‘Directory’, as shown in figure below.



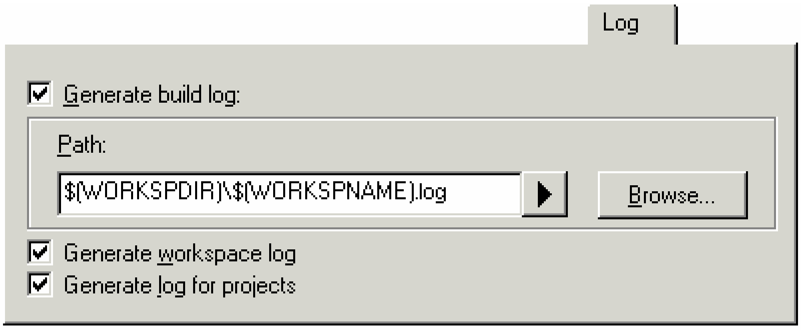
Placeholder: USERDIR

Description: User define directory

Directory: C:\User\Src

6.4 Using the workspace and project log facilities

The HEW has workspace and project logging facilities integrated into the application. These facilities can be switched on via the log tab on the Tools customize dialog. This option is especially useful when the network database is in operation. This is because user names and changes are logged to this file.



☑ Generate build log:

Path: $(WORKSPDIR)\$(WORKSPNAME).log

☑ Generate workspace log

☑ Generate log for projects

When the **Generate workspace log** is clicked any workspace changes will be logged to a file with the same name as the workspace with a ".log" extension. This file will be located in the same directory as the workspace file.

When the **Generate log for projects** is clicked any projects in the current workspace that have changes made to them will be logged to a file with the same name as the project with a ".log" extension. This file will be located in the same directory as the project file.

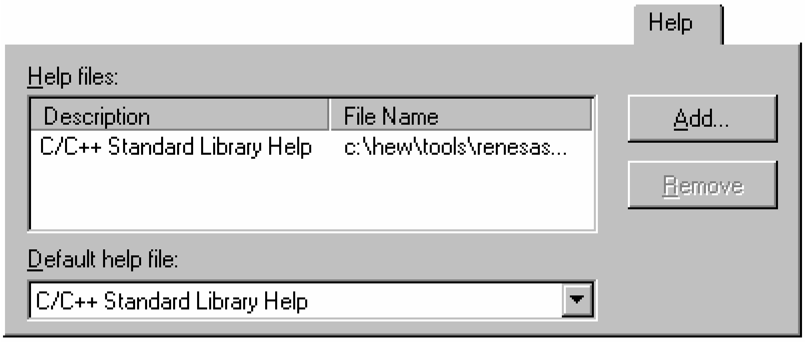
The log file is updated when the workspace is saved.

6.5 Configuring the help system

The High-performance Embedded Workshop provides context sensitive help within the **Editor** window. In other words, if you select some text in the **Editor** window and then press F1, the High-performance Embedded Workshop will attempt to locate help on that selected item. The help files that will be searched are listed in the **Help** tab of the **Setup Customize** dialog box.

To add a new help file

1. Select the [**Setup->Customize**] menu option. The **Setup Customize** dialog box will be displayed. Select the **Help** tab.

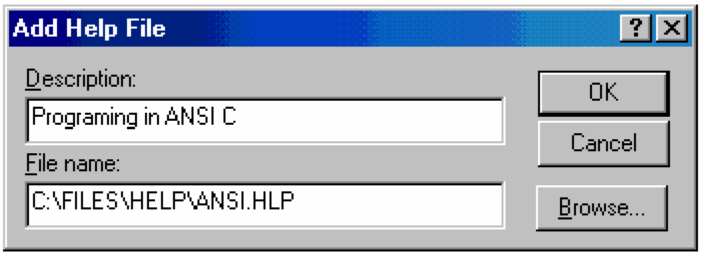


Help files:

Description	File Name
C/C++ Standard Library Help	c:\hew\tools\renesas...

Default help file: C/C++ Standard Library Help

2. Click the **Add** button. The **Add Help File** dialog box will be displayed.



Description: Programing in ANSI C

File name: C:\FILES\HELP\ANSI.HLP

3. Enter a description of the help file into the **Description** field.
4. Enter the full path of the help file into the **File name** field (or browse to it graphically by clicking the **Browse** button).
5. Click the **OK** button to add the new help file to the list.

To make a help file the default choice, select it from the **Default Help File** drop-down list or set it to **None** if you would like to be prompted for a help file whenever F1 is pressed.

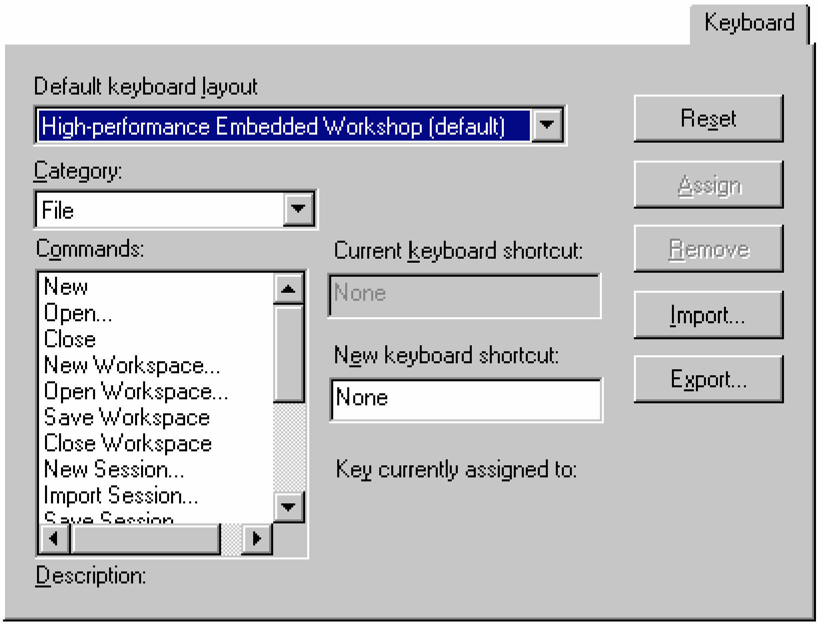
To remove a help file

1. Select the [**Setup->Customize**] menu option. The **Setup Customize** dialog box will be displayed. Select the **Help** tab.
2. Select the help file to be removed and then click **Remove**.
3. Click the **OK** button to confirm the new help file settings.

6.6 Keyboard shortcut customization

The HEW allows the keyboard shortcuts to be customized to your own preferences. This means that major operations can be configured to different keys especially useful if you are migrating from a different tool.

To reach the keyboard shortcut customization dialog click the [**Setup->Customize**] menu item. Then when **Customize** dialog box is invoked click the **Keyboard** tab.



Default keyboard layout: High-performance Embedded Workshop (default)

Category: File

Commands:

New
Open...
Close
New Workspace...
Open Workspace...
Save Workspace
Close Workspace
New Session...
Import Session...
Save Session

Current keyboard shortcut: None

New keyboard shortcut: None

Key currently assigned to:

Description:

This dialog box allows instant selection of either the default HEW keyboard shortcuts or the PD debugger shortcuts. To change the entire keyboard layout select an item in the **Default keyboard layout** drop-down list. By default it uses the **High-performance Embedded Workshop** settings.

A number of operations are possible on this dialog box:

To add a new keyboard shortcut

1. Select the main menu category of the command you wish to modify. It is only possible to modify the commands that have a menu. Only some cases are special that allow modification, these are named global.
2. Select the command you wish to modify or add a new keyboard shortcut for in the **Commands** list.
3. The current shortcut is displayed in the **Current keyboard shortcut** field.

4. Enter the new shortcut in the **New keyboard shortcut** field. Various combinations of buttons can be used here. For example "CTRL+J" or "CTRL+SHIFT+O", etc. If the chosen shortcut is already in use it is displayed below the New keyboard shortcut field.
5. If you are happy with your selection click the **Assign** button.
6. Changes are not saved until the OK button is clicked on the **Customize** dialog box.

To remove a keyboard shortcut

1. Select the main menu category of the command you wish to modify. It is only possible to modify the commands that have a menu. Only some cases are special that allow modification, these are named global.
2. Select the command you wish to modify or add a new keyboard shortcut for in the **Commands** list.
3. The current shortcut is displayed in the **Current keyboard shortcut** field.
4. Click the **Remove** button.
5. Changes are not saved until the OK button is clicked on the **Customize** dialog box.

To reset all the keyboard shortcuts

1. Click the **Reset** button. All shortcuts revert to the default settings for the currently selected keyboard layout.
2. Changes are not saved until the OK button is clicked on the **Customize** dialog box.

The keyboard shortcuts dialog box allows you to import and export keyboard settings to a defined file. This allows you to easily transfer settings from one machine to another.

To export keyboard shortcuts

1. Click the **Export** button.
2. A standard file dialog is displayed. Choose the filename to save the settings of the currently selected keyboard layout to.
3. Click OK.

To import keyboard shortcuts

1. Select the keyboard layout you wish to replace with your imported settings in the **Default keyboard layout** drop-down list.
2. Click the **Import** button.
3. A standard file dialog box is displayed. Choose the filename to load the keyboard layout from.
4. Click OK.

6.7 Scope of a control in the setup

6.7.1 Scope of a control in the Customize dialog box

The scope of each control in the **Customize** dialog box, which is launched via [**Setup->Customize**], differs. This can be confusing so these have been listed below:

Tab	Control	Scope
Toolbar	All	Each workspace
Command	All	The whole system
Menu	Application wide tools	The whole system
	Workspace wide tools	Each workspace
Placeholders	Application wide custom placeholder	The whole system

	Workspace wide placeholder	Each workspace
Debugger	Debugger tool	Each project
	Debugger location	Default: whole system and each project basis
	Command line options	Each project
	Session file	Each project
	Download module	Each project
Log	All	Each workspace
Help	All	Each workspace
Keyboard	All	The whole system

6.7.2 Scope of a control in the Options dialog box

Scope of each control of each tab of the **Options** dialog box, which is launched via [**Setup->Options**], affects the whole system.

6.8 Specifying workspace options

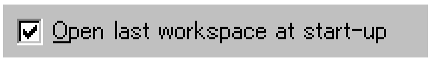
The High-performance Embedded Workshop allows you to control several aspects of a workspace via the **Options** dialog box. To invoke it select the [**Setup->Options**] menu option, and select the **Workspace** tab.

6.8.1 Open last workspace at start-up

When you exit the High-performance Embedded Workshop, the last workspace you had open is stored. On subsequently launching the HEW, you may want the last workspace to be opened automatically.

To open the last workspace at start-up

1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Workspace** tab.
3. Select the **Open last workspace at start-up** checkbox if you would like the HEW to automatically open the last workspace when it is launched.
4. Click the OK button.



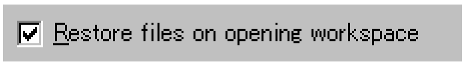
☑ Open last workspace at start-up

6.8.2 Restore files on opening a workspace

When you close a workspace, the HEW stores the names of the files that were open at that time. When you open a workspace, the HEW can restore (i.e. open) the same files so that you can continue your session in exactly the same state as when you left it. If you would like the files associated with a workspace to be opened when the workspace is opened, then set this checkbox.

To restore files on opening a workspace

1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Workspace** tab.
3. Select the **Restore files on opening workspace** checkbox if you would like the files associated with a workspace to be opened when the workspace is opened.
4. Click the OK button.



☑ Restore files on opening workspace

6.8.3 Display workspace information on opening a workspace

When many workspaces are being used, it is sometimes difficult to remember exactly what is contained within each workspace. The High-performance Embedded Workshop allows you to enter a textual description of each workspace. This description can be displayed whenever a workspace is opened.

To enter a workspace description

1. Select the workspace icon from the Projects tab of the workspace window.
2. Right-click to invoke the pop-up menu and then select the **Properties** option. The **Workspace Properties** dialog box will be displayed.
3. Enter the description into the **Information** field.
4. Select the **Show workspace information on workspace open** checkbox if you want a **Workspace Properties** dialog box to be launched on opening a workspace. This checkbox has the same role as the **Display workspace information dialog on opening workspace** checkbox on the **Workspace** tab of the **Options** dialog box.
5. Click the OK button.

☑ Display workspace information dialog on opening workspace

6.8.4 Save workspace before executing any phases

It is possible to force the High-performance Embedded Workshop into saving the current workspace before executing any build phases (i.e. build, build all or build file operations) or version control commands.

To save the workspace before executing any phases

1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Workspace** tab.
3. Select the **Save workspace before executing any tools** checkbox.
4. Click the OK button.

☑ Save workspace before executing any tools

6.8.5 Prompt before saving a workspace

If you are using the **Save workspace before executing any phases** function, you may want the HEW to prompt you before saving the workspace. For further information about saving a workspace before executing phases, see section 6.8.4, Save workspace before executing any phases.

To display a prompt before saving the workspace

1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Workspace** tab.
3. Select the **Prompt before saving workspace** checkbox.
4. Click the OK button.

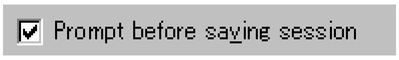
☑ Prompt before saving workspace

6.8.6 Prompt before saving session

Checking this option will force the High-performance Embedded Workshop into displaying a prompt before the session is saved to disk.

To display a prompt before saving the session

1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Workspace** tab.
3. Select the **Prompt before saving session** checkbox.
4. Click the OK button.



☑ Prompt before saving session

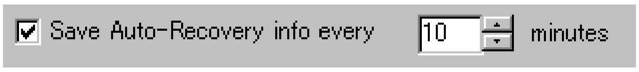
6.8.7 Enable auto-backup facilities

The HEW supports the facility to backup the workspace, project and session files at regular intervals. This means that if your application or development system should fail you will not lose so much work. Any changes you have made will be saved to temporary files.

When re-opening the workspace you will be prompted and asked if you wish to auto-recover the files that were not saved during your last session.

To enable auto-file backup facilities

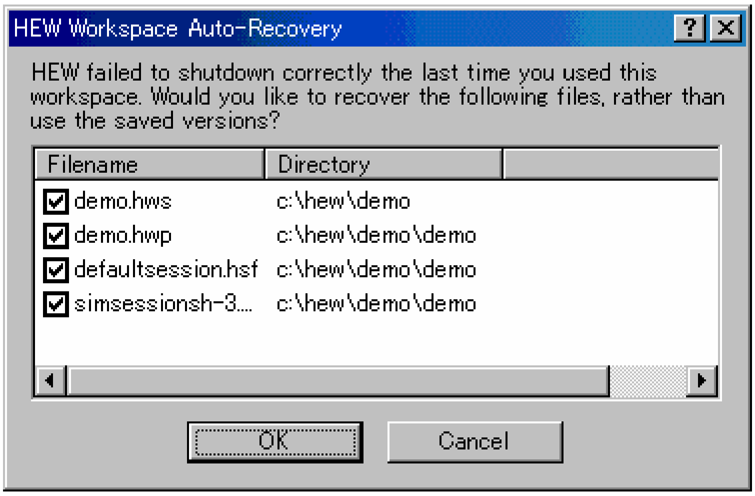
1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Workspace** tab.
3. Select the **Save Auto-Recovery info** checkbox as necessary.
4. Select the number of minutes you wish the auto-backup facility to be launched.
5. Click the OK button.



☑ Save Auto-Recovery info every 10 minutes

Restoring your files

If you open your workspace and the following dialog is displayed it means that the last time the workspace was used problems were encountered.



HEW Workspace Auto-Recovery

HEW failed to shutdown correctly the last time you used this workspace. Would you like to recover the following files, rather than use the saved versions?

Filename	Directory	
☑ demo.hws	c:\hew\demo	
☑ demo.hwp	c:\hew\demo\demo	
☑ defaultsession.hsf	c:\hew\demo\demo	
☑ simsessionsh-3....	c:\hew\demo\demo	

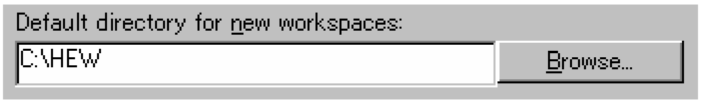
To recover the files check the checkbox alongside the filenames you wish to recover and click OK. Clicking Cancel will discard the auto-recovery files and load from the original files.

6.8.8 Change the default directory for new workspaces

When a new workspace is created, the High-performance Embedded Workshop invokes the **New Project Workspace** dialog box. One of the fields on this dialog is the directory in which the new workspace will be created. By default, this is the root directory. However, it is also possible to set this default directory to another location (e.g. ‘C:\Workspaces’).

To change the default directory for new workspaces

1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Workspace** tab.
3. Enter the directory in which to create new workspaces into the **Default directory for new workspaces** field, or browse to it graphically by clicking the **Browse** button.
4. Click the OK button.



Default directory for new workspaces: C:\HEW

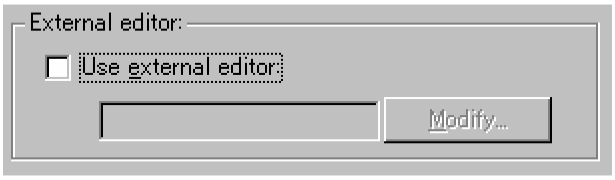
6.9 Using an external editor

The High-performance Embedded Workshop allows you to use an external editor. Once an external editor has been specified, it will be launched when the following actions are performed:

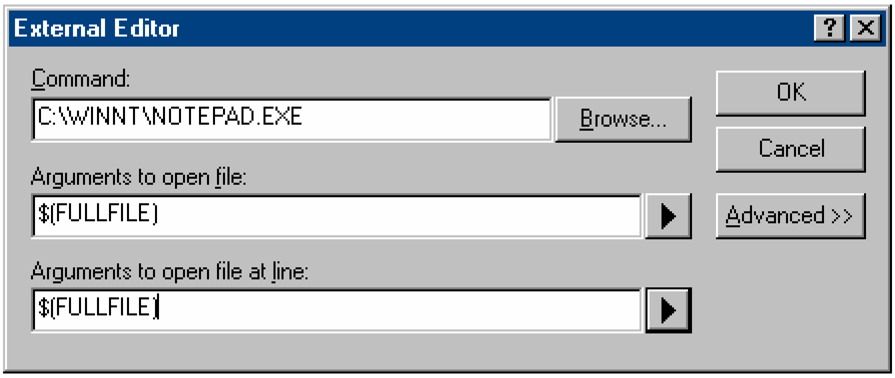
- Selecting a file in the **Open File** dialog box opened by selecting [**File->Open**] menu.
- Selecting the [**File->Recent Files**] menu.
- Double-clicking on a file in the **Projects** tab of the workspace window.
- Double-clicking on an entry in the **Navigation** tab of the workspace window.
- Double-clicking on an error/warning in the **Build** tab of the output window.
- Double-clicking on an entry in the **Find in Files** tab of the output window.
- Right-clicking on a file within the Projects tab of the workspace window and selecting the **Open <file> in external editor** option from the pop-up menu.

- Clicking the **Edit Code** button in the **Breakpoints** dialog box opened by selecting [**Edit->Source Breakpoints**] menu.
- Drag and drop into the HEW window:
  When a file is dragged from the Windows® Explorer and dropped into the HEW window (note that, however, a drag and drop of a file onto the current project and user folder into the **Projects** tab of the workspace window only adds this file to the project and does not open the file.)

To specify an external editor



1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Editor** tab.
3. Check the **Use external editor** checkbox. The **External Editor** dialog box will be displayed.



Command: C:\WINNT\NOTEPAD.EXE

Arguments to open file: $(FULLFILE)

Arguments to open file at line: $(FULLFILE)

4. Enter the path of the executable (without any arguments) into the **Command** field.
5. Enter the arguments required to open a file into the **Arguments to open file** field. Use the $(FULLFILE) placeholder to represent the path of the file to be opened.
6. Enter the arguments required to open a file at a specific line into **Arguments to open file at line** field. Use the $(FULLFILE) placeholder to represent the path of the file to be opened and the $(LINE) placeholder to represent the line number at which the cursor should be initially positioned.
7. Clicking the **Advanced>>** button allows you to select the external or internal (HEW) editor for use when you attempt to open a file by any of the following operations.

- File open menu

When a file is selected in the **Open File** dialog box.

By default, the internal editor is selected.

- Workspace window double click

When a file in the Projects tab or an entry in the Navigation tab is double-clicked in the workspace window

By default, the external editor is selected.

- Output window double click

When an error or warning in the Build tab or an entry in the Find in Files tab is double-clicked in the output window

By default, the external editor is selected.

- Drag and drop file into HEW main window

  When a file is dragged from the Windows® Explorer and dropped into the HEW window (note that, however, a drag and drop of a file onto the current project and user folder into the **Projects** tab of the workspace window only adds this file to the project and does not open the file.)

  By default, the internal editor is selected.

8. Click the OK button.

When the **Use external editor** checkbox is selected, the following actions always select the external editor for use.

- Clicking the **Edit Code** button in the **Breakpoints** dialog box opened by selecting [**Edit-> Source Breakpoints**] menu.
- Selecting the [**File->Recent Files**] menu.

If you select a file in the Projects tab of the workspace window and right-click to display a pop-up menu, the **Open <file name> in external editor** menu is available under the **Open <file name>** menu.

Note:

When using an external editor be aware of the following issues:

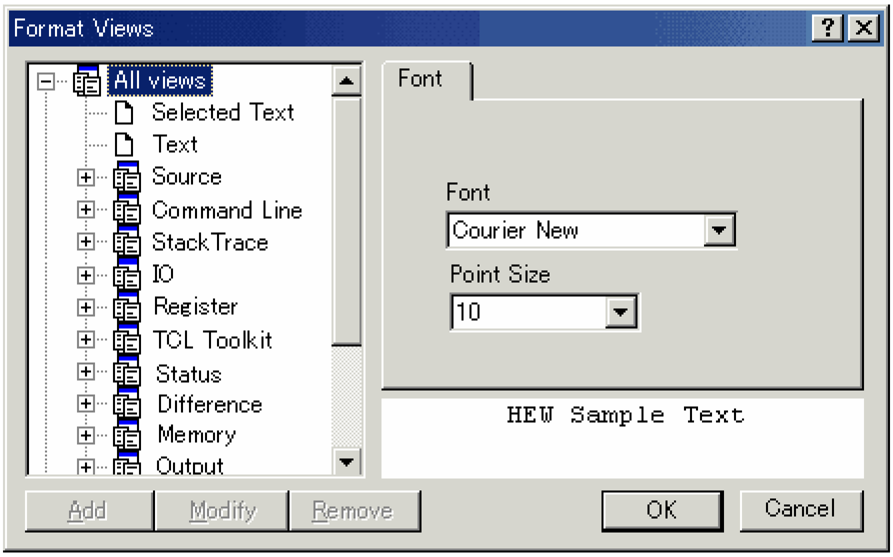
- Each time you invoke the external editor, in whichever way, a separate instance of the editor will be launched.
- You must save your own files before you perform a Build, Build All or Build File operation.

6.10 Customizing the font in your views

The HEW contains many components which you may wish to make look differently. It is possible to change the font and text coloring for the views.

To change the look of your windows

1. Click the [**Setup->Format Views**] menu item.
2. Select the view you are interested in changing the appearance of. To change all views select the "All views" category.
3. Expand the item in the tree to see all items you can change the look of.
4. Select the item. Notice the tab changes on the right of this dialog box.
5. Change the font or text color.
6. Click OK to save the changes and the views will be automatically updated with the new coloring.



Font: Courier New

Point Size: 10

The **Format Views** dialog box allows you to customize the color and font of text shown in the windows.

Item	Category	Objects to be Customized
Source	Selected Text	A selected text shown in the editor window and disassembly window (in source mode)
	PC Line Highlight	The line at the PC (program counter) in the editor window and disassembly window (in source mode) while the **Enable line highlight for PC position** checkbox is selected in the Editor tab of the Options dialog box
	Other keyword groups based on the syntax of respective file types	Comment or keywords***1** shown in the editor window and disassembly window (in source mode) ***1**. Control, Declaration, Operator, PreProcessor, Value, Directive
Difference	LHS Different Lines	Unmatched text lines shown in the left pane in the Difference window
	LHS Moved Lines	Moved text lines shown in the left pane in the Difference window
	RHS Different Lines	Unmatched text lines shown in the right pane in the Difference window
	RHS Moved lines	Moved text lines shown in the right pane in the Difference window
Register	Modified	Text sections of changed values shown in the Register window
Memory	Accessed	Text sections of executed codes at coverage measurement shown in the Memory window
	Not Accessed	Text sections of codes not executed at coverage measurement shown in the Memory window
	Unknown	Text outside the coverage range shown in the Memory window
	No Memory	Text outside the memory range shown in the Memory window
	Modified	Text sections of changed values shown in the Memory window
Output	Build Error	Lines of error messages shown in the Build tab of the Output window
	Build Warning	Lines of warning messages shown in the Build tab of the Output window
	Information	Lines of information messages shown in the Build tab of the Output window
Disassembly	PC Line Highlight	The line at the PC (program counter) in the editor window and disassembly window (in mixed mode or disassembly mode) or Disassembly window while the **Enable line highlight for PC position** checkbox is selected in the Editor tab of the Options dialog box
	Source Lines	Source codes shown in the editor window and disassembly window (in mixed mode)
All	Text	Text shown in the windows

6.11 Using the virtual desktop

HEW has implemented the concept of the virtual desktop. This allows window configurations to be defined that can be switched with the click of a button. When a particular button is clicked the windows are hidden or displayed depending on the current settings of that window configuration.

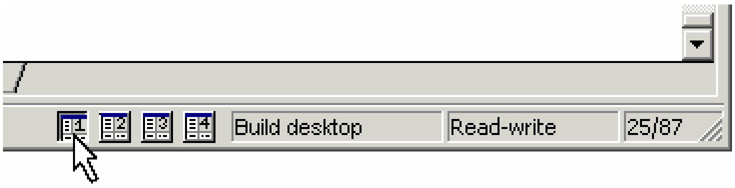
It is possible to have a maximum of 4 desktop configurations in use. When the session is saved the window positions for each configuration are saved to the session file. You can then switch simply between each configuration to gain access to the other windows. The toolbars and windows are dependent on the virtual desktop configuration. Source files are independent of the virtual desktop system and will remain in view.

To rename your configuration to a more meaningful name

1. Click the [**Window->Virtual Desktop**] menu item. Select is cascaded menu.
2. Select the **Desktop Manager** dialog box.
3. Select the window configuration you wish to change the name for.
4. Click rename. Enter the new meaningful name in the edit field and click OK.
5. Click OK to keep the changes and revert to the HEW main window.

To switch desktop configurations

There are a number of ways to switch desktop configuration. The first and easiest method is using the virtual desktop buttons located on the status bar. These are shown below.



In this example the selected desktop is number 1. This has been given the name "Build" by the user. Its description is seen in the edit box to the right on the buttons. Clicking a different desktop selects that button and changes the description control. Once clicked HEW then loads the windows in the new configurations style.

Another method of changing the desktop configuration is as follows:

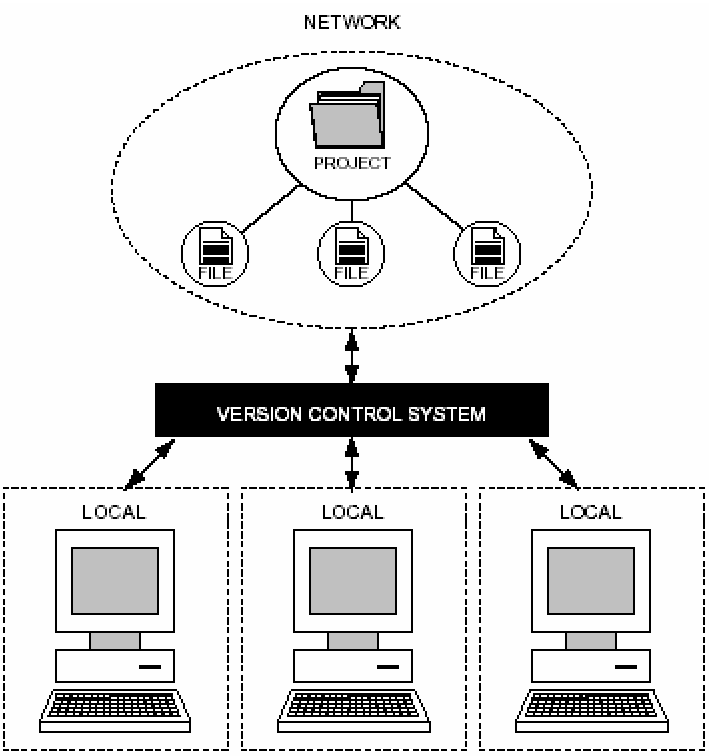
1. Click the [**Window->Virtual Desktop**] menu item. Select is cascaded menu.
2. Then select the desktop configuration you wish to view on this menu. The selected item is ticked.
3. Select the menu item and the setup is altered automatically.

7. Version Control

The High-performance Embedded Workshop (HEW) provides facilities for connecting to version control system. Some of the reasons why version control systems are used with a project are:

- To maintain the integrity of a project.
- To store each stage of a project.
- To enable different users to co-develop a project by controlling revisions to its source files.

Figure below illustrates a typical project where a version control system is in use. This shows three users who all use the same-shared network drive to exchange source code. The version control system provides access and updates to the source files.



- **Files Supported for Version Control by the HEW**

Version control by the HEW basically supports the following types of files.

File type	Extension Group	File Extension	Supported CPU Type
Workspace file	HEW Workspace	hws	Common
Project file	HEW Projects	hwp	Common
File for version control	Configuration file	hvc	Common
Source file	C source file	c	Common
	C++ source file	cpp	SuperH RISC engine, H8S, H8SX, H8/300
	Assembly source file	a30	M32C, M16C
		asm	SuperH RISC engine, H8S, H8SX, H8/300
		mar	
		src	
Include file	C header file	h	Common
	C++ header file	hpp	SuperH RISC engine, H8S, H8SX, H8/300
	Assembly include file	inc	Common

Other types of files can also be supported once they are added to the project.

- **Version Control System that can be Connected with the HEW**

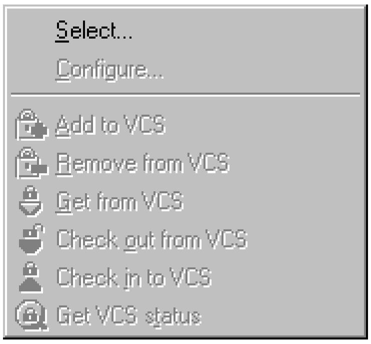
HEW does not provide the version control system itself. It must be installed in your PC in advance. The HEW can be connected to the version control system via the GUI interface.

First, select a version control system, which you wish to connect with the HEW. Either of the followings can be selected:

Type	Version Control System Name Displayed on Screen	Description
Version control by custom settings	**Custom VCS**	Any version control system in which command-line commands should be executed via the MS-DOS command prompt (such as CVS, RCS, or PVCS). The commands in the version control system must be defined as being associated with the GUI interface of the HEW.
Version control by Microsoft(R) Visual SourceSafe 5.0 or 6.0	**Microsoft Visual SourceSafe 5.0/6.0**	Microsoft(R) Visual SourceSafe 5.0 or 6.0. Main features are already defined.

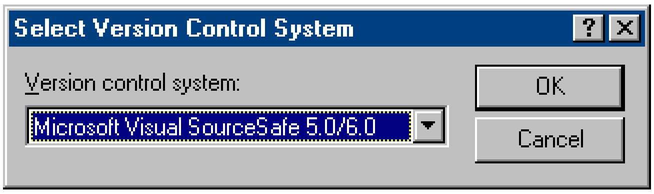
7.1 Selecting a Version Control System

The [**Tools->Version Control**] sub-menu contains eight menu options but only the [**Tools->Version Control->Select**] option is initially available. This is because a version control system is not yet active for the current workspace.



To select a version control system

1. Select the [**Tools->Version Control->Select**] menu option. The **Select Version Control System** dialog box will be displayed, which lists all of the supported version control systems.



Select Version Control System

Version control system: Microsoft Visual SourceSafe 5.0/6.0

2. Select the desired version control system from the **Version control system** list.
3. Click the OK button.

- Once the "Custom VCS" is selected as a version control system, the [**Version Control->Configure**] menu option will become available. See Chapter 8, Using the Custom Version Control System.
- Once "Microsoft Visual SourceSafe 5.0/6.0" is selected as a version control system, other menu options of the [**Version Control->Select**] menu option will become available. See Chapter 9, Using Visual SourceSafe.

To deselect a version control system

1. Select the [**Tools->Version Control->Select**] menu option. The **Select Version Control System** dialog box will be displayed.
2. Select "<None>" from the **Version control systems** list.
3. Click the OK button.

7.2 Importing and exporting a set-up

Each workspace can have a different version control setup. The HEW allows you to store the version control settings independently so that you can import them into other workspaces. This greatly reduces the amount of time it takes to configure the same version control settings across several workspaces.

To export a version control setup

1. Select the [**Tools->Version Control->Configure**] menu option. The **Version Control Setup** dialog box will be displayed.
2. Click the **Export** button. An **Export current Version Control configuration** dialog box will be displayed.
3. Browse to the directory in which you would like to save the configuration.
4. Enter the name of the file and then click the **Save** button.

To import a version control setup

1. Select the [**Tools->Version Control->Configure**] menu option. The **Version Control Setup** dialog box will be displayed.
2. Click the **Import** button. An **Import a Version Control configuration** dialog box will be displayed.
3. Browse to the *.HVC file that you would like to import.
4. Select the file and then click the **Import** button.

8. Using the Custom Version Control System

If you have selected "Custom VCS" as the version control system in section 7.1, Select a Version Control System, the following definitions are necessary for connection of the HEW and Custom VCS.

- Version control menu options and locations of the associated command executables (.EXE), command parameters, how to control the execution result of version control commands, etc.
- Locations of files for version control (directory mapping) and global variables
- Execution control of version control commands, user settings, and other general options

After defining this information, you can execute a command of the custom version control system by selecting a HEW menu option or toolbar button and view the result in the HEW.

For installation and setting of the version control system, refer to the user's manual for respective version control systems.

For details on operations of the HEW with the custom version control system, see section 8.11, Usage example of the Custom Version Control System.

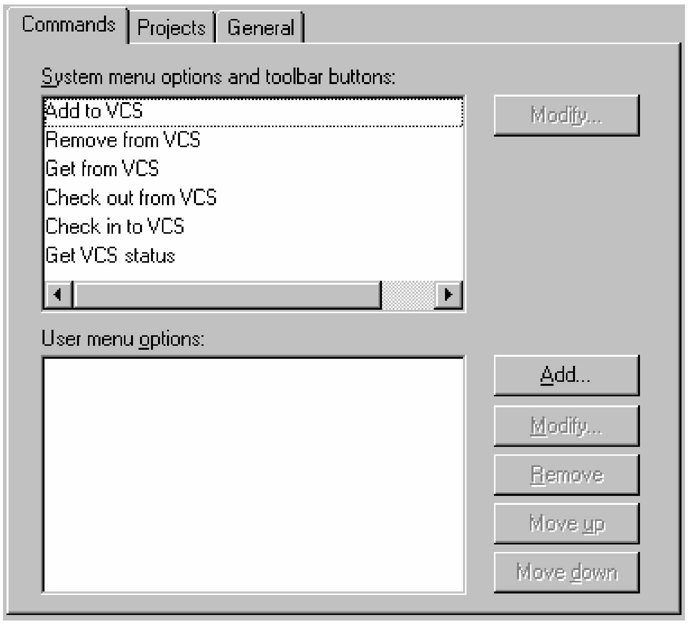
8.1 Defining Version Control menu options

The custom version control system allows you to invoke a version control command, either by selecting an option from the [**Tools->Version Control**] sub-menu, or by clicking a version control toolbar button. When either of these actions are performed, the associated commands are executed and the output is displayed in the **Version Control** tab of the output window.

To execute a command by a version–control menu option or toolbar button

1. Select the items to which you want to apply the version control command, from the Projects tab of the workspace window. This may include a workspace, projects, folders and files. Right-click to invoke a pop-up menu. You can select a menu option you have defined from the **Version Control** submenu. A command associated with this menu option will be executed on the files contained in the workspace, project, or folder, or file itself selected in the workspace window.
2. For example, if you select the workspace icon, all of the files in all of the projects will be passed to the version control command (this will include any HEW system files).
3. Select the required menu option from the [**Tools->Version Control**] sub-menu or click the desired version control toolbar button.

The custom version control support allows you the highest degree of flexibility in specifying how a version control system is to be used. To configure it, select the [**Tools->Version Control->Configure**] menu option. The **Version Control Setup** dialog box will be displayed.



The **Commands** tab has two lists of menu options.

- **System-defined menu options (System menu options and toolbar buttons)**

System menu options and toolbar buttons lists the menu options always shown as the [**Tools->Version Control**] sub-menus. These menu options also have an associated toolbar button on the **Version Control** toolbar. This contains the six menu options associated with the most frequently used version control commands.

- **User-defined menu options (User menu options)**

User menu options lists user-defined menu options. Menu options defined in this list will be added to the end of [**Tools->Version Control**] sub-menus. If you wish to add menu options not included in the system menu options, they must be defined in this list. User-defined menu options are not shown on the version control toolbar.

8.1.1 System-defined menu options

There are six version control toolbar buttons. They provide you with a shortcut to the most commonly used version control commands. Initially, when you first create a workspace, these toolbar buttons are inactive because you have not yet associated any version control commands to them.

The toolbar buttons are equivalent to the six menu options on the [**Tools->Version Control**] sub-menu. In other words, selecting the [**Tools->Version Control->Get from VCS**] menu option will have exactly the same effect as clicking the **Get from VCS** toolbar button. As the toolbar buttons themselves are fixed, the only operation that you can perform upon them is to define which commands should be executed when they are clicked.

In order to invoke commands from the toolbar or the system defined options of the [**Tools->Version Control**] sub-menu, you must first define the associated commands that should be executed when they are activated. The names of the options and their intended action are listed.

Option	Description
Add to VCS	Add selected files to version control system.
Remove from VCS	Remove selected files from version control system.
Get from VCS	Get a read only local copy of the selected files from version control system.
Check out from VCS	Get a writable local copy of the selected files from version control system.
Check in to VCS	Put back, i.e. update, the selected files in version control system with the local copy.
Get VCS status	View the status of the selected files.

To define a command to be executed via the system menu or toolbar button

1. Select the [**Tools->Version Control->Configure**] menu option. The **Version Control Setup** dialog box will be displayed.
2. Select a menu option in which you wish to define a command from **System menu options and toolbar buttons** list and click the **Modify** button. The **Define Commands** dialog box opens.
3. Click the **Add** button to define a command in the selected menu option. See section 8.2, Defining Version Control commands, for further information.
4. Close the **Define Commands** dialog box by clicking the OK button.
5. Close the **Version Control Setup** dialog box by clicking the OK button.

8.1.2 User-defined menu options

You can create as many user-defined menu options as you like, name them how you want and define their order in the menu.

If you have already created user-defined menu options, definitions of the commands can be modified or deleted. It is also possible to change the order of menu options being displayed.

User-defined menu options do not appear on the version control toolbar.

To create a new user-defined version control menu option

1. Select the [**Tools->Version Control->Configure**] menu option. The **Version Control Setup** dialog box will be displayed.
2. Click the **Add** button. The **Add Menu Option** dialog box will be displayed.
3. Enter the name of the menu option into the **Option** field.
4. Click the **Add** button. The **Add Command** dialog box will be displayed. For details, see section 8.2, Defining Version Control commands.
5. Click the OK button to close the **Add Command** dialog box.
6. Click the OK button to close the **Add Menu Option** dialog box.
7. Click the OK button to close the **Version Control Setup** dialog box.

To modify a user-defined version control menu option

1. Select the [**Tools->Version Control->Configure**] menu option. The **Version Control Setup** dialog box will be displayed.
2. Select the menu option to be modified from the **User menu options** list and then click the **Modify** button. The **Define Commands** dialog box will be displayed.
3. Select the menu option to be modified from the **Commands** list and then click the **Modify** button. The **Modify Commands** dialog box will be displayed. For details, see section 8.2, Defining Version Control commands.
4. Click the OK button to close the **Modify Commands** dialog box.
5. Click the OK button to close the **Define Commands** dialog box.
6. Click the OK button to close the **Version Control Setup** dialog box.

To remove a user-defined version control menu option

1. Select the [**Tools->Version Control->Configure**] menu option. The **Version Control Setup** dialog box will be displayed.
2. Select the menu option to be removed from the [**User menu options** list and click the **Remove** button.
3. Close the **Version Control Setup** dialog box by clicking the OK button.

To change the ordering of user-defined version control menu options

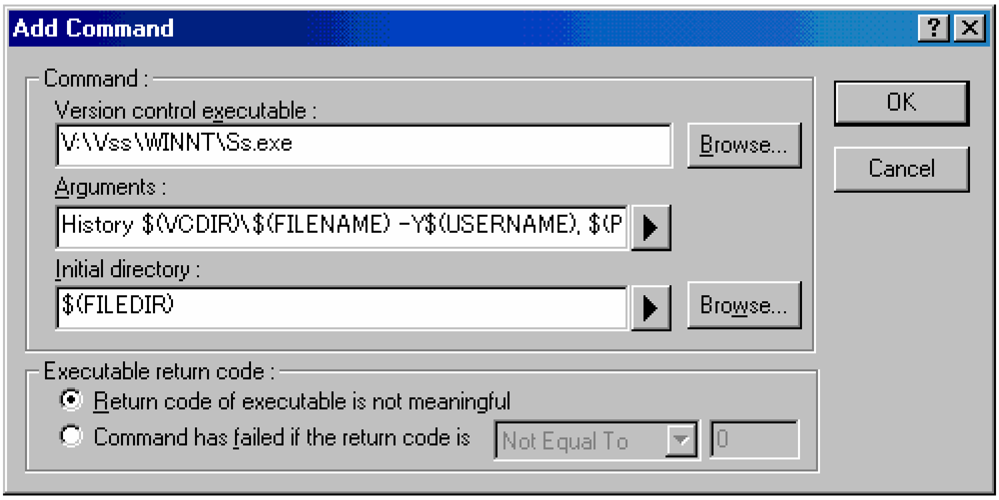
1. Select the [**Tools->Version Control->Configure**] menu option. The **Version Control Setup** dialog box will be displayed.
2. Select the menu option to be moved and then click the **Move up** and **Move down** buttons as necessary.
3. Close the **Version Control Setup** dialog box by clicking the OK button.

8.2 Defining Version Control commands

Version control commands are listed in the **Define Commands** dialog box. You can define as many commands as you want to and specify the order in which they execute. Existing commands can be modified or removed.

To define a new version control command

1. Click the **Add** button on the **Define Commands** dialog box. The **Add Command** dialog box will be invoked.



Version control executable : V:\Vss\WINNT\Ss.exe

Arguments : History $(VCDIR)\$(FILENAME) -Y$(USERNAME), $(P

Initial directory : $(FILEDIR)

● Return code of executable is not meaningful

○ Command has failed if the return code is Not Equal To 0

2. Enter the full path of the command into the **Version control executable** field, or browse to it graphically by clicking the **Browse** button.
3. Enter the arguments for the command into the **Arguments** field.
4. Enter into **Initial directory** the directory from which you would like to run the executable or browse to it graphically by clicking **Browse**. In most cases this should be set to the $(FILEDIR) placeholder, which means that the command should be executed from the same directory as the file.
5. Set the **Executable return code** options as appropriate (see section 8.5, Executable return code).
6. Click the OK button to define the new command.

To modify a version control command

1. Select the command to be modified from the **Commands** list of the **Define Commands** dialog box.
2. Click the **Modify** button. The **Modify Command** dialog box will be displayed.
3. Modify the information as necessary and then click the OK button.

To remove a version control command

1. Select the command to be removed from the **Commands** list of the **Define Commands** dialog box.
2. Click the **Remove** button.

To change the ordering of version control menu options

1. Select the menu option to be moved from the **Commands** list of the **Define Commands** dialog box.
2. Click the **Move up** and **Move down** buttons as necessary.

8.3 Specifying arguments

It is obvious that arguments must be specified correctly, otherwise the version control tool executed will not function as intended. However, it is also important, when using custom version control support, to specify the arguments in a flexible way, as a single version control command can be applied to more than one file. To facilitate this, the **Arguments** field has a placeholder button (see Reference 5, Placeholders, for an in depth discussion of placeholders), which when clicked on, invokes a pop-up menu of all available placeholders. An explanation of each version control placeholder and how their values are derived can be found in the table below.

Pop-up menu	Placeholder	Value And How It Is Determined
User login name	$(USERNAME)	Current user login ('General' tab)
User login password	$(PASSWORD)	Current user password ('General' tab)
Version control directory	$(VCDIR)	‘Virtual’ version control mapping ("Projects" tab)
Comment	$(COMMENT)	Comment specified before command execution
File path + name	$(FULLFILE)	Full path and name of the file involved in the operation
Filename	$(FILENAME)	Filename (including extension) of the file involved in the operation
File leaf	$(FILELEAF)	Filename (excluding extension) of the file involved in the operation
File extension	$(EXTENSION)	Extension of the file involved in the operation
File directory	$(FILEDIR)	Directory of the file involved in the operation
Configuration directory	$(CONFIGDIR)	Current configuration directory
Project directory	$(PROJDIR)	Current project directory
Workspace directory	$(WORKSPDIR)	Current workspace directory
Temp Directory	$(TEMPDIR)	Temporary directory
Command directory	$(EXEDIR)	Version control executable directory
Windows directory	$(WINDIR)	Directory where Windows® is installed
Windows system directory	$(WINSYSDIR)	Directory where Windows® system files exist
Workspace name	$(WORKSPNAME)	Current workspace name
Project name	$(PROJECTNAME)	Current project name
Configuration name	$(CONFIGNAME)	Current configuration name

8.4 Specifying comments

If a version control command contains the placeholder ‘$(COMMENT)’, the HEW will request that you enter the comment when the command is executed (via the **Please Enter Comment** dialog box).

You may specify a comment for each file or, if you would like to specify the same comment for all files, check the **Apply comment to all files** checkbox before clicking the OK button.

8.5 Executable return code

While each version control command executes, its output is redirected to the **Version Control** tab of the output window. When the command’s execution is complete, its return code is obtained. When defining a command, you can determine whether this return code can be used to indicate that the command failed and that the remaining commands should not be executed (i.e. abort). The controls contained in the **Executable return code** group allow you to specify this behavior.

If the return code of the commands can be used to indicate a failure then you should select the **Command has failed if the return code is** option and set the drop-down list and edit box as required. If the **Command has failed if the return code is** option is selected then the HEW will check the return code of each command to determine whether a failure occurred. If this is the case, no further commands will be executed and any other processes which would follow the commands (e.g. build) will not be executed.

If the **Return code of tool is not meaningful** option is selected then the HEW will not check the return code of each command. Consequently, all commands will execute regardless.

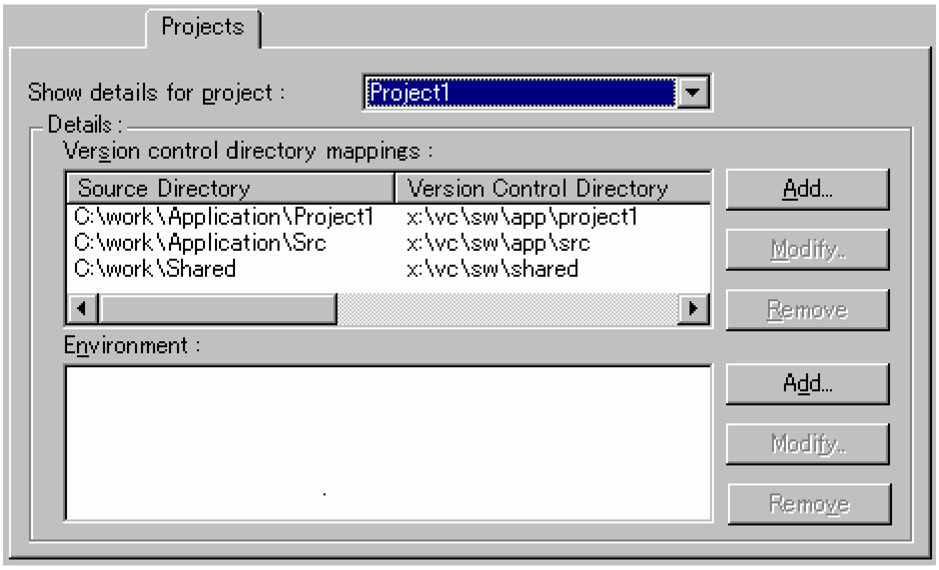
8.6 Specifying file locations

When referring to a file's location, be sure to use a placeholder, otherwise the command will only relate to a hardwired file. For example, let's imagine that a version control executable has been selected which uses a –GET command to obtain a read-only copy of a file. The **Arguments** field could be specified as:

```
-GET 'c:\vc\files\project\main.c'
```

However, when executed, this command can only ever -GET the file MAIN.C. To resolve this problem, HEW uses a system of placeholders and directory mappings. Directory mappings tell the HEW which 'working' directories (i.e. where source files are being worked on) map to which 'controlled' directories (i.e. where the source files are stored in the version control system). Mappings between these two directory systems can be specified via the **Projects** tab of the **Version Control Setup** dialog box.

Once the mappings have been defined, you can use the **Version control directory** placeholder, $(VCDIR), to represent the directory in which the project file is stored.



Show details for project : Project1

Details :

Version control directory mappings :

Source Directory	Version Control Directory
C:\work\Application\Project1	x:\vc\sw\app\project1
C:\work\Application\Src	x:\vc\sw\app\src
C:\work\Shared	x:\vc\sw\shared

Environment :

To define a new directory mapping

1. Select [**Tools->Version Control->Configure**]. The **Version Control Setup** dialog box will be displayed.
2. Select the **Projects** tab.
3. Click the **Add** button, which is next to the **Version Control Directory Mappings** list. The **Version Control Directory Mapping** dialog box will be displayed.
4. Specify a HEW workspace directory or project directory in the **Source directory** field. Select one of the following operations:

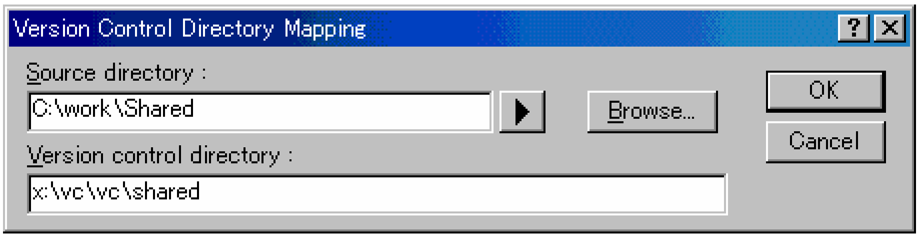
- Enter the directory name, **OR**

- Click the placeholder button. Then select "Project directory" or "Workspace directory" from the menu, **OR**
- Click the **Browse** button to open the **Browse to Source Directory** dialog box. Select the directory and click the **Select** button.

5. Enter the version control directory into the **Version Control Directory** field.
6. Click the OK button.

To modify a directory mapping

1. Select [**Tools->Version Control->Configure**]. The **Version Control Setup** dialog box will be displayed. Select the **Projects** tab.
2. Select the mapping to be modified from the **Version Control Directory Mappings** list and then click the **Modify** button. The **Version Control Directory Mapping** dialog box will be displayed.
3. Make the necessary changes to the two directories and then click the OK button to confirm the new settings.



Version Control Directory Mapping

Source directory : C:\work\Shared

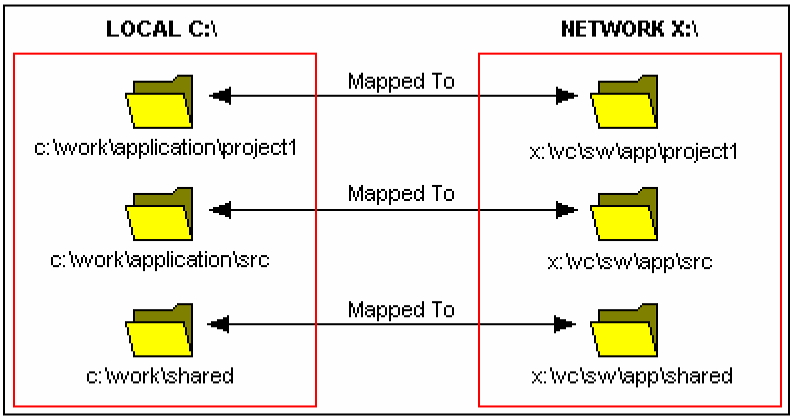
Version control directory : x:\vc\vc\shared

To remove a directory mapping

1. Select [**Tools->Version Control->Configure**]. The **Version Control Setup** dialog box will be displayed. Select the **Projects** tab.
2. Select the mapping to be removed from the **Version Control Directory Mappings** list and then click the **Remove** button.

8.7 Specifying file locations example

Consider the scenario shown in the figure below. It shows three directories, which are mapped from a shared version control drive (X:\) to a local drive where the development is being done (C:\).



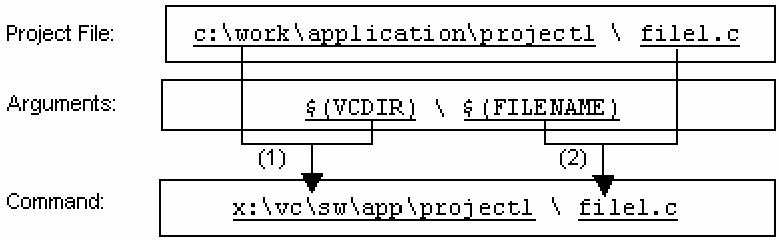
Now let’s imagine that a version control executable has been selected which uses a -GET command to obtain a read-only copy of a file. In order to get all of the files in a project we need to use the following command:

-GET '$(VCDIR)\$(FILENAME)'

When the HEW executes the command for a given project file, it will replace $(VCDIR) for the equivalent version control directory in the file mapping.

For example, suppose FILE1.C is located at c:\work\application\project1\FILE1.C. If the -GET command is applied to FILE1.C then:

1. 'x:\vc\sw\app\project1' is substituted for '$(VCDIR)', as this is the version control directory mapping for 'c:\work\application\project1'.
2. 'FILE1.C' is substituted for '$(FILENAME)'.

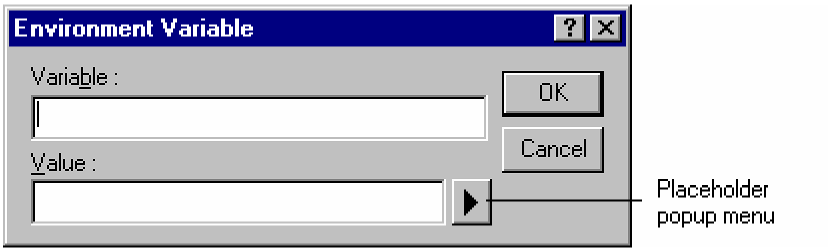

Project File:
c:\work\application\project1 \ file1.c
Arguments:
$(VCDIR) \ $(FILENAME)
(1)
(2)
Command:
x:\vc\sw\app\project1 \ file1.c

8.8 Specifying environment

Select the **Projects** tab of the **Version Control Setup** dialog box to view the current settings.

To add a new environment variable

1. Click the **Add** button beside the **Environment** list (the **Environment Variable** dialog box will be invoked).



2. Enter the variable name into the **Variable** field.
3. Enter the variable's value into the **Value** field.
4. Click the OK button to add the new variable to the Environment list.

To modify an environment variable

1. Select the variable that you want to modify from the **Environment** list.
2. Click the **Modify** button beside the list.
3. Make the required changes to the **Variable** and **Value** fields.
4. Click the OK button to add the modified variable back to the list.

To remove an environment variable

1. Select the variable that you want to remove from the **Environment** list.
2. Click the **Remove** button beside the list.

8.9 Controlling execution of a Version Control System

The **General** tab of the **Version Control Setup** dialog box allows you to control the way in which the version control system is executed. It also shows the full path to the current version control configuration file.

The execution of a version control system can be modified via the following three checkboxes:

- **Prompt before executing command**

If this checkbox is selected then, before any version control commands are executed, a dialog is displayed, which lists all of the files involved in the operation. Files may be deselected by clearing the associated checkbox. Clicking the OK button will apply the command to each of the selected files. Clicking the Cancel button will abort the operation.

- **Run in DOS window**

By default, the output of the version control commands is redirected to the **Version Control** tab of the output window. If you would rather run each command in a separate DOS window then set this checkbox. Select **Close DOS window on exit** checkbox if you wish the DOS window to close after execution has been completed.

- **Use forward slash '/' as version control directory delimiter**

By default, when the HEW substitutes the $(VCDIR) placeholder, it uses the backward slash character '\' to divide directories. However, if the version control system you are using uses a forward slash character (e.g. Visual SourceSafe) to divide directories then select the **Use forward slash '/' as version control directory delimiter** checkbox.

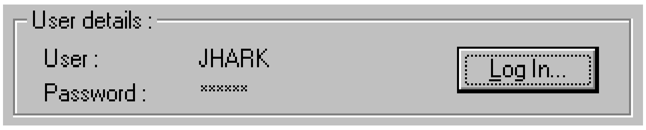
8.10 Specifying a user name and password

Most version control tools will require you to pass a username and password on the command line in order to keep files secure, and to keep a record of which files were changed by which users. The custom version control support provides two placeholders: **User login name**, $(USERNAME), and **User login password**, $(PASSWORD). When the command is executed, these placeholders will be replaced with the current settings in the **General** tab of the **Version Control Setup** dialog box.

In order to give the $(USERNAME) and $(PASSWORD) fields a value you will first need to login. If you have not logged in before a command is executed, which uses either of these placeholders, then you will be prompted to do so before the command can be executed.

To login (i.e. specify a username and password)

1. Select the [**Tools->Version Control->Configure**] menu option. The **Version Control Setup** dialog box will be displayed.
2. Select the **General** tab.

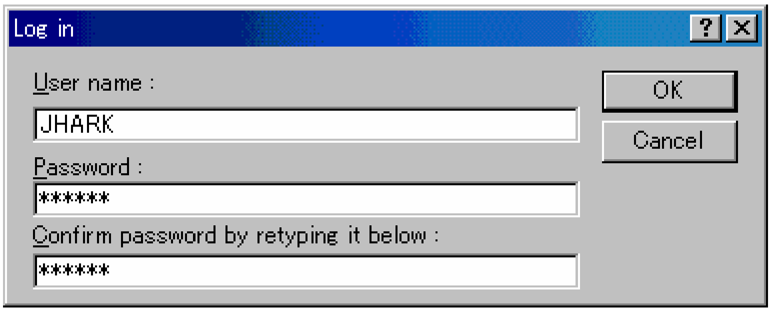


User details :

User : JHARK

Password : ******

3. Click the **Log In** button. The **Log in** dialog box will be displayed.



Log in

User name : JHARK

Password : ******

Confirm password by retyping it below : ******

4. Enter your username into the **User name** field.
5. Enter your password into the **Password** field.
6. Re-type your password again into the **Confirm password by retyping it below** field.
7. Click the OK button to set the new username and password. If there is any inconsistency between the two versions of the password that you entered, you will be requested to type your password again.

8.11 Usage example of the Custom Version Control System

The custom version control system includes any version control system in which command-line commands should be executed via the MS-DOS command prompt. The usage examples in this section use the following version control systems to be connected with the HEW.

- Connecting the HEW with RCS
- Connecting the HEW with CVS

8.11.1 Connecting the HEW with RCS

In this example, the HEW is to be connected with RCS.

- **RCS environment**

The following types of version control system can be used:

- GNU RCS
- GNU diff

This procedure assumes that the version control system is located at "C:\RCS".

- **Workspace**

Create a HEW workspace in the following folder.

C:\WorkSpace\rcs_sample

Then create a folder for RCS.

C:\WorkSpace\rcs_sample\rcs_sample\RCS

The following pages describe the procedures to create a workspace, make settings for version control, check-in, compare, check-out, and view logs.

Step 1: Create a workspace

Step 2: Selecting Custom Version Control System

Step 3: Making settings for Version Control

Step 4: Using the Version Control facility (Check-in)

Step 5: Using the Version Control facility (Compare)

Step 6: Using the Version Control facility (Check-out)

Step 7: Using the Version Control facility (View Logs)

(1) Step 1: Create a workspace

1. Select [**File->New Workspace**] to open the **New Project Workspace** dialog box.
2. Create a workspace with the following conditions.

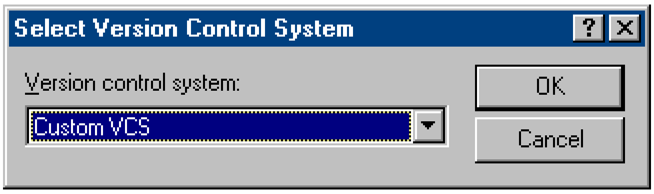
- Workspace Name: rcs_sample
- Project Name: rcs_sample
- Directory: C:\WorkSpace\rcs_sample

3. Create a folder for RCS.

   C:\WorkSpace\rcs_sample\rcs_sample\RCS

(2) Step 2: Selecting Custom Version Control System

1. Select [**Tools->Version Control->Select**]. The **Select Version Control System** dialog box opens.



Custom VCS

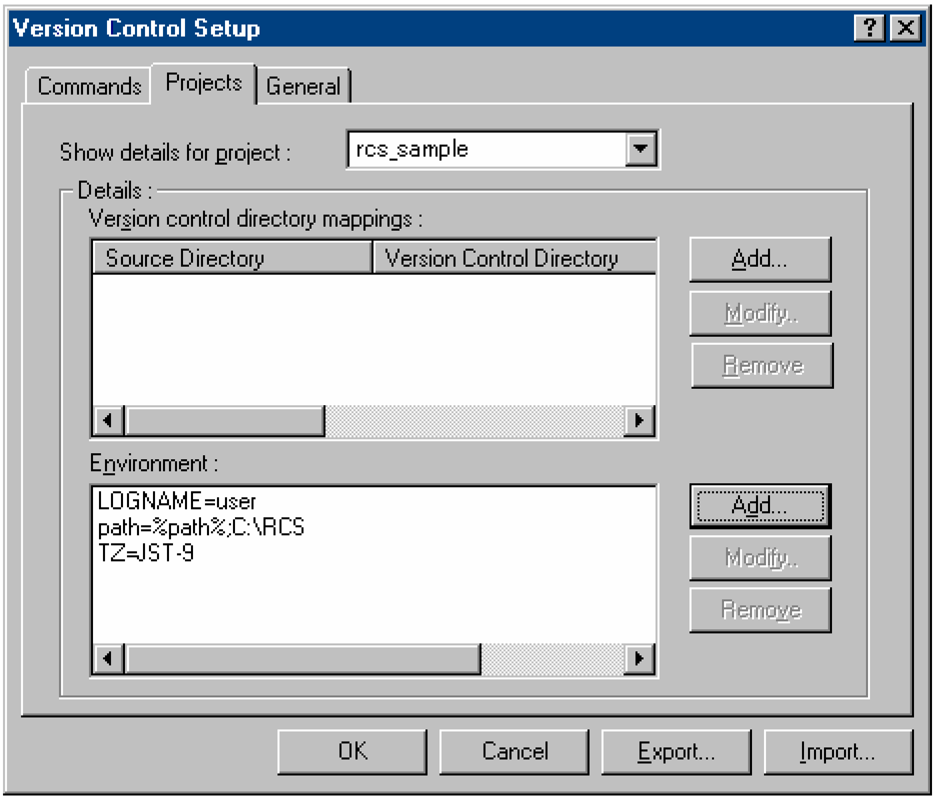
2. Select **Custom VCS**.
3. Click the OK button.

(3) Step 3: Making settings for Version Control

Select [**Tools->Version Control->Configure**] to open the **Version Control Setup** dialog box.

- **Setting environment variables**

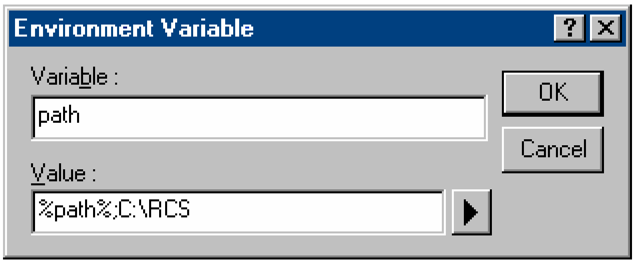
1. Select the **Projects** tab.



Show details for project : rcs_sample

```
LOGNAME=user
path=%path%;C:\RCS
TZ=JST-9
```

2. Click the **Add** button on **Environment**.
3. The **Environment** dialog box appears.



Variable : path

Value : %path%;C:\RCS

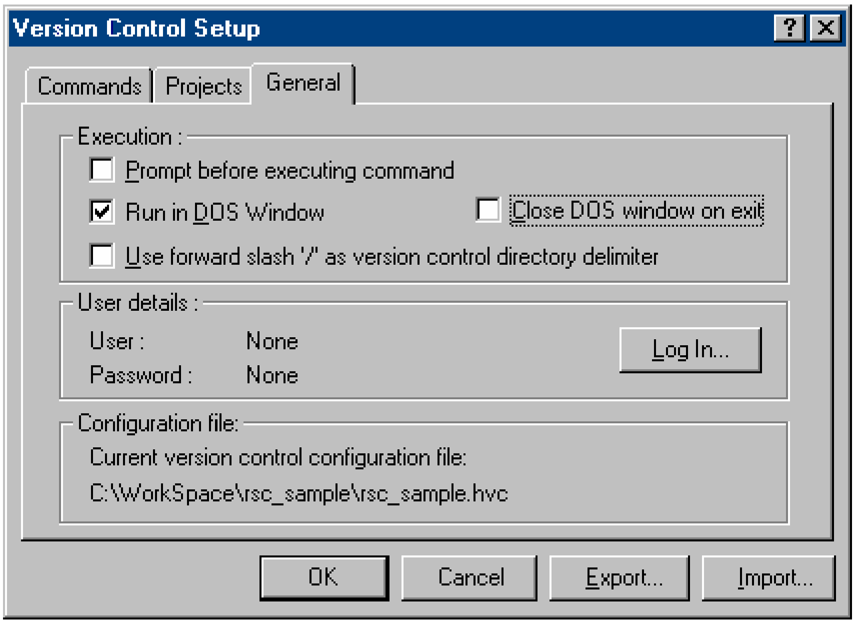
4. Enter "path" in **Variable** and "%path%;C:\RCS" in **Value**, respectively.
5. Click the OK button.
6. Click the **Add** button. Also enter the following environment variables in **Environment**.

LOGNAME=user

TZ=JST-9

- **Specifying actions**

1. Select the **General** tab.



Execution :

□ Prompt before executing command

☑ Run in DOS Window □ Close DOS window on exit

□ Use forward slash '/' as version control directory delimiter

User details :

User : None

Password : None

Configuration file:

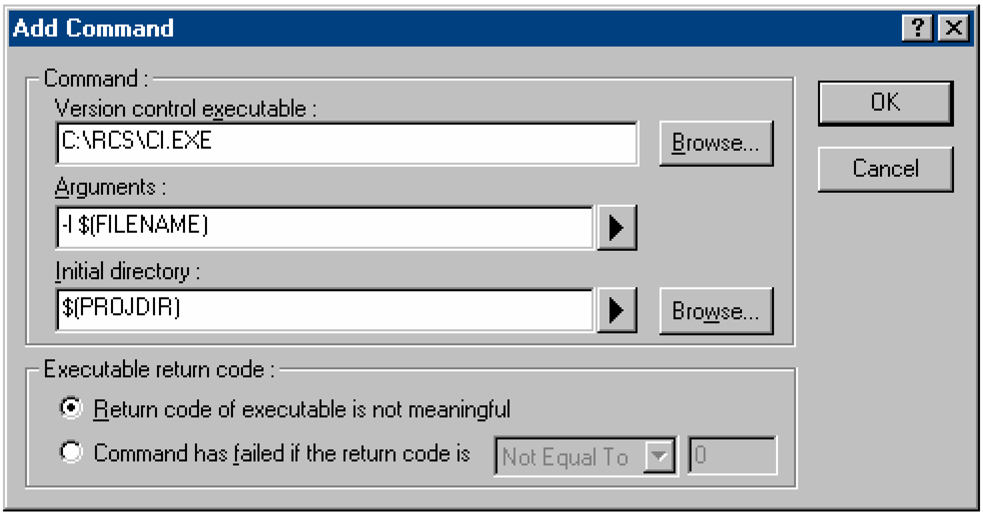
Current version control configuration file:

C:\WorkSpace\rsc_sample\rsc_sample.hvc

2. Deselect the **Prompt before executing command** checkbox. (Leave it selected, however, if you wish to see the file name before executing commands.)
3. Select the **Run in DOS Window** checkbox. (This is because the RCS commands must be input via the MS-DOS prompt.)
4. Deselect the **Close DOS Window on exit** checkbox. (This is because errors must be viewed in the MS-DOS prompt.)

- **Setting up commands**

1. Select the **Commands** tab.
2. Click the **Add** button on **User menu options**.
3. The **Add Menu Option** dialog box appears.
4. Enter "Initialize" in **Option** and click the **Add** button.
5. The **Add Command** dialog box appears.



Command :
Version control executable : C:\RCS\CI.EXE
Arguments : -l $(FILENAME)
Initial directory : $(PROJDIR)
Executable return code :
● Return code of executable is not meaningful
○ Command has failed if the return code is Not Equal To 0

6. Enter "C:\RCS\CI.EXE" in **Version control executable**.
7. Enter "-l $(FILENAME)" in **Arguments**.
8. Enter "$(PROJDIR)" in **Initial directory**.
9. Click the OK button.
10. Click the OK button in the **Add Menu Option** dialog box.

Tip:

It is recommended that you only register some commands frequently used for version control. Other commands should be used via the MS-DOS command prompt.

Also set up the following tree commands in the same way.

Command	Option Name	Characters to be Input
Check out	Version control executable	C:\RCS\CO.EXE
	Argument	-l -f $(FILENAME)
	Initial directory	$(PROJDIR)
Diff	Version control executable	C:\RCS\RCSDIFF.EXE
	Argument	(FILENAME)
	Initial directory	$(PROJDIR)
Log	Version control executable	C:\RCS\RLOG.EXE
	Argument	(FILENAME)
	Initial directory	$(PROJDIR)

Preparation for version control is now completed.

(4) Step 4: Using the Version Control facility (Check-in)

1. Enter a keyword at line 12 in the file "rcs_sample.c" so that you will be able to see the changes made in the file.

   //Id

2. Select the file "rcs_sample.c" in the **Projects** tab of the workspace window and then select [**Tools->Version Control->Check in**].
3. The MS-DOS command prompt window opens.
4. Enter a comment for check-in after the prompt (">>") and press Enter.
5. Enter one period '.' and press Enter.

```
rcs_sample.c,v  <--  rcs_sample.c
enter description, terminated with single '.' or end of file:
NOTE: This is NOT the log message!
>> .
initial revision: 1.1
done
C:\RCS>
```

6. Enter "exit" and press Enter. This closes the MS-DOS command prompt window.
7. A HEW confirmation dialog box appears. Click **Yes**. The keyword now includes the file name, revision number, date, and user name.

Tip:

If you do not wish to see the result of command execution in the MS-DOS command prompt window, select the **Close DOS Window on exit** checkbox in the **General** tab of the **Version Control Setup** dialog box.

(5) Step 5: Using the Version Control facility (Compare)

1. To compare with the checked-in file, modify the file "rcs_sample.c" and save it.
2. Select the file "rcs_sample.c" in the **Projects** tab of the workspace window and select [**Tools->Version Control->Diff**].
3. The MS-DOS command prompt window appears and shows the result of comparison.
4. Enter "exit" and press the Enter key. This closes the MS-DOS command prompt window.

```
RCS file: rcs_sample.c,v
Working file: rcs_sample.c
head: 1.1
branch:
locks: strict
        user: 1.1
access list:
symbolic names:
keyword substitution: kv
total revisions: 1;     selected revisions: 1
description:
----------------------------
revision 1.1    locked by: user;
date: 2006/03/14 07:22:42;  author: user;  state: Exp;
Initial revision
=============================================================================
```

(6) Step 6: Using the Version Control facility (Check-out)

Use the check-out command to undo the changes made in the file.

1. Select the file "rcs_sample.c" in the **Projects** tab of the workspace window and select [**Tools->Version Control->Check out**].
2. The MS-DOS command prompt window appears and shows the result of check-out.
3. Enter "exit" and press the Enter key. This closes the MS-DOS command prompt window.

4. A HEW confirmation dialog box appears. Click **Yes**. The file has returned to its original state.

(7) Step 7: Using the Version Control facility (View Logs)

Use the log command to view the changes.

1. Select the file "rcs_sample.c" in the **Projects** tab of the workspace window and select [**Tools->Version Control->Log**].
2. The MS-DOS command prompt window appears and shows the log information.
3. Enter "exit" and press the Enter key. This closes the MS-DOS command prompt window.

8.11.2 Connecting the HEW with CVS

In this example, the HEW is to be connected with CVS.

- **CVS tool environment**

The following type of version control system can be used:

- CVS

This procedure assumes that the version control system is located at "C:\cvs-1-11-17".

- **Workspace**

Create a HEW workspace in the following folder.

C:\WorkSpace\sampleCVS

Then create a folder for CVS.

C:\src\sampleCVS

Note:

In this version control using the HEW and CVS, only versions of source files can be controlled. It cannot control versions of any other files (such as workspace or project files).

The following pages describe the procedures to create a workspace, make settings for version control, create a repository, register a module, check-out, check the status, compare, register the result, and view logs.

Step 1: Create a workspace

Step 2: Selecting Custom Version Control System

Step 3: Making settings for Version Control

Step 4: Using the Version Control facility (Create a Repository)

Step 5: Using the Version Control facility (Register a Module)

Step 6: Using the Version Control facility (Check-out)

Step 7: Using the Version Control facility (Modify the File)

Step 8: Using the Version Control facility (Check the Status)

Step 9: Using the Version Control facility (Compare)

Step 10: Using the Version Control facility (Register the Result)

Step 11: Using the Version Control facility (View Logs)

(1) Step 1: Create a workspace

1. Select [**File->New Workspace**] to open the **New Project Workspace** dialog box.
2. Create a workspace with the following conditions.

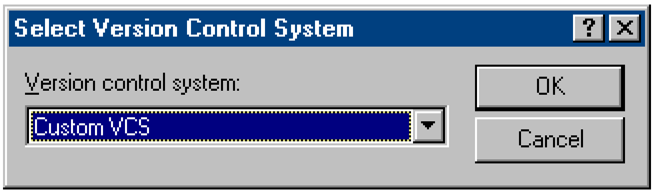
- Workspace Name: sampleCVS
- Project Name: sampleCVS
- Directory: C:\WorkSpace\sampleCVS

3. Exit the HEW.
4. Create a folder for CVS.

   C:\src\sampleCVS
5. Move the file you wish to control from "C:\WorkSpace\sampleCVS\sampleCVS" to "C:\src\sampleCVS".
6. Start up the HEW and open the created workspace "sampleCVS".
7. Select the [**Project->Remove Files**] menu to open the **Remove Project Files** dialog box.
8. Select the file you wish to control and click the **Remove** button.
9. Click OK button.
10. Select the [**Project->Add Files**] menu to open the **Add files to project 'sampleCVS'** dialog box.
11. Select the file you wish to control and click the **Add** button.

(2) Step 2: Selecting Custom Version Control System

1. Select [**Tools->Version Control->Select**].



Select Version Control System

Version control system: Custom VCS

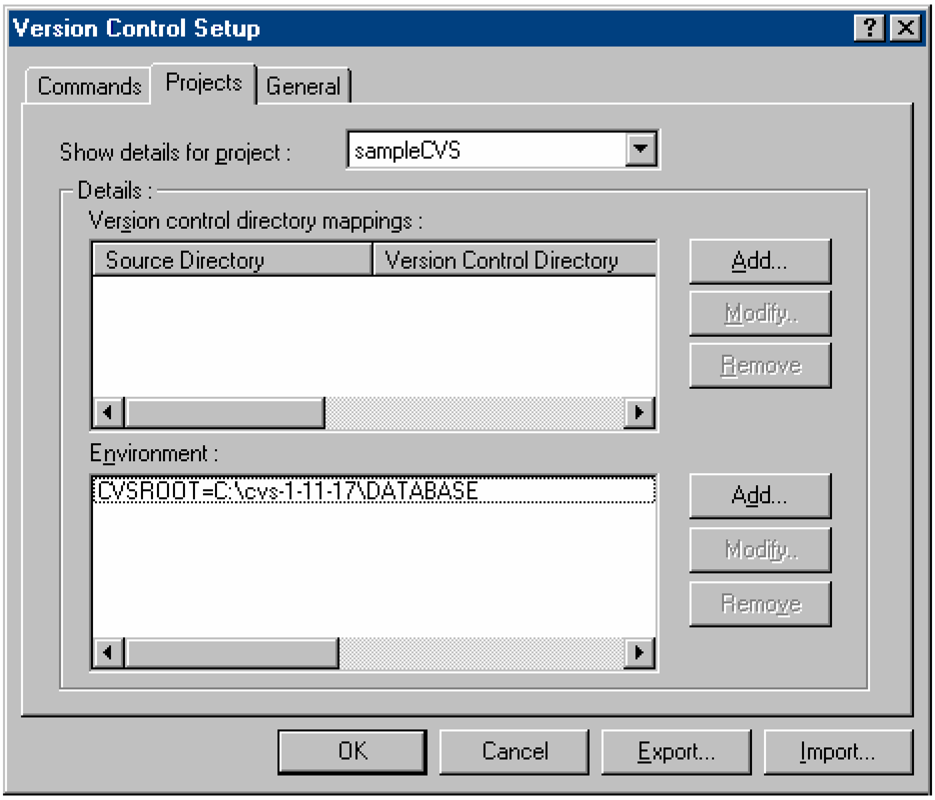
2. The **Select Version Control System** dialog box opens.
3. Select **Custom VCS**.
4. Click the OK button.

(3) Step 3: Making settings for Version Control

Select [**Tools->Version Control->Configure**] to open the **Version Control Setup** dialog box.

- **Setting environment variables**

1. Select the **Projects** tab.
2. Click the **Add** button on **Environment**.
3. The **Environment** dialog box appears.
4. Enter "CVSROOT" in **Variable** and "C:\cvs-1-11-17\DATABASE" in **Value**, respectively.



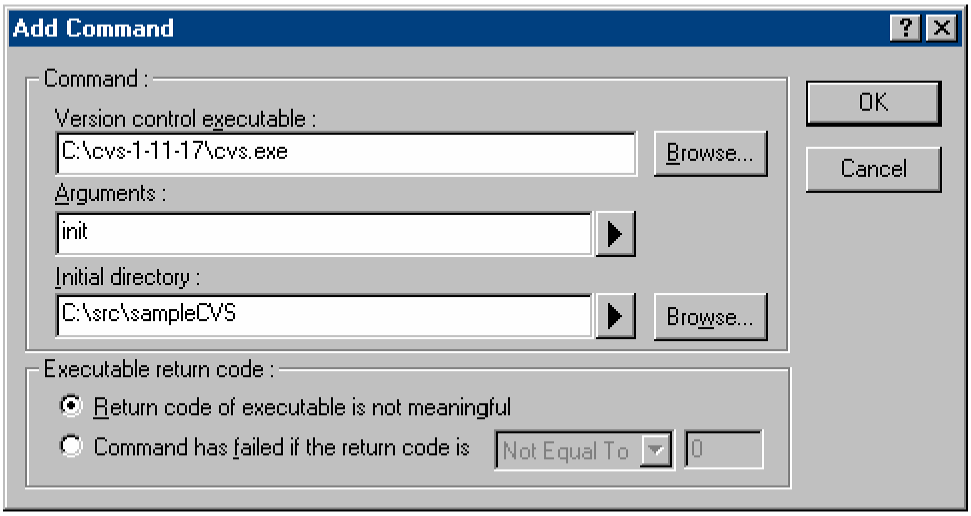
Show details for project : sampleCVS

Environment : CVSROOT=C:\cvs-1-11-17\DATABASE

5. Click the OK button.

- **Setting up commands**

1. Select the **Commands** tab.
2. Click the **Add** button on **User menu options**.
3. The **Add Menu Option** dialog box appears.
4. Enter "Initialize" in **Option** and click the **Add** button.
5. The **Add Command** dialog box appears.
6. Enter "C:\cvs-1-11-17\cvs.exe" in **Version control executable**.
7. Enter "init" in **Arguments**.
8. Enter "C:\src\sampleCVS" in **Initial directory**.



Version control executable : C:\cvs-1-11-17\cvs.exe

Arguments : init

Initial directory : C:\src\sampleCVS

● Return code of executable is not meaningful

○ Command has failed if the return code is Not Equal To 0

9. Click the OK button.

10. Click the OK button in the **Add Menu Option** dialog box.

Also set up the following six commands in the same way.

Command	Option Name	Characters to be Input
Register	Version control executable	C:\cvs-1-11-17\cvs.exe
	Argument	import -m "New Source" sampleCVS Renesas rel-20060228
	Initial directory	C:\src\sampleCVS
Check out	Version control executable	C:\cvs-1-11-17\cvs.exe
	Argument	checkout sampleCVS
	Initial directory	C:\src
Status	Version control executable	C:\cvs-1-11-17\cvs.exe
	Argument	status $(FILENAME)
	Initial directory	C:\src\sampleCVS
Diff	Version control executable	C:\cvs-1-11-17\cvs.exe
	Argument	diff $(FILENAME)
	Initial directory	C:\src\sampleCVS
Commit	Version control executable	C:\cvs-1-11-17\cvs.exe
	Argument	commit -m "modify" $(FILENAME)
	Initial directory	C:\src\sampleCVS
Log	Version control executable	C:\cvs-1-11-17\cvs.exe
	Argument	log $(FILENAME)
	Initial directory	C:\src\sampleCVS

Preparation for version control is now completed.

(4) Step 4: Using the Version Control facility (Create a Repository)

1. Select the [**Tools->Version Control->Initialize**] menu.
2. Select the file in the **Initialize** dialog box and click the OK button.

```
Executing C:\cvs-1-11-17\cvs.exe init
Command(s) completed successfully.
```

(5) Step 5: Using the Version Control facility (Register a Module)

1. Select the [**Tools->Version Control->Register**] menu.
2. Select the file in the **Register** dialog box and click the OK button.

```
Executing C:\cvs-1-11-17\cvs.exe import -m "New Source" sampleCVS Renesas rel-
20060228
N sampleCVS/dbsct.c
(Omit)
No conflicts created by this import
Command(s) completed successfully.
```

(6) Step 6: Using the Version Control facility (Check-out)

1. Select the [**Tools->Version Control->Checkout**] menu.
2. Select the file in the **Checkout** dialog box and click the OK button.
3. The **Version Control** tab of the output window shows the following message.

```
Executing C:\cvs-1-11-17\cvs.exe checkout sampleCVS
U sampleCVS/dbsct.c
```

```
(Omit)
cvs.exe checkout: Updating sampleCVS
Command(s) completed successfully.
```

(7) Step 7: Using the Version Control facility (Modify the File)

Use the HEW editor to modify the file (and save it). In this example, the file "sampleCVS.c" is modified.

(8) Step 8: Using the Version Control facility (Check the Status)

1. The status of the file "sampleCVS.c" should be checked.
2. Select the [**Tools->Version Control->Status**] menu.
3. Select the file in the **Status** dialog box and click the OK button.
4. The **Version Control** tab of the output window shows the following message.

```
Executing C:\cvs-1-11-17\cvs.exe status sampleCVS.c
===================================================================
File: sampleCVS.c      Status: Locally Modified
   Working revision: 1.1.1.1 Tue Feb 28 02:46:30 2006
   Repository revision: 1.1.1.1 C:\cvs-1-11-17\DATABASE/sampleCVS/sampleCVS.c,v
   Sticky Tag: (none)
   Sticky Date: (none)
   Sticky Options: (none)
Command(s) completed successfully.
```

The status of the file "sampleCVS.c" is shown as "Locally Modified".

(9) Step 9: Using the Version Control facility (Compare)

1. The changes made in the file "sampleCVS.c" should be checked.
2. Select the [**Tools->Version Control->Diff**] menu.
3. Select the file in the **Diff** dialog box and click the OK button.
4. The **Version Control** tab of the output window shows the following message.

```
Executing C:\cvs-1-11-17\cvs.exe diff sampleCVS.c
Index: sampleCVS.c
===================================================================
RCS file: C:\cvs-1-11-17\DATABASE/sampleCVS/sampleCVS.c,v
retrieving revision 1.1.1.1
diff -r1.1.1.1 sampleCVS.c
28c28,31
< printf("### Data Input ###\n");
---
> //loop
> while(1)
> {
> printf("### Data Input ###\n"); //loop
49a53,54
> //loop
> }
Command(s) completed successfully.
```

You can see the changes made in the file "sampleCVS.c".

(10)Step 10: Using the Version Control facility (Register the Result)

1. Select the [**Tools->Version Control->Commit**] menu.
2. Select the file in the **Commit** dialog box and click the OK button.
3. The **Version Control** tab of the output window shows the following message.

```
Executing C:\cvs-1-11-17\cvs.exe commit -m "modify" sampleCVS.c
Checking in sampleCVS.c;
C:\cvs-1-11-17\DATABASE/sampleCVS/sampleCVS.c,v  <--  sampleCVS.c
new revision: 1.2; previous revision: 1.1
done
Command(s) completed successfully.
```

(11)Step 11: Using the Version Control facility (View Logs)

1. Select the [**Tools->Version Control->Log**] menu.
2. Select the file in the **Log** dialog box and click the OK button.
3. The **Version Control** tab of the output window shows the following message.

```
Executing C:\cvs-1-11-17\cvs.exe log sampleCVS.c
RCS file: C:\cvs-1-11-17\DATABASE/sampleCVS/sampleCVS.c,v
Working file: sampleCVS.c
head: 1.2
branch:
locks: strict
access list:
symbolic names:
rel-20060228: 1.1.1.1
Renesas: 1.1.1
keyword substitution: kv
total revisions: 3; selected revisions: 3
description:
----------------------------
revision 1.2
date: 2006/02/28 03:39:22;  author: user;  state: Exp;  lines: +6 -1
modify
----------------------------
revision 1.1
date: 2006/02/28 02:46:30;  author: user;  state: Exp;
branches:  1.1.1;
```

```
Initial revision
----------------------------
revision 1.1.1.1
date: 2006/02/28 02:46:30;  author: user;  state: Exp;  lines: +0 -0
New Source
=============================================================================
Command(s) completed successfully.
```

9. Using Visual SourceSafe

The High-performance Embedded Workshop provides specific support for the Visual SourceSafe version control system.

The Visual SourceSafe version control system associates a project in your workspace with a project inside a Visual SourceSafe database.

Select a menu option from the [**Tools->Version Control**] submenu or click a **Version Control** toolbar button to quickly activate the commands most frequently used in the Visual SourceSafe version control system.

For installation and option settings of Visual SourceSafe, refer to the online help or user's manual for Visual SourceSafe. In this section, it is assumed that the login username and password have been set.

The outline of procedures for version control by the HEW connected with Visual SourceSafe is described below. For other functions available, see section 9.2, Visual SourceSafe commands.

1. Attaching Visual SourceSafe to a workspace

Associate a project in a HEW workspace with a project inside a Visual SourceSafe database.

- Select "Microsoft Visual SourceSafe 5.0/6.0" as the version control system to be connected with the HEW.
- Login Visual SourceSafe.
- Specify the name and location of the project to be created in Visual SourceSafe.

2. Adding files

Select the files in a HEW workspace that you would like to add to Visual SourceSafe.

3. Checking out a file

Select the files that you would like to modify in the HEW workspace and check them out.

4. Checking in a file

Select the files modified in the HEW workspace and check them in.

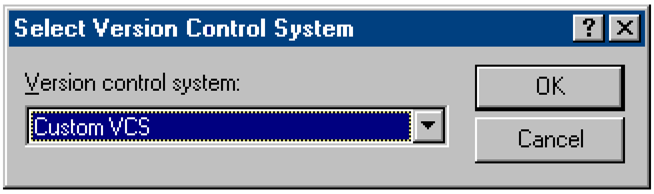
9.1 Attaching Visual SourceSafe to a workspace

The following sections describe how you can associate Visual SourceSafe with your current workspace.

First, you need to select Visual SourceSafe as the version control system to be connected with the HEW.

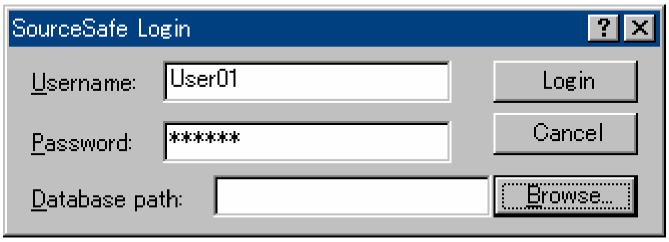
To attach Visual SourceSafe to a workspace

1. Select [**Tools->Version Control->Select**]. The **Select Version Control System** dialog box was displayed.
2. Select the "Microsoft Visual SourceSafe 5.0/6.0" entry from the **Version control systems** list.



Version control system: Custom VCS

3. Click OK. The **SourceSafe Login** dialog box was displayed.



SourceSafe Login

Username: User01

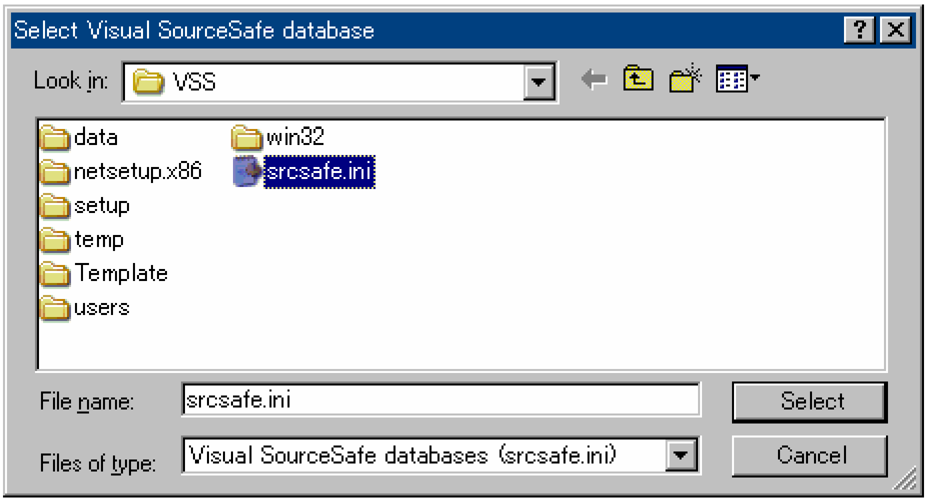
Password: ******

Database path:

4. Enter your Visual SourceSafe username into the **Username** field and password into the **Password** field.

5. For the **Database path** field, specify the Visual SourceSafe database file (SRCSAFE.INI) into which you would like to add this project. Select either of the following operations:

- Click the **Browse** button to open the **Select Visual SourceSafe database** dialog box. Select the Visual SourceSafe database file and click the **Select** button, **OR**



Select Visual SourceSafe database

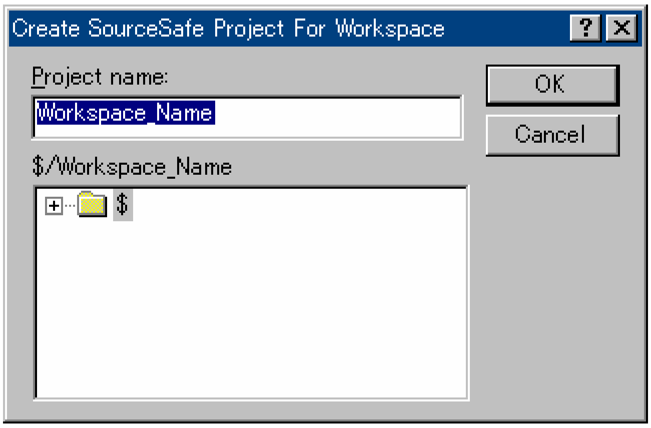
Look in: VSS

File name: srcsafe.ini

Files of type: Visual SourceSafe databases (srcsafe.ini)

- Enter the full path of the Visual SourceSafe database file in the **Database path** field.

6. The **Create SourceSafe Project For Workspace** dialog box was displayed.



Create SourceSafe Project For Workspace

Project name: Workspace_Name

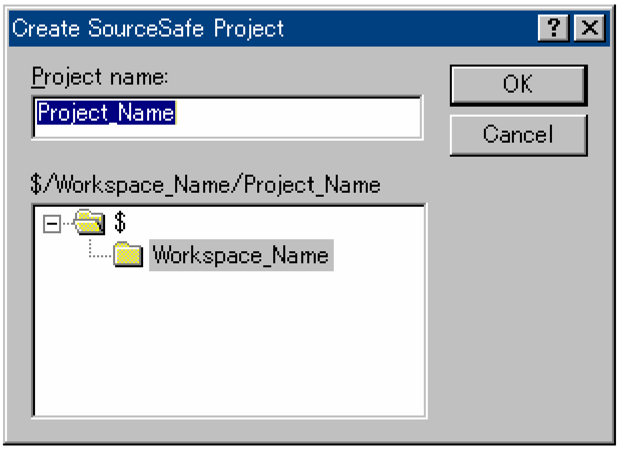
$/Workspace_Name

7. The **Project name** field shows the name of the HEW workspace. The **Project name** field displays the name of the project (i.e. folder) to be created in the database. If necessary you can change this name to another.

8. The tree underneath the **Project name** field shows the structure of the database specified in Step 5. Select the folder into which you would like to create the folder specified in **Project name**.

9. Click the OK button in the **Create SourceSafe Project For Workspace** dialog box.

10. The **Create SourceSafe Project** dialog box was displayed. The example below shows that the HEW workspace “Workspace_Name” is added as a project “Workspace_Name” into Visual SourceSafe database. The **Project name** field shows “Poject_Name” that is to be created next.



Create SourceSafe Project

Project name: Project_Name

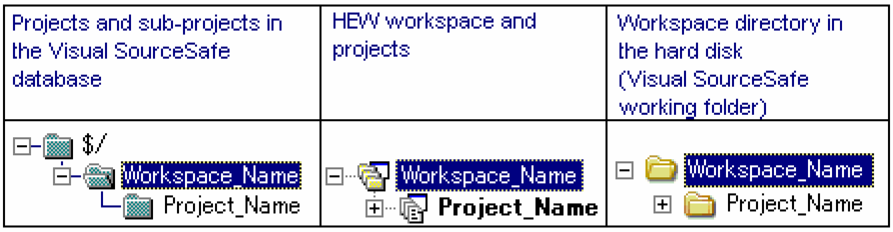
$/Workspace_Name/Project_Name

$

Workspace_Name

11. Click the OK button in the **Create SourceSafe Project** dialog box.
12. Repeat steps 10 and 11 for each of the projects in the current workspace.

The structure image of projects and sub-projects in the Visual SourceSafe database, HEW workspace and projects, and the workspace directory in the hard disk is shown below.



Projects and sub-projects in the Visual SourceSafe database	HEW workspace and projects	Workspace directory in the hard disk (Visual SourceSafe working folder)
$/ Workspace_Name Project_Name	Workspace_Name Project_Name	Workspace_Name Project_Name

The High-performance Embedded Workshop has now created the necessary projects within Visual SourceSafe, and set up the version control toolbar and menu for immediate access.

9.2 Visual SourceSafe commands

The following eight commands are available via the version control toolbar or menu:

Tools->Version Control menu	Version Control Toolbar	Function
Add to VCS		Adding files to Visual SourceSafe
Remove from VCS		Removing files from Visual SourceSafe
Get from VCS		Getting read-only copy of files from Visual SourceSafe
Check out from VCS		Checking out writable copy of files from Visual SourceSafe
Check in to VCS		Checking in writable copy of files into Visual SourceSafe
Get VCS status		Viewing the status of files in Visual SourceSafe
Undo Check Out	-	Undoing a check out operation in Visual SourceSafe
Show History	-	Viewing the history of files in Visual SourceSafe

9.2.1 Adding files to Visual SourceSafe

In the previous section, a mapping between the workspace directory on your hard disk (i.e. the working directory) and the project directory in Visual SourceSafe (i.e. the directory controlled by Visual SourceSafe) has been established. However, the project directory (and any subdirectories) on your hard disk may contain many source files, whereas the directory it is mapped to in Visual SourceSafe will be initially empty.

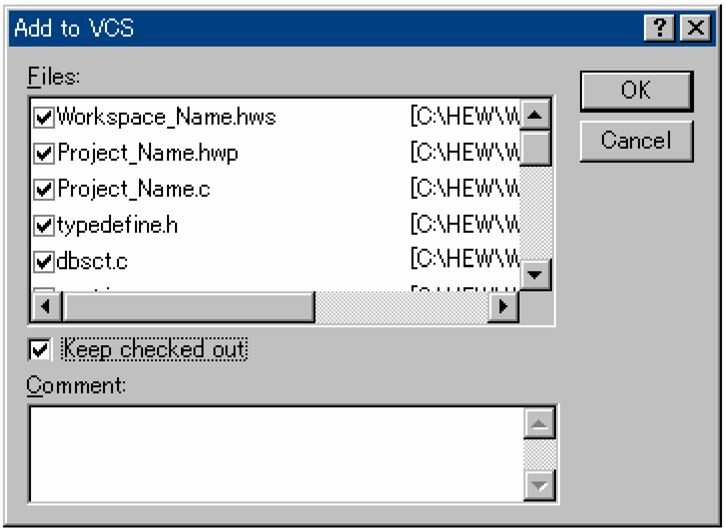
You need to add files to the Visual SourceSafe project.

To add files to Visual SourceSafe

1. Select the files that you would like to add to Visual SourceSafe, in the **Projects** tab of the workspace window. You may also select a file folder, project folder, workspace folder or combination thereof. When selecting the project or workspace folder, the HEW system files will be added to the selected file list. For example, selecting the project folder will also add the project file (.HWP) to the **Files** list. If the project file is then checked out and the version is newer than when it was last loaded, you will be asked whether you want to reload the project.
2. Select one of the following operations:

- Click the **Add to VCS** toolbar button (), **OR**
- Select the [**Tools->Version Control->Add to VCS**] menu option, **OR**
- Right-click to invoke a pop-up menu and select [**Version Control->Add to VCS**].

3. The **Add to VCS** dialog box opens. Select the checkbox on the left to the name of the file that you wish to add and click the OK button.



Add to VCS

Files:

☑ Workspace_Name.hws [C:\HEW\W
☑ Project_Name.hwp [C:\HEW\W
☑ Project_Name.c [C:\HEW\W
☑ typedefine.h [C:\HEW\W
☑ dbsct.c [C:\HEW\W

☑ Keep checked out

Comment:

When you add files to Visual SourceSafe, the local versions in your working directory will become read-only. If you select the **Keep checked out** checkbox in the **Add to VCS** dialog box before you check the files into Visual SourceSafe, these files can be writable even when they are added. To check that the **Add to VCS** operation was carried out as you expected, or to quickly review the status of all of the files in a project:

1. Select the project folder whose files you want to check, in the **Projects** tab of the workspace window.
2. Select one of the following operations:

- Click the **Get VCS status** toolbar button (), **OR**
- Select the [**Tools->Version Control->Get VCS status**] menu option, **OR**

- Right-click the selected item to invoke a pop-up menu and select [**Version Control->Get VCS status**].

3. The status of each file will be displayed in the **Version Control** tab of the output window, or the **File(s) Status** dialog box. For setting of location where the status is to be displayed, see section 9.3, Visual SourceSafe command options. The information shown includes whether the file is added to the project, if the file is checked out and, if it is checked out, and who did so.

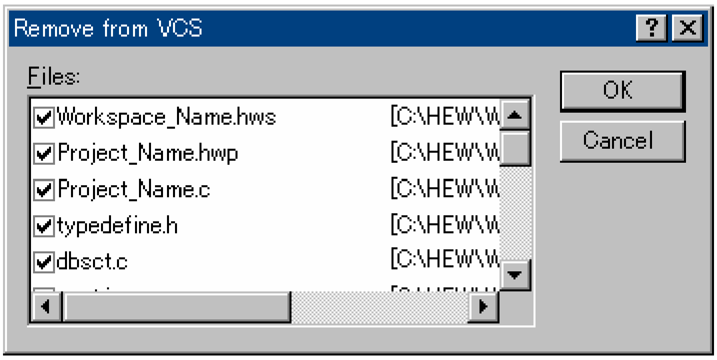
9.2.2 Removing files from Visual SourceSafe

If you wish to take files out of Visual SourceSafe, the files must be removed.

To remove files from Visual SourceSafe

1. Select the files that you would like to remove from Visual SourceSafe, in the **Projects** tab of the workspace window. You may also select a file folder, project folder, workspace folder or combination thereof.
2. Select one of the following operations:

- Click the **Remove from VCS** toolbar button (), **OR**
- Select the [**Tools->Version Control->Remove from VCS**] menu option, **OR**
- Right-click to invoke a pop-up menu and select [**Version Control->Remove from VCS**].



Remove from VCS

Files:

☑ Workspace_Name.hws [C:\HEW\W
☑ Project_Name.hwp [C:\HEW\W
☑ Project_Name.c [C:\HEW\W
☑ typedefine.h [C:\HEW\W
☑ dbsct.c [C:\HEW\W

3. The **Remove from VCS** dialog box opens. Select the checkbox on the left to the name of the file that you wish to remove and click the OK button.

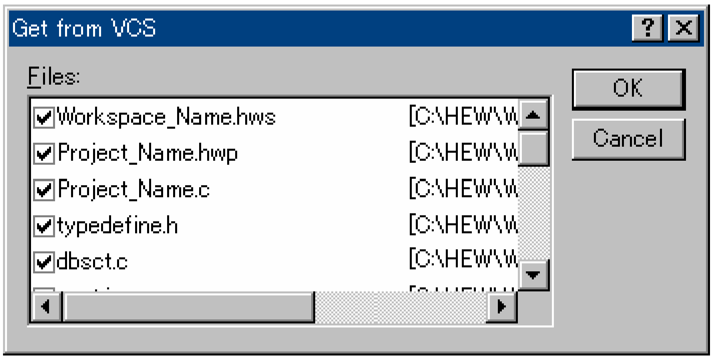
9.2.3 Getting read-only copy of files from Visual SourceSafe

Visual SourceSafe protects your source files and ensures that only one user can have a writable copy of a controlled file at any one time. However, it is possible for any user to obtain a read-only copy of any file.

To get read-only copy of files from Visual SourceSafe

1. Select the files that you would like to get from Visual SourceSafe, in the **Projects** tab of the workspace window. You may also select a file folder, project folder, workspace folder or combination thereof.
2. Select one of the following operations:

- Click the **Get from VCS** toolbar button (), **OR**
- Select the [**Tools->Version Control->Get from VCS**] menu option, **OR**
- Right-click to invoke a pop-up menu and select [**Version Control->Get from VCS**].


Get from VCS
Files:
☑ Workspace_Name.hws [C:\HEW\W
☑ Project_Name.hwp [C:\HEW\W
☑ Project_Name.c [C:\HEW\W
☑ typedefine.h [C:\HEW\W
☑ dbsct.c [C:\HEW\W
OK
Cancel

3. The **Get from VCS** dialog box opens. Select the checkbox on the left to the name of the file that you wish to obtain and click the OK button.

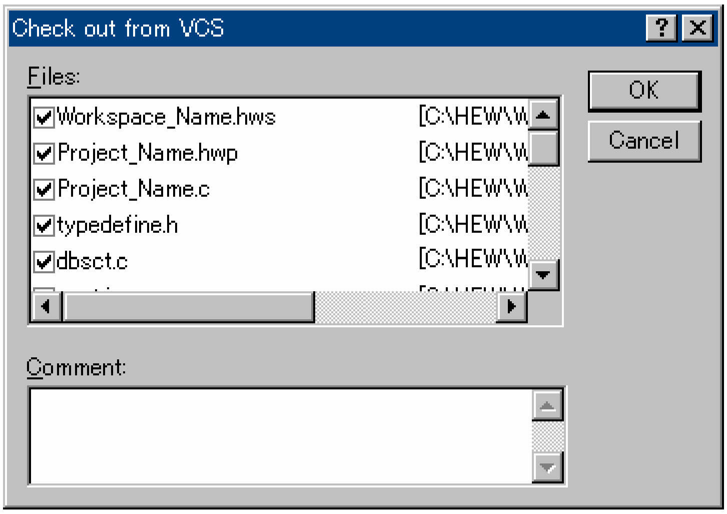
9.2.4 Checking out writable copy of files from Visual SourceSafe

Visual SourceSafe protects your source files and ensures that only one user can have a writable copy of a controlled file at any one time. The check out operation takes a writable copy of the file from Visual SourceSafe and places it on your local drive. This can only be done if another user has not already checked out (a blue mark next to file name) the files in question.

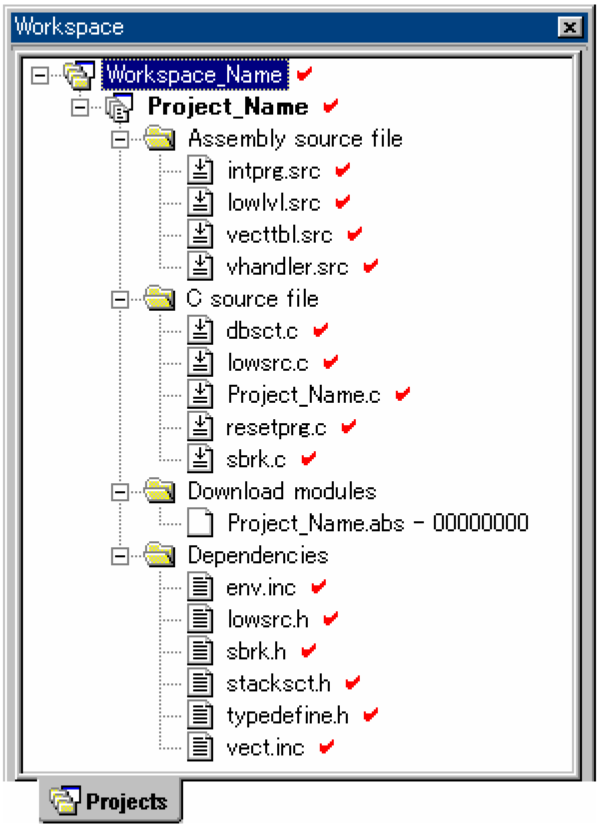
To check out writable copy of files from Visual SourceSafe

1. Select the files that you would like to check out from Visual SourceSafe, in the **Projects** tab of the workspace window. You may also select a file folder, project folder, workspace folder or combination thereof.
2. Select one of the following operations:

- Click the **Check out from VCS** toolbar button ( ), **OR**
- Select the [**Tools->Version Control->Check out from VCS**] menu option, **OR**
- Right-click to invoke a pop-up menu and select [**Version Control->Check out from VCS**].


Check out from VCS
Files:
☑ Workspace_Name.hws [C:\HEW\W
☑ Project_Name.hwp [C:\HEW\W
☑ Project_Name.c [C:\HEW\W
☑ typedefine.h [C:\HEW\W
☑ dbsct.c [C:\HEW\W
Comment:
OK
Cancel

3. The **Check out from VCS** dialog box opens. Select the checkbox on the left to the name of the file that you wish to obtain and click the OK button.
4. When the operation is finished the file has a red mark next to its name. This means you as the current user of HEW has checked it out.



Workspace

- Workspace_Name ✓
  - Project_Name ✓
    - Assembly source file
      - intprg.src ✓
      - lowlvl.src ✓
      - vecttbl.src ✓
      - vhandler.src ✓
    - C source file
      - dbsct.c ✓
      - lowsrc.c ✓
      - Project_Name.c ✓
      - resetprg.c ✓
      - sbrk.c ✓
    - Download modules
      - Project_Name.abs - 00000000
    - Dependencies
      - env.inc ✓
      - lowsrc.h ✓
      - sbrk.h ✓
      - stacksct.h ✓
      - typedefine.h ✓
      - vect.inc ✓

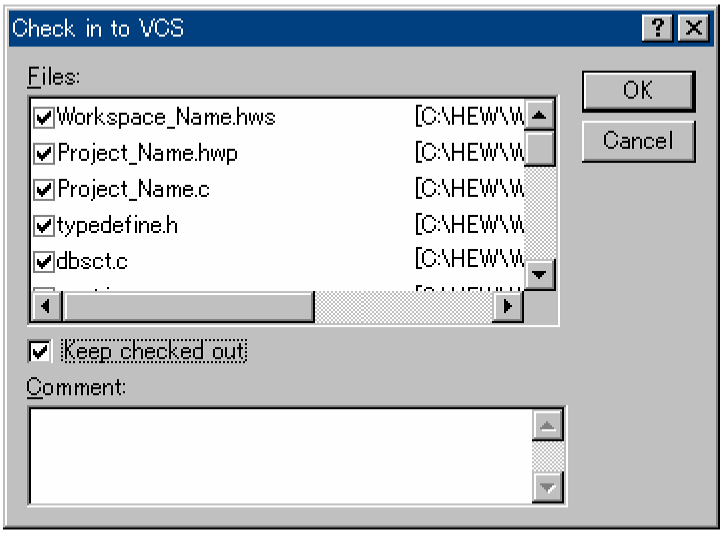
9.2.5 Checking in writable copy of files into Visual SourceSafe

Visual SourceSafe protects your source files and ensures that only one user can have a writable copy of a controlled file at any one time. The check out operation takes a writable copy of the file from Visual SourceSafe and places it on your local drive. Once a file is checked out it is edited and then checked back in so that the edits can be made available to other users.

To check in edits made to files in Visual SourceSafe

1. Select the files that you would like to check back into Visual SourceSafe, in the **Projects** tab of the workspace window. You may also select a file folder, project folder, workspace folder or combination thereof.
2. Select one of the following operations:

- Click the **Check in to VCS** toolbar button (), **OR**
- Select the [**Tools->Version Control->Check in to VCS**] menu option, **OR**
- Right-click to invoke a pop-up menu and select [**Version Control->Check in to VCS**].


Check in to VCS

Files:

☑ Workspace_Name.hws [C:\HEW\W

☑ Project_Name.hwp [C:\HEW\W

☑ Project_Name.c [C:\HEW\W

☑ typedefine.h [C:\HEW\W

☑ dbsct.c [C:\HEW\W

☑ Keep checked out

Comment:

3. The **Check in to VCS** dialog box opens. Select the checkbox on the left to the name of the file that you wish to check back into Visual SourceSafe and click the OK button.
4. When the operation is finished the file has a red mark next to its name. This means you as the current user of HEW has checked it out.
5. The red mark next to the file name is removed, which indicates that the file has been checked into Visual SourceSafe. If you select the **Keep checked out** checkbox in the **Check in to VCS** dialog box before you check the files into Visual SourceSafe, these files can be writable even when they are checked out.

9.2.6 Viewing the status of files in Visual SourceSafe

Although files appear in your HEW project (in the **Projects** tab of the Workspace window), Visual SourceSafe is not necessarily controlling them. Some of the files that are being controlled by Visual SourceSafe will be checked in, and others will be checked out (i.e. being edited by a user). The **Get VCS status** command displays the current status of files.

The status of each file will be displayed in the **Version Control** tab of the output window, or the **File(s) Status** dialog box. For setting of location where the status is to be displayed, see section 9.3, Visual SourceSafe command options.

To view the status of files in Visual SourceSafe

1. Select the files whose status you would like to view, in the **Projects** tab of the workspace window. You may also select a file folder, project folder, workspace folder or combination thereof.
2. Select one of the following operations:

- Click the **Get VCS status** toolbar button (), **OR**
- Select the [**Tools->Version Control->Get VCS status**] menu option, **OR**
- Right-click to invoke a pop-up menu and select [**Version Control->Get VCS status**].

3. The **Version Control** tab of the output window or the **File(s) Status** dialog box shows the status of files.

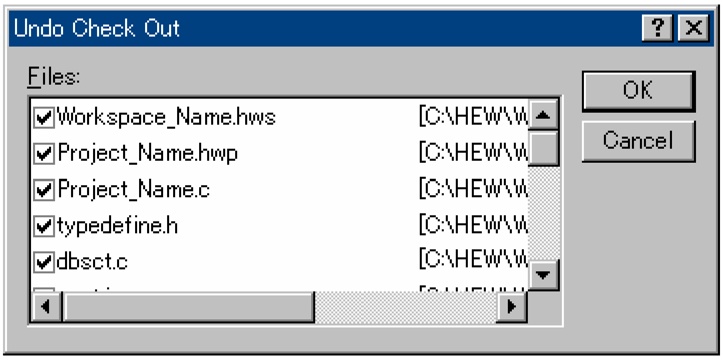
9.2.7 Undoing a check out command in Visual SourceSafe

Visual SourceSafe protects your source files and ensures that only one user can have a writable copy of a controlled file at any one time. The check out operation takes a writable copy of the file from Visual SourceSafe and places it on your local drive. Once a file is checked out it is edited and then checked back in so that the edits can be made available to other users. However, if the check out operation was carried out by mistake, or perhaps is no longer required, then the operation can be undone.

To undo a check out of files from Visual SourceSafe

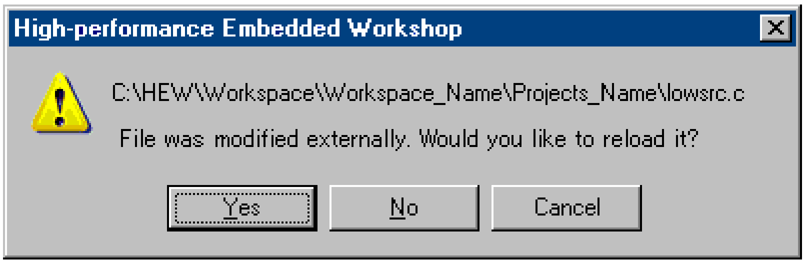
1. Select the files upon which you would like to undo a previous check out operation, in the **Projects** tab of the workspace window. You may also select a file folder, project folder, workspace folder or combination thereof.
2. Select one of the following operations:

- Select the [**Tools->Version Control->Undo Check Out**] menu option, **OR**
- Right-click to invoke a pop-up menu and select [**Version Control->Undo Check Out**].



☑Workspace_Name.hws [C:\HEW\W
☑Project_Name.hwp [C:\HEW\W
☑Project_Name.c [C:\HEW\W
☑typedefine.h [C:\HEW\W
☑dbsct.c [C:\HEW\W

3. The **Undo Check Out** dialog box opens. Select the checkbox on the left to the name of the file that you would like to undo a previous check out operation and click the OK button.
4. If the file has been modified, the confirmation message box shown below appears. To continue the check-out operation, click the Cancel button. To load the unmodified file from the version control system, click the Yes button. If you click the No button, the contents of the local file will be different from those of the file in the version control system.



C:\HEW\Workspace\Workspace_Name\Projects_Name\lowsrc.c

File was modified externally. Would you like to reload it?

9.2.8 Viewing the history of files in Visual SourceSafe

Visual SourceSafe controls the edits to the files in its projects and allows you to view the complete history of these edits right back to the time that the file was first added to the project.

The status of each file will be displayed in the **Version Control** tab of the output window, or the **File(s) History** dialog box. For setting of location where the status is to be displayed, see section 9.3, Visual SourceSafe command options.

To view the revision history of files in Visual SourceSafe

1. Select the files whose history you would like to view, in the **Projects** tab of the workspace window. You may also select a file folder, project folder, workspace folder or combination thereof.
2. Select one of the following operations:

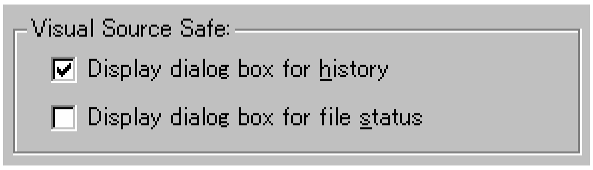
- Select the [**Tools->Version Control->Show History**] menu option, **OR**
- Right-click to invoke a pop-up menu and select [**Version Control->Show History**].

3. The **Version Control** tab of the output window or the **File(s) History** dialog box shows the revision history of files.

9.3 Visual SourceSafe command options

You can control the way in which the history and status commands are displayed by selecting the [**Tools->Version Control->Configure**] menu option.

1. Select the [**Tools->Version Control->Configure**] menu option. The **Version Control Setup** dialog box will be displayed.
2. Select the **General** tab.



Visual Source Safe:

☑ Display dialog box for history

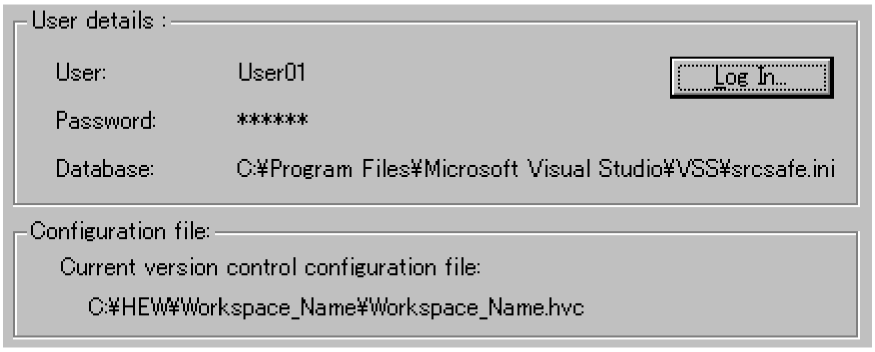
□ Display dialog box for file status

3. To display the results of a history command in a dialog box then check the **Display dialog box for history** checkbox or clear it if you would rather display the output in the **Version Control** tab of the Output window.
4. To display the results of a status command in a dialog box, check the **Display dialog box for file status** checkbox or clear it if you would rather display the output in the **Version Control** tab of the Output window.

9.4 Version Control setup

You can view or modify the general settings for the HEW connected with Visual SourceSafe.

1. Select [**Tools->Version Control->Configure**]. The **Version Control Setup** dialog box is displayed.
2. Select the **General** tab.



User details :

User: User01

Password: ******

Database: C:¥Program Files¥Microsoft Visual Studio¥VSS¥srcsafe.ini

Configuration file:

Current version control configuration file:

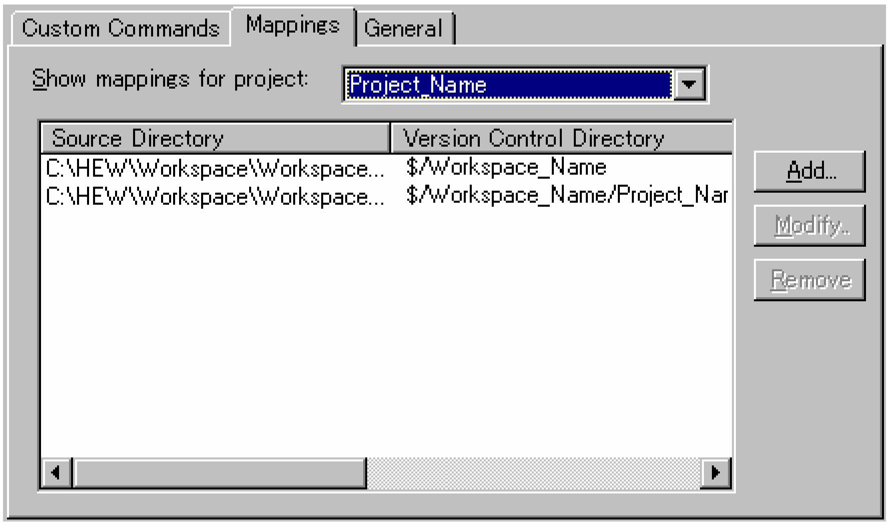
C:¥HEW¥Workspace_Name¥Workspace_Name.hvc

3. The **User details** group box shows the login user ID and password, and database file in Visual SourceSafe.
4. You can modify the user ID, password, or database in the **Log in** dialog box opened by clicking the **Log In** button.
5. The **Configuration file** group box shows the configuration file in the current version control system.

9.5 Specifying file locations

You can view the current mapping in the project. Select the project that you wish to view the mapping from the **Show mappings for project** drop-down list. The HEW workspace and project directory, and Visual SourceSafe project directory are shown in the list. Mapping of projects can be added, modified, or removed.

For detail, see section 8.6, Specifying file locations.



Show mappings for project: Project_Name

Source Directory	Version Control Directory
C:\HEW\Workspace\Workspace...	$/Workspace_Name
C:\HEW\Workspace\Workspace...	$/Workspace_Name/Project_Nar

To define a new directory mapping

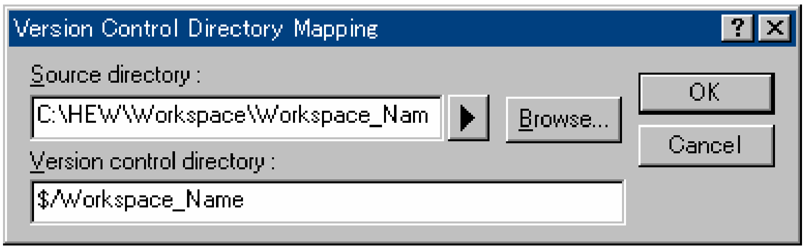
1. Select [**Tools->Version Control->Configure**]. The **Version Control Setup** dialog box will be displayed.
2. Select the **Mapping** tab.
3. Click the **Add** button. The **Version Control Directory Mapping** dialog box will be displayed.
4. Specify a HEW workspace directory or project directory in the **Source directory** field. Select one of the following operations:

- Enter the directory name, **OR**
- Click the placeholder button. Then select "Project directory" or "Workspace directory" from the menu, **OR**
- Click the **Browse** button to open the **Browse to Source Directory** dialog box. Select the directory and click the **Select** button.

5. Enter the name of the Visual SourceSafe project directory to be mapped to the source directory in the **Version Control Directory Mapping** field.
6. Click the OK button.

To modify a directory mapping

1. Select [**Tools->Version Control->Configure**]. The **Version Control Setup** dialog box will be displayed.
2. Select the **Mapping** tab.
3. Select the mapping to be modified from the list and then click the **Modify** button. The **Version Control Directory Mapping** dialog box will be displayed.


Version Control Directory Mapping

Source directory : C:\HEW\Workspace\Workspace_Nam

Version control directory : $/Workspace_Name

4. To modify a mapping, make the same setting as required for adding a mapping in the Version Control Directory Mapping dialog box.
5. Click the OK button.

To remove a directory mapping

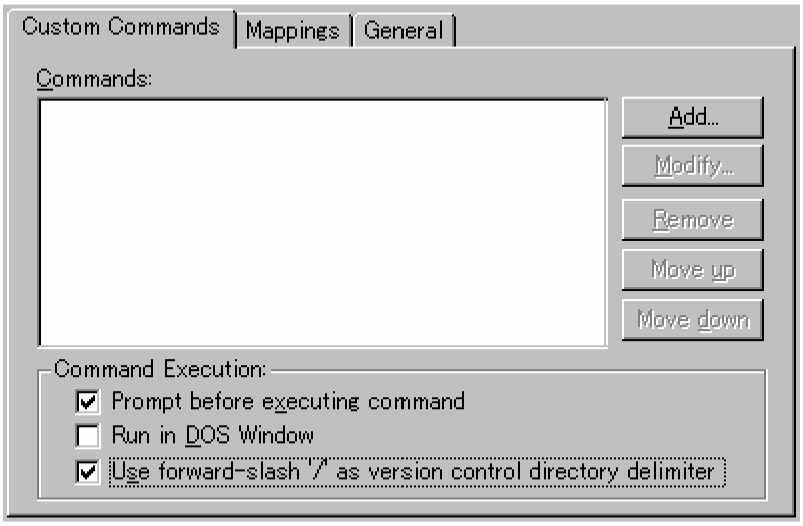
1. Select [**Tools->Version Control->Configure**]. The **Version Control Setup** dialog box will be displayed.
2. Select the **Mapping** tab.
3. Select the mapping to be removed from the list and then click the **Remove** button.

9.6 Adding Visual SourceSafe commands

In the **Custom Commands** tab of the **Version Control Setup** dialog box, you can add a menu defining a Visual SourceSafe command not included in the version control submenu.

Selecting this menu executes the defined command. Output information on execution of this command will be shown in the **Version Control** tab of the Output window.

This section introduces the definition of Properties command as an example.



Command Execution:

☑ Prompt before executing command

□ Run in DOS Window

☑ Use forward-slash '/' as version control directory delimiter

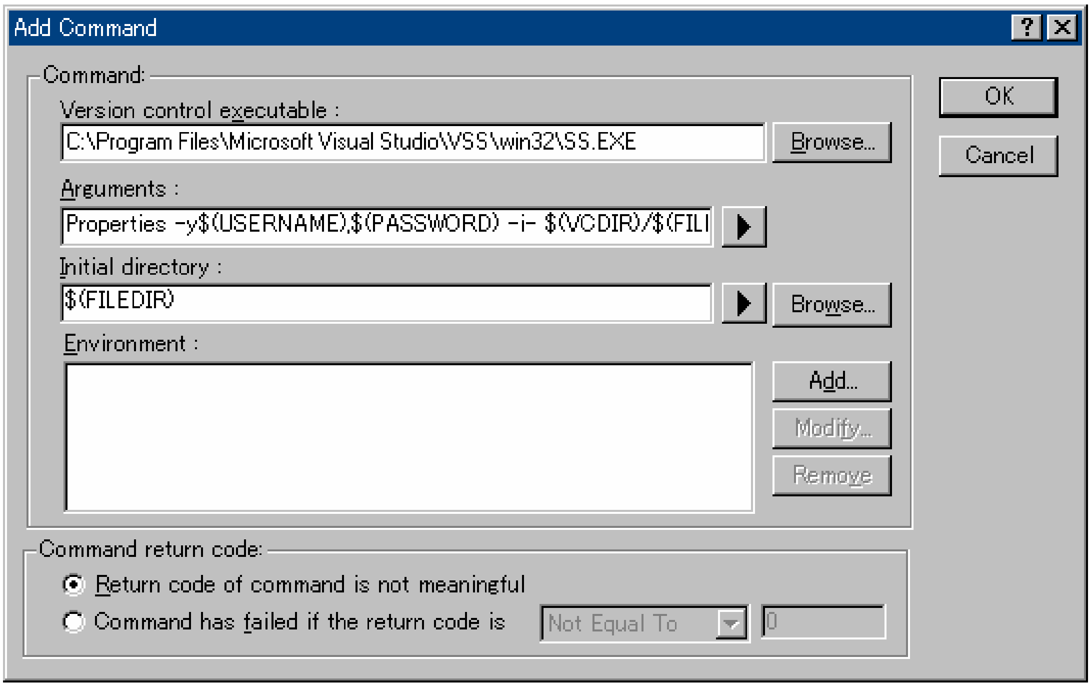
- **Execution control of version control commands**

1. Select the **Prompt before executing command** checkbox.
2. Deselect the **Run in DOS Window** checkbox. (This is because the RCS commands must be input via the MS-DOS prompt.)
3. Select the **User forward-slash '/' as version control directory delimiter** checkbox. (Visual SourceSafe uses this format.)

For detail, see section 8.9, Controlling execution of a Version Control System.

- **Setting up commands**

1. Click the **Add** button. The **Add Menu Option** dialog box appears.
2. Enter "Properties" in **Option** and click the **Add** button. The **Add Command** dialog box appears.



Version control executable : C:\Program Files\Microsoft Visual Studio\VSS\win32\SS.EXE

Arguments : Properties -y$(USERNAME),$(PASSWORD) -i- $(VCDIR)/$(FILI

Initial directory : $(FILEDIR)

Command return code:

● Return code of command is not meaningful

○ Command has failed if the return code is Not Equal To 0

3. Enter the executable file (SS.EXE) for version control system in **Version control executable**.
4. Enter "Properties -y$(USERNAME),$(PASSWORD) -i- $(VCDIR)/$(FILENAME)" in **Arguments**.
5. Enter "$(FILEDIR)" in **Initial directory**.
6. Click the OK button.
7. Click the OK button in the **Add Menu Option** dialog box.

For detail, see section 8.2, Defining Version Control commands.

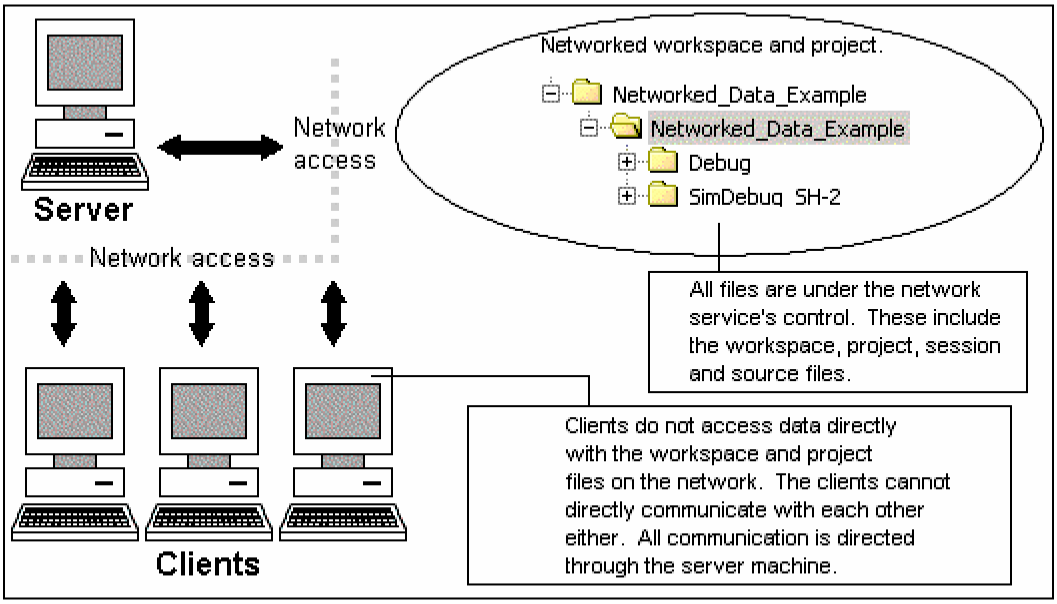
- **Command Execution**

1. Select the [**Tools->Version Control->Properties**] menu option.
2. Select the file in the Properties dialog box and click the OK button.
3. Output information on execution of this command will be shown in the **Version Control** tab of the Output window.

10. Sharing Projects by Network Facilities

The High-performance Embedded Workshop (HEW) is capable of sharing workspaces and projects across a network. This allows users to concurrently work on shared projects and see each other's changes as they happen. This system can be used in conjunction with version control. The major difference with using this system is that each user can modify and update the workspace and project without making all of the other users reload their project and potentially lose all their changes.

This system is implemented by making one of the machines attached to the network the server machine. All other client machines then use the service this machine is providing. So if one of the client machines adds a new file, the server machine is notified. The server then notifies all other clients the action has taken place. This procedure is shown below.



All files are under the network service's control. These include the workspace, project, session and source files.

Clients do not access data directly with the workspace and project files on the network. The clients cannot directly communicate with each other either. All communication is directed through the server machine.

The network system allows users to be given access rights to files. This allows the project administrator to make sure the only people who can modify the project and source files are allowed to do so. This might allow the administrator to limit each user to only have write capabilities for their own area of the project, other areas would be read only. This could limit any potential conflict or damage one user could do to other areas of the project. These limitations can be set to a number of different levels. This is outlined later in this section.

Note:

- Certain operations are locked when other clients are carrying them out. This means that if one user is currently changing the toolchain options all of the other users can only see read only versions of this data.
- The performance of HEW does suffer when using the network facilities. If working in a small team it might be more suitable to use the single user mode and version control.

Procedures necessary before invoking the HEW on a PC with Windows XP Service Pack 2

See the following section, 10.1, Sharing projects by network facilities in a PC with Windows XP Service Pack 2.

To setup a project file on the server machine

1. Create a project file for the network.

2. As the directory of the New Project Workspace dialog, specify a network path that can be recognized by both the server machine and the client machine.
   Example: \\WorkGroup\Severl\ShareDirectory

Sharing workspaces

The directory specified by the above procedure (step 2) must be shared. The HEW's network facilities share the source, object, and other files in the server machine. These files exist only in the server machine. From the client machine, the files are accessed via the network.

To open a workspace from the client machine

To open a workspace on the network from the client machine, search the network path and select the workspace in the Open Workspace dialog.

10.1 Sharing projects by network facilities in a PC with Windows XP Service Pack 2

Windows XP Service Pack 2 provides enhanced security measures.

In a PC with Windows XP Service Pack 2, you cannot share projects by network facilities if the OS is in its default state. Before invoking the HEW, you should take the procedures listed below to change the settings of the PC so that HEW system will be accessed via network. When your server machine and client machine have Windows XP Service Pack 2, the settings are required for both machines.

- **For a server machine with Windows XP Service Pack 2**

Step1: Firewall

1. Select **Control Panel** from the Windows start menu.
2. Click **Network and Internet Connections i**n the **Control Panel**.
3. Click **Windows Firewall** in the **Network and Internet Connections**.
4. The **Windows Firewall** dialog box is invoked.
5. Click the **Exceptions** tab.
6. Click the **Add Program** button to open the **Add a Program** dialog box.
7. Click the **Browse** button and select **\System\Sec\HewServer\Hew3Server.exe** under the HEW installation directory. Click the OK button and close the **Add a Program** dialog box.
8. Click the **Add Port** button to open the **Add a Port** dialog box.
9. Enter "DCOM" in **Name** and "135" in **Port Number** and select **TCP** option button.
10. Click the OK button and close the **Add a Port** dialog box.
11. Click the OK button in the **Windows Firewall** dialog box.
12. Close the **Network and Internet Connections**.
13. Close the **Control Panel**.

Step2: DCOM

1. Select **Run** from the Windows start menu.
2. The **Run** dialog box is invoked.
3. Enter "dcomcnfg" in the **Open** field and click the OK button.
4. The **Component Services** window opens.
5. Expand the tree. If the **Window Security Alert** dialog box appears at expansion, click the **Keep Blocking** button.

   Console Root -> Component Services -> Computers -> My Computer

6. Right-click on **My Computer** and select **Properties** from the pop-up menu.
7. The **My Computer Properties** dialog box is invoked.
8. Click **COM Security** tab.
9. Click the **Edit Limits** button in the **Access Permissions** area to open the **Access Permission** dialog box.
10. Select "ANONYMOUS LOGON" in the **Group or user names** list.
11. Select the **Allow** checkbox for **Remote Activation** in the **Permission for ANONYMOUS LOGON** list.
12. Click the OK button and close the **Access Permission** dialog box.
13. Click the **Edit Limits** button in the **Launch and Activation Permission** area to open the **Launch Permission** dialog box.
14. Select "Everyone" in the **Group or user names** list.
15. Select the **Allow** checkbox for **Local Launch** in the **Permission for Everyone** list. Select the **Allow** checkbox for **Remote Activation** in the **Permission for Everyone** list.
16. Click the OK button and close the **Launch Permission** dialog box.
17. Click the OK button on the **My Computer Properties** dialog box.
18. Close the **Component Services** window.

- **For a client machine with Windows XP Service Pack 2**

Step1: Firewall

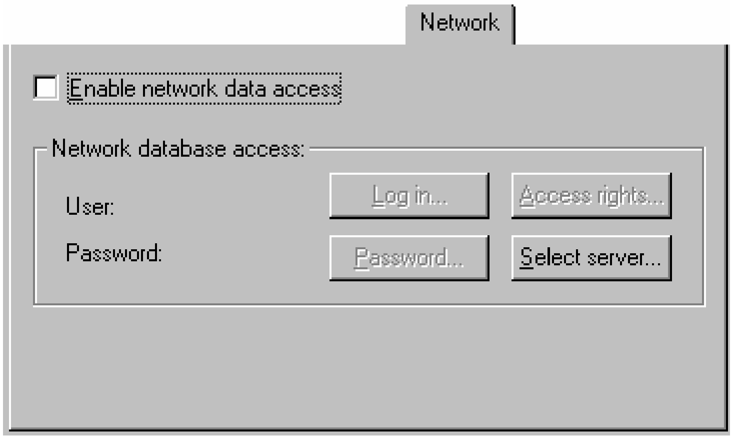
1. Select **Control Panel** from the Windows start menu.
2. Click **Network and Internet Connections i**n the **Control Panel**.
3. Click **Windows Firewall** in the **Network and Internet Connections**.
4. The **Windows Firewall** dialog box is invoked.
5. Click the **Exceptions** tab.
6. Click the **Add Program** button to open the **Add a Program** dialog box.
7. Click the **Browse** button and select **Hew2.exe** under the HEW installation directory. Click the OK button and close the **Add a Program** dialog box.
8. Click the **Add Port** button to open the **Add a Port** dialog box.
9. Enter "DCOM" in **Name** and "135" in **Port Number** and select **TCP** option button.
10. Click the OK button and close the **Add a Port** dialog box.
11. Click the OK button in the **Windows Firewall** dialog box.
12. Close the **Network and Internet Connections**.
13. Close the **Control Panel**.

10.2 Enabling network facilities to share projects

To use network facilities to share projects

1. Select [**Setup->Options**]. The **Options** dialog box is displayed.
2. Select the **Network** tab.
3. Click the **Enable network data access** checkbox. This should add an administrator to the system without a password. The administrator is the only user that can add additional users to the system and change user access rights. The administrator has the highest level of access.
4. Before leaving the network dialog the administrator must set their password. It is not possible to leave this dialog box until this is completed. This is described in the following section.
5. Close the **Options** dialog box.

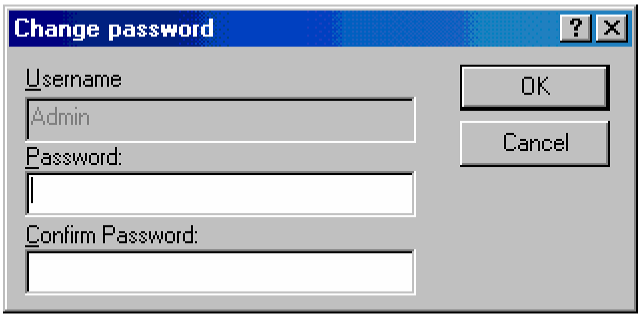
6. When the dialog box is closed you are asked if you want to save the workspace and then re-open it. This is because the workspace must be re-opened in the shared access mode. If the changes are not saved then they will be lost. Click OK to share projects by network facilities.
7. When you re-open the workspace and log into the system, a dialog is displayed showing your current access rights. For example if you are the admin user the level will be administrator. When this dialog is closed the HEW server window is opened and the network facilities to share projects are enabled.



10.3 Setting the administrator user's password

To set the administrator users password

1. Continue from the previous sections (step 4).
2. Click the **Password** button. This should have been enabled when the network data access was enabled.
3. The **Change password** dialog box is displayed.



Username: Admin

4. The user name is read only in the top field. In this case it should be Admin.
5. Type the new password into both of the fields and click OK.
6. This should set the user and password on the **Network** tab of **Options** dialog box.

10.4 Adding new users to the system

The initial setting of the network database adds an administrator user and a guest user to the system. The following levels of access are possible in the HEW system:

Administrator	Full access to every aspect of HEW. The user can add and remove users from the projects and change access rights. The administration user can change the workspace and project files and also the source files.
Full read/write access	The workspace and project files can be modified, as can the source files. But it is not possible to change user access rights from this access level.

Read/write file access — Only the source files can be modified. All project settings can only be viewed not modified.

Read only — All source files and project files can only be viewed as read only. Nothing can be modified.

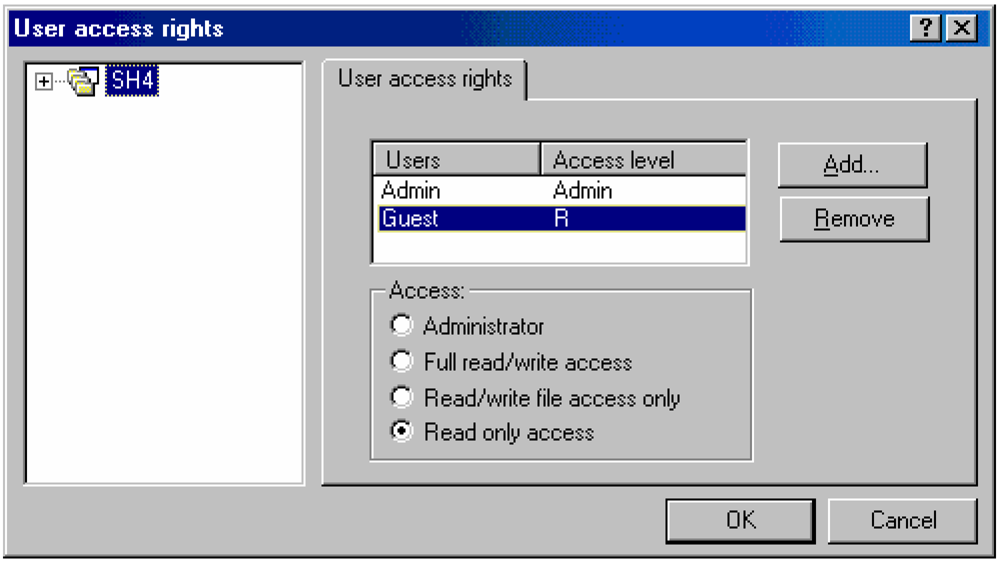
When any user opens a network-enabled project they must type in their user name and password. Until this is done no access can be granted. Once entered the user is given one of the levels of access as seen above.

To add a new user to the system

1. Log in with a user who has administrator access rights. The process for doing this is described above.
2. Select [**Setup->Options**]. The **Options** dialog box is displayed.
3. Select the **Network** tab.
4. Click the **Access rights** button. The **User access rights** dialog is displayed.
5. Click the **Add** button. The **Add new user** dialog box is displayed. This allows you as the administration user to add a new log in name and password. Normally the password should be set to some default text or left blank. Then click OK.
6. Once OK is clicked the user is added with read only rights. To change the access level select the user you wish to modify and then click the required radio button. Then click OK to save the access rights changes.

To remove an existing user to the system

1. Log in with a user who has administrator access rights. The process for doing this is described above.
2. Select [**Setup->Options**]. The **Options** dialog box is displayed.
3. Select the **Network** tab.
4. Click the **Access rights** button. The **User access rights** dialog is displayed.
5. Select the user you wish to remove in the users list.
6. Press the **Remove** button.
7. Then click OK to save the access rights changes.



User access rights

SH4

User access rights

Users	Access level
Admin	Admin
Guest	R

Access:

○ Administrator
○ Full read/write access
○ Read/write file access only
● Read only access

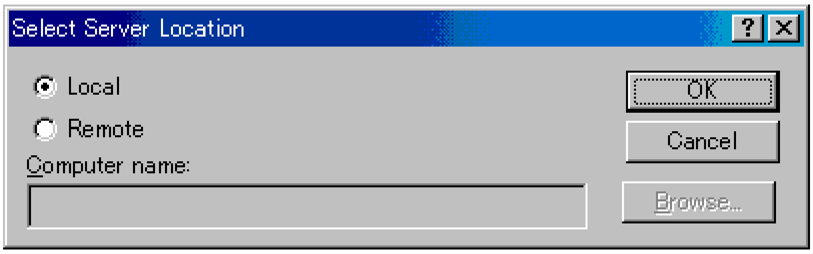
10.5 Changing your password

To change your password

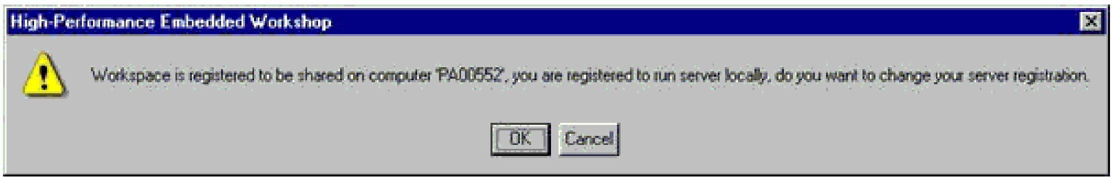
1. Log into the HEW network database you are changing your password for. Select [**Setup->Options**]. The **Options** dialog box is displayed.
2. Select the **Network** tab.
3. Click the **Password** button.
4. Enter your new password and confirm it in the second edit box.
5. Click OK.
6. Then click OK to save the password change.

10.6 Note on using the network facilities to share projects

When you share a networked project for the first time the HEW automatically connects you to the correct network HEW service. This is defined using machine name. If the workspace cannot be found using the machine name then the dialog below appears. Simply type or browse the location of the machine and click OK. If you want to be the server machine then leave the radio button on its default selection, use local machine.



If you have previously been the server of a workspace then the following message will be displayed when you attempt connection to another machine. Clicking OK then connects your machine to the new location.



Workspace is registered to be shared on computer 'PA00552', you are registered to run server locally, do you want to change your server registration.

Note:

If the network is running multiple HEW workspaces with the network facilities to share projects enabled then a user can only access one of them at one time. The only instance when this is not the case is if the same machine is serving all of the network workspaces.

11. Comparing Files

The High-performance Embedded Workshop (HEW) has an integrated difference view. You can perform a difference comparison with two files on your local drive or a local file and a file in the Microsoft® Visual SourceSafe system.

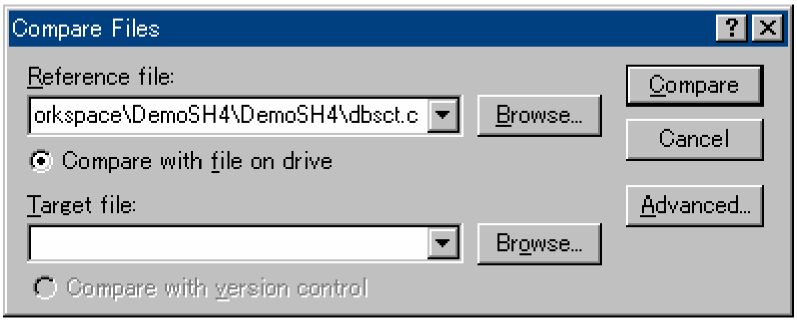
11.1 Opening the Difference window

To perform a difference comparison with two files on your local drive

1. Select one of the following operations:

- Select a file in the Projects tab of the workspace window. Right-click on the selected file to open a pop-up menu. Select **Show Differences**, **OR**
- Click the **Compare Files** toolbar button (), **OR**
- Select the [**View->Differences**] menu. Right-click within the window to open a pop-up menu. Select [**Compare**].

2. The **Compare Files** dialog box is displayed.



Compare Files

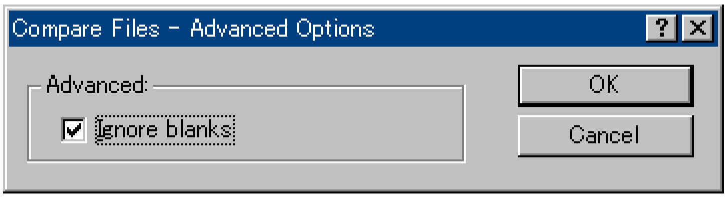
Reference file: orkspace\DemoSH4\DemoSH4\dbsct.c

● Compare with file on drive

Target file:

○ Compare with version control

3. Ensure the **Compare with file on drive** radio button is enabled.
4. Enter the name of a file to be compared to in **Reference file**. If you have opened the **Compare Files** dialog box via the workspace window at step 1, the file name is already entered in **Reference file**. Select a previously used file from the drop-down list box or click the **Browse** button to browse a file.
5. Clicking the **Advanced** button displays the **Compare Files - Advanced Options** dialog box. This allows you to perform the difference comparison without taking white space into account. Click OK when you are finished with this options dialog box.



Compare Files - Advanced Options

Advanced:

☑ Ignore blanks

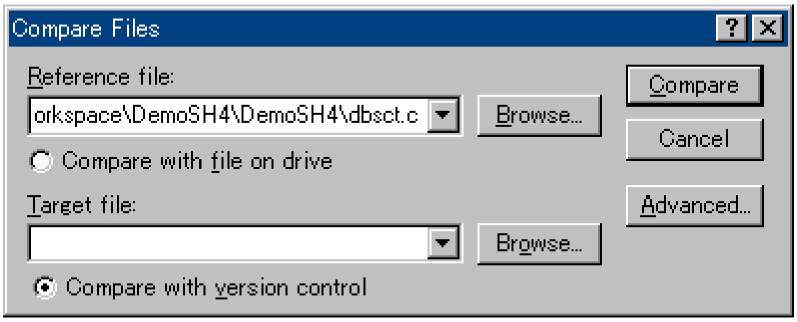
6. Click **Compare**.

To perform a difference comparison with a local file and a file in Visual SourceSafe system

1. The HEW must be connected to the Visual SourceSafe system. Add the two files you wish to compare to the Visual SourceSafe system in advance.
2. Select one of the following operations:

- Select a file in the Projects tab of the workspace window. Right-click on the selected file to open a pop-up menu. Select **Show Differences**, **OR**
- Click the **Compare Files** toolbar button (), **OR**
- Select the [**View->Differences**] menu. Right-click within the window to open a pop-up menu. Select **Compare**.

3. The **Compare Files** dialog box is displayed.



Reference file: orkspace\DemoSH4\DemoSH4\dbsct.c

○ Compare with file on drive

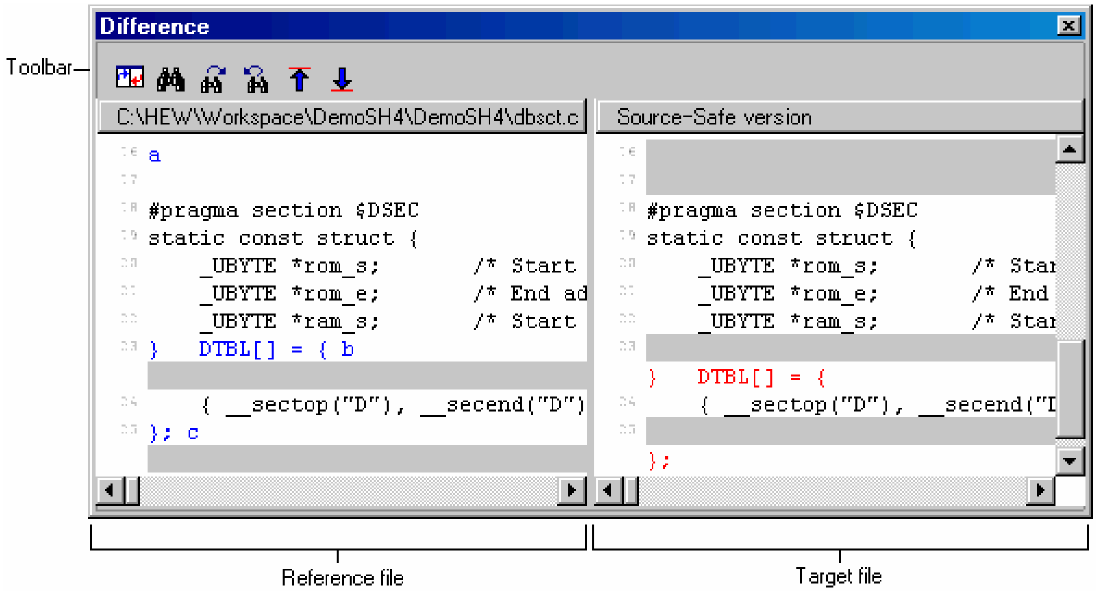
Target file:

● Compare with version control

4. Click the **Compare with version control** radio button. This radio button is selectable if the HEW has been connected to the Visual SourceSafe system.
5. Enter the name of a file to be compared to in **Reference file**. If you have opened the **Compare Files** dialog box via the workspace window at step 2, the file name is already entered in **Reference file**. Select a previously used file from the drop-down list box or click the **Browse** button to browse a file.
6. Clicking the **Advanced** button displays the **Compare Files - Advanced Options** dialog box. This allows you to perform the difference comparison without taking white space into account. Click OK when you are finished with this options dialog.
7. Click **Compare**.

The **Difference** window is displayed.

Window configuration



```
a

#pragma section $DSEC
static const struct {
    _UBYTE *rom_s;       /* Start
    _UBYTE *rom_e;       /* End ad
    _UBYTE *ram_s;       /* Start
}   DTBL[] = { b

    { __sectop("D"), __secend("D")
}; c
```

```

#pragma section $DSEC
static const struct {
    _UBYTE *rom_s;       /* Star
    _UBYTE *rom_e;       /* End
    _UBYTE *ram_s;       /* Star

}   DTBL[] = {
    { __sectop("D"), __secend("I

};
```

- In the Difference window, the left and right panes respectively show the files to be compared.
- Their names are at the top of each pain.
- If you hover the mouse cursor on the boundary of the two panes, the cursor turns into a double-sided arrow. To adjust the widths of the panes, drag the mouse to a desirable position.

Option

Right-clicking displays a pop-up menu containing available options.

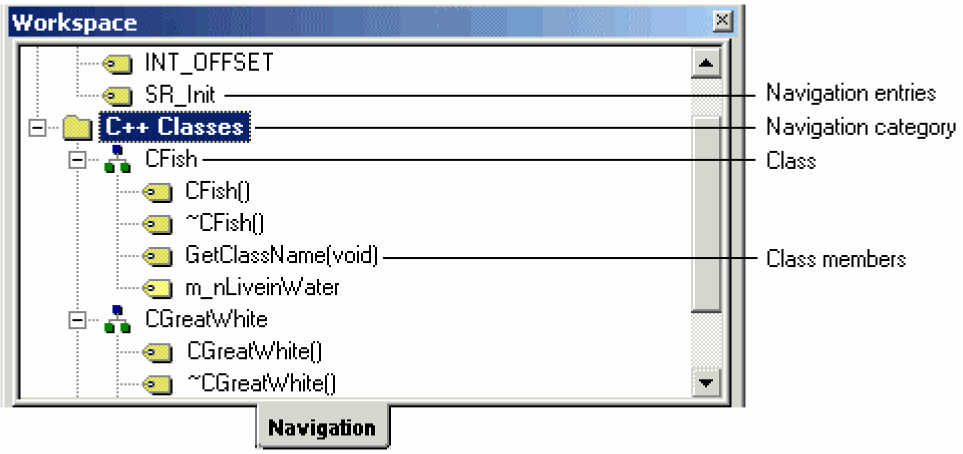
A basic operation is allocated to the toolbar.

The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up menu options	Toolbar button	Macro Recording	Function
Compare	⧉	-	This opens a new compare window so that some new files can be compared and the differences displayed.
Export results to file	-	-	This opens a dialog which allows you to choose a file to export the current difference results to a textual format.
Ignore white space	☒	-	The ignore white space option which is on the advanced options dialog can be toggled via this menu item.
Find	🔍	-	Displays a standard find dialog. This uses the same find dialog as the HEW editor.
Find previous	🔍	-	Finds the next previous string that meets the find requirements.
Find next	🔍	-	Finds the next string that meets the find requirements.
Previous difference	↑	-	Automatically jumps the view to the next previous difference.
Next difference	↓	-	Automatically jumps the view to the next difference.
Refresh comparison	-	-	Refreshes the view to manually run the difference comparison again. This can be used if either file has been modified since the last comparison.
Toolbar display	-	-	Shows or hides the toolbar
Customize toolbar	-	-	Customizes toolbar buttons.

12. Navigation Facilities

The High-performance Embedded Workshop (HEW) has a number of new integrated navigation facilities.



The **Navigation** tab of the workspace window contains categories for all supported navigation types. In HEW the following navigation components are supported as standard:

Navigation Type (Category)	Function
C Defines	All #defines for C and C++ source files are displayed.
C Functions	All ANSI C standard functions are for C source files displayed.
C++ Class	All classes, functions and members are displayed for C++ source files.

By default, it is possible to display the navigation items grouped by a navigation category.

Underneath each category, the navigation items belonging to the active project are displayed in the alphabetical order.

To group the display of navigation items by a navigation category

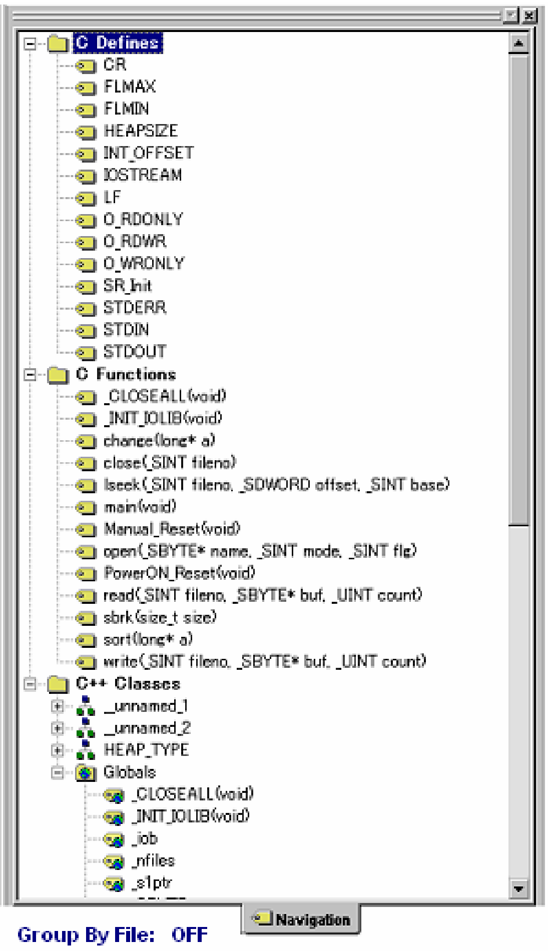
1. If you right-click anywhere inside the **Navigation** tab, a pop-up menu will be invoked.
2. Un-check **Group By File** menu item. This option is unchecked by default.

The Navigation view lets you see the navigation items on a file-by-file basis.

Each file in the active project is shown in the tree, and the navigation items belonging to each file is displayed below it in the alphabetical order.

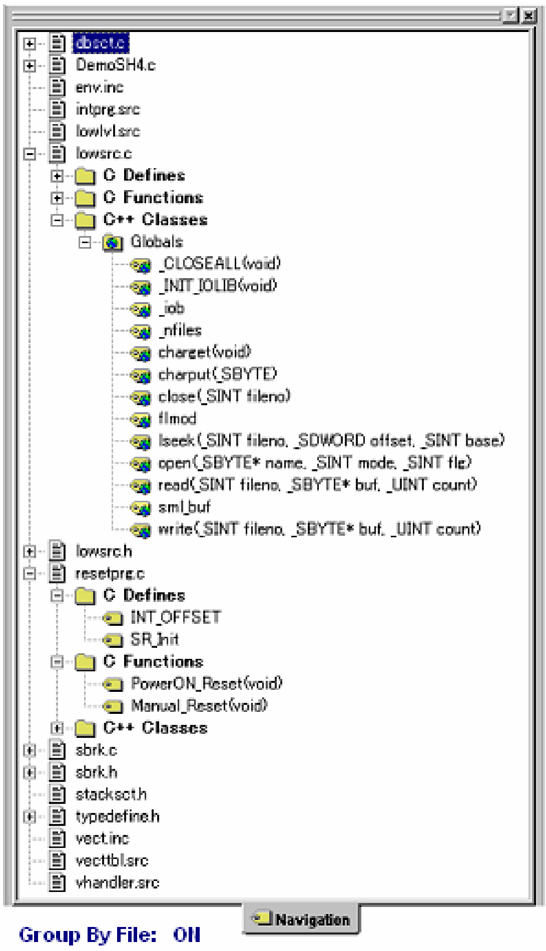
To group the display of navigation items by a file

1. If you right-click anywhere inside the **Navigation** tab, a pop-up menu will be invoked.
2. Check **Group By File** menu item.



```
C Defines
    CR
    FLMAX
    FLMIN
    HEAPSIZE
    INT_OFFSET
    IOSTREAM
    LF
    O_RDONLY
    O_RDWR
    O_WRONLY
    SR_Init
    STDERR
    STDIN
    STDOUT
C Functions
    _CLOSEALL(void)
    _INIT_IOLIB(void)
    change(long* a)
    close(_SINT fileno)
    lseek(_SINT fileno, _SDWORD offset, _SINT base)
    main(void)
    Manual_Reset(void)
    open(_SBYTE* name, _SINT mode, _SINT flg)
    PowerON_Reset(void)
    read(_SINT fileno, _SBYTE* buf, _UINT count)
    sbrk(size_t size)
    sort(long* a)
    write(_SINT fileno, _SBYTE* buf, _UINT count)
C++ Classes
    _unnamed_1
    _unnamed_2
    HEAP_TYPE
    Globals
        _CLOSEALL(void)
        _INIT_IOLIB(void)
        _iob
        _nfiles
        _s1ptr
```

Group By File: OFF



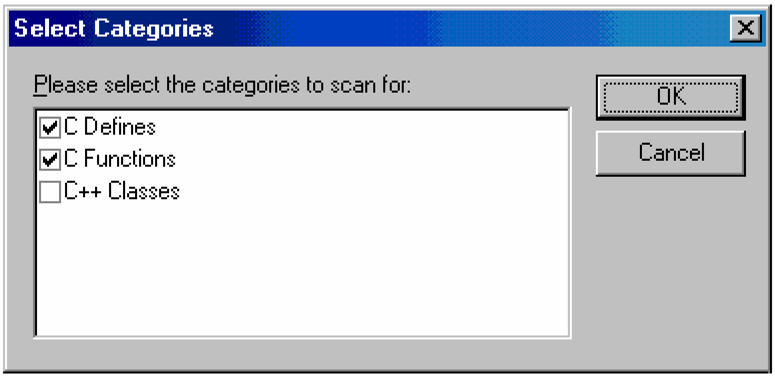
```
dbsct.c
DemoSH4.c
env.inc
intprg.src
lowlvl.src
lowsrc.c
    C Defines
    C Functions
    C++ Classes
        Globals
            _CLOSEALL(void)
            _INIT_IOLIB(void)
            _iob
            _nfiles
            charget(void)
            charput(_SBYTE)
            close(_SINT fileno)
            flmod
            lseek(_SINT fileno, _SDWORD offset, _SINT base)
            open(_SBYTE* name, _SINT mode, _SINT flg)
            read(_SINT fileno, _SBYTE* buf, _UINT count)
            sml_buf
            write(_SINT fileno, _SBYTE* buf, _UINT count)
lowsrc.h
resetprg.c
    C Defines
        INT_OFFSET
        SR_Init
    C Functions
        PowerON_Reset(void)
        Manual_Reset(void)
    C++ Classes
sbrk.c
sbrk.h
stacksct.h
typedefine.h
vect.inc
vecttbl.src
vhandler.src
```

Group By File: ON

It is possible to disable scanning for certain navigation categories if you do not require the information.

To switch off a navigation category

1. If you right-click anywhere inside the **Navigation** tab, a pop-up menu will be invoked.
2. Select the **Select Categories** menu item. The **Select Categories** dialog box is displayed.



Select Categories

Please select the categories to scan for:

☑ C Defines
☑ C Functions
□ C++ Classes

3. Un-check any categories you are not interested in seeing definitions for.
4. Click **OK**.

To update the navigation view

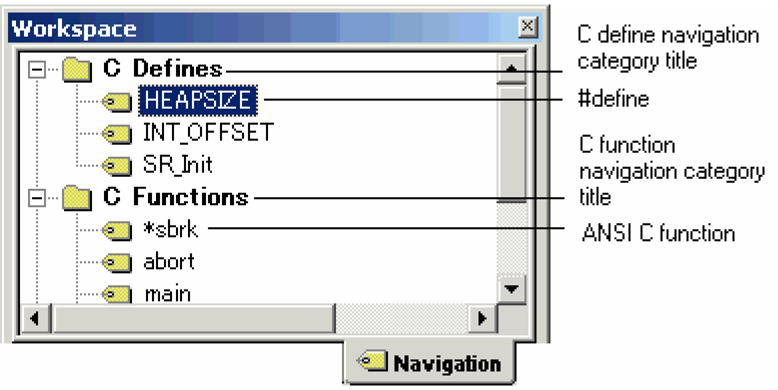
1. If you right-click anywhere inside the **Navigation** tab, a pop-up menu will be invoked.
2. Select **Refresh**.

Note:

- The navigation items are displayed gradually as the files are scanned. This means it may take some time if there are many files to fully complete the Navigation view update.
- Files are re-scanned when they are saved. This means that navigation information will not be available for new classes and functions until the file or files are saved.
- **Group By File** and **Group By Access** cannot both be on at the same time. Switching one on will switch the other one off.

12.1 C function and #define navigation component

These components simply add the function and #define definitions to the navigation view.



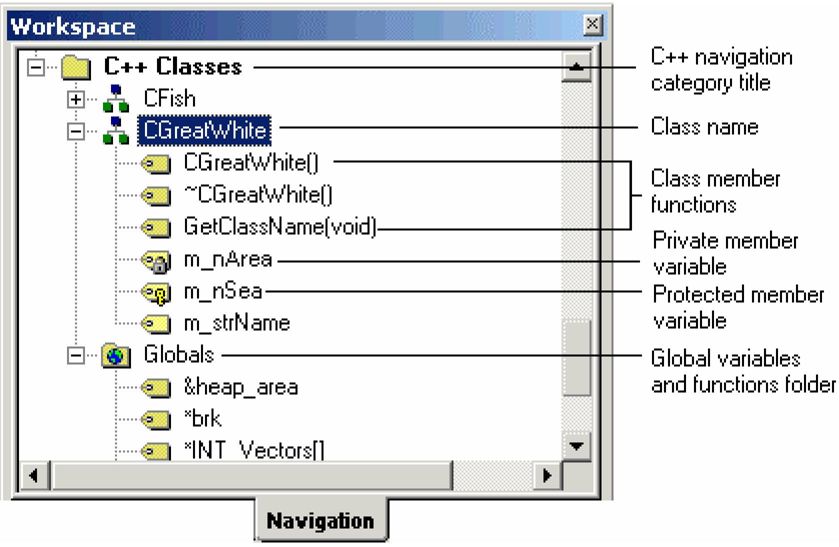
To jump to a definition

Select either of the following ways.

- Double-click a function or a #define definition on the **Navigation** tab.
- Right-click on a function or a #define definition on the **Navigation** tab. Select **Go to Definition** from the pop-up menu.

12.2 C++ navigation component

The C++ navigation component supports the following structures in the view for C++ source files. The basic structure of the information is shown below.



The C++ navigation view uses a number of icons to describe the type of function or variable the icon belongs too. These are listed in the table below:

Icon	Description
	Public member function
	Protected member function
	Private member function
	Public member variable
	Protected member variable
	Private member variable

Double clicking on a navigation item by default jumps you to the associated navigation items declaration. This default behavior can be modified by selecting **Jump To Definition On Double-Click** from the pop-up menu (this option is unchecked by default). When this option is checked, double-clicking a navigation item jumps you to the associated navigation items definition.

To jump to the definition

Select either of the following ways.

- Right-click on navigation items on the **Navigation** tab to display a pop-up menu and check that **Jump To Definition On Double-Click** is checked. Double-click a navigation item on the **Navigation** tab.
- Right-click on navigation items on the **Navigation** tab to display a pop-up menu. Select **Go to Definition**.

To jump to the declaration

Select either of the following ways.

- Right-click on navigation items on the **Navigation** tab to display a pop-up menu and check that **Jump To Definition On Double Click** is unchecked. Double-click a navigation item on the **Navigation** tab.

- Right-click on navigation items on the **Navigation** tab to display a pop-up menu. Select **Go to Declaration**.

To list the member variables and functions in the alphabetical order

1. Right-click on navigation items on the **Navigation** tab to display a pop-up menu.
2. Un-check **Group by Access**. This option is unchecked by default.

To group the display of public, private, and protected member variables and functions together

1. Right-click on navigation items on the **Navigation** tab to display a pop-up menu.
2. Check **Group by Access**. This option is unchecked by default.

Note that **Group By File** and **Group By Access** cannot both be on at the same time. Switching one on will switch the other one off.

Another useful facility is the capability of viewing the base or derived classes for a certain selection.

To view the Base or derived classes

1. Right-click on class on the **Navigation** tab to display a pop-up menu.
2. To see the derived classes for the selection click the **Show Derived Classes** menu item. To see the base classes for the selection click the **Show Base Classes** menu item.
3. Depending on the selection a dialog is displayed which shows the class structure selected in an expanded tree format.
4. Click **Close** to close this dialog once you have the information you require.

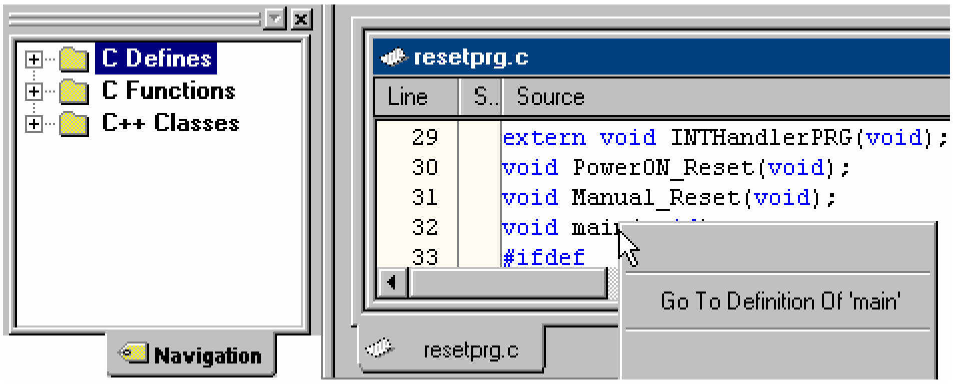
12.3 Jump to a definition from the editor

It is possible to select navigation items of #defines, C functions, or C++ classes from the source codes shown in the editor window and view the positions where these navigation items are defined.

Note, however, that these navigation items must be under the categories checked on the Navigation tab of the workspace window.

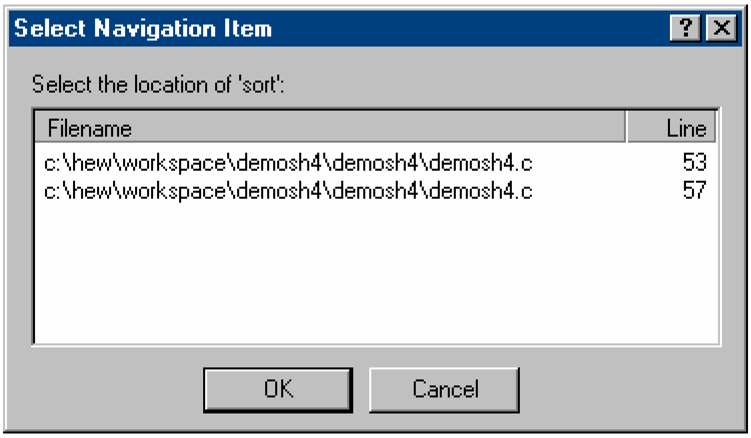
To jump to a definition

1. Right-click on a navigation item, which you wish to view where it is defined, within the **Source** field of the Editor window.
2. Select **Go to Definition Of 'Navigation item'** from the pop-up menu.



```
29  extern void INTHandlerPRG(void);
30  void PowerON_Reset(void);
31  void Manual_Reset(void);
32  void mai
33  #ifdef
```

3. If two or more navigation items are found, the **Select Navigation Item** dialog box appears. File names and line numbers are listed in the dialog box. Double-click an item, or select an item and press the OK button.



Select the location of 'sort':

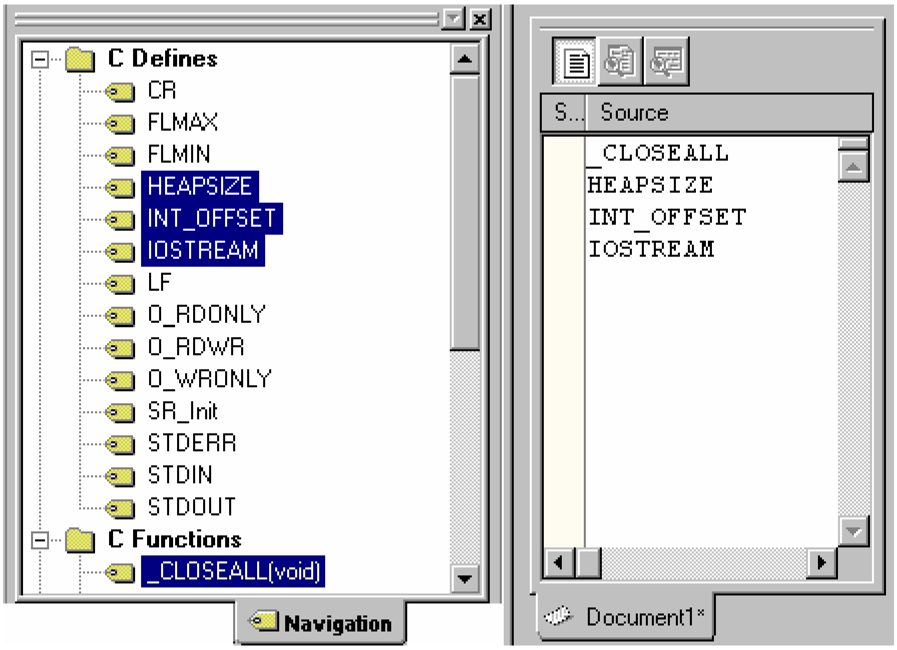
Filename	Line
c:\hew\workspace\demosh4\demosh4\demosh4.c	53
c:\hew\workspace\demosh4\demosh4\demosh4.c	57

12.4 Drag and drop navigation items

It is possible 'drag and drop' navigation items of #defines, C functions, or C++ classes shown in the Navigation tab of the workspace window. The Category (C Defines, C Functions, and C++ Classes), Globals Folder, and File items will not be draggable.

Drag and drop of navigation items into a file that's currently open in the editor

This makes it easier for you to write code, as you will be able to drag the relevant items into the file, instead of having to type them.



```
_CLOSEALL
HEAPSIZE
INT_OFFSET
IOSTREAM
```

- Multiple items can be dragged at the same time (these will appear on separate lines).
- The navigation item names will be inserted in alphabetical order.
- Note that if you drag a function, only the function's name will be dragged.

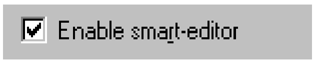
12.5 Smart edit capability in the HEW editor

Another feature of the High-performance Embedded Workshop is its smart edit facility. This is enabled by default for all C++ source files. This feature allows the HEW editor to access C++ navigation information and provide auto-completion help when using C functions, #defines, C++ classes, and member functions.

The HEW editor accesses C functions, #defines, and C++ navigation information and shows the smart editor's list in a pop-up window.

To use the Smart edit facilities

1. Click on the [**Setup->Options**] menu item. The **Options** dialog box will be invoked.
2. Select the **Editor** tab.
3. The **Enable Smart-edit** should be checked.
4. Click OK.



☑ Enable smart-editor

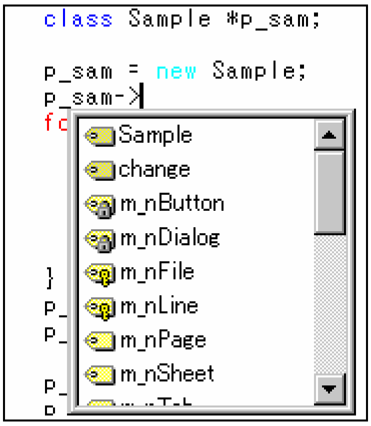
With this option switched on if you are working on C or C++ file the smart-edit capability should be enabled.

Note:

When the navigation category **C Defines, C Functions**, or **C++ Classes** is unchecked in the Navigation tab of the workspace window, the HEW's smart editor function is disabled.

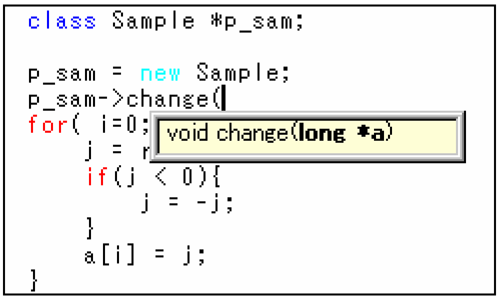
During normal usage the following editor operations will make the smart edit facilities visible.

- If you are using an object and are trying to access the members using the '.' or '->'. If you do this a pop-up window will be displayed which may help you select the correct member more efficiently than typing. Whilst typing the pop-up window will keep track of the keys you have pressed to help your selection. If you press return then the currently selected member will be added. This pop-up window is also used when using the '::' method and it is displayed in figure below. Pressing CTRL+SPACE displays the pop-up window. This pop-up window disappears when a member is added.



```
class Sample *p_sam;

p_sam = new Sample;
p_sam->
fc
}
p_
p_

p_
p
```

- If you are trying to use C or C++ functions then the pop-up window in Figure below is displayed when the first open bracket is entered. This pop-up window allows you to see what functions are available for the current object. Selecting the function automatically enters the remaining parameters for you. Pressing CTRL+SHIFT+SPACE opens this pop-up window, which is visible until a closing bracket is entered.



```
class Sample *p_sam;

p_sam = new Sample;
p_sam->change(
for( i=0;
     j = r
     if(j < 0){
          j = -j;
     }
     a[i] = j;
}
```

- It is possible to select a C function, #define, C++ class, or member function in the pop-up window and add it to the HEW editor window by either of the following ways:

  - Right-click within the HEW editor window to display a pop-up menu and select **List Members**. A pop-up window opens. Select an item you wish to add and press ENTER, **OR**

  - Press CTRL+SPACE to display a pop-up window. Select an item you wish to add and press ENTER.

13. Map

Among the section settings of the linkage editor and the contents of the linkage list file output by the linkage editor, information on sections and symbols are respectively shown. There are the following features.

Sections

- A tree view in the form of address – section group – section
- Assignment of sections can be changed by drag and drop
- You can add, modify, or delete addresses, section groups, and sections
- The source file for the address in the selected line can be opened in the editor window
- The contents can be printed out

Symbols

- Lists all symbol information or those for respective sections
- You can search for, filter, or sort (this function is not supported by the evaluation version) symbol information
- The source file for the address in the selected line can be opened in the editor window
- The contents can be printed out

- **Supported CPUs**

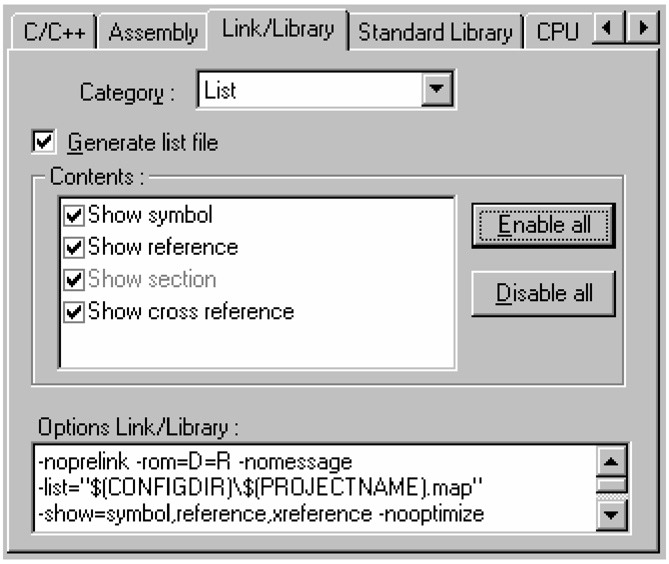
In HEW V.4.01 or later version, the SuperH RISC engine and H8SX, H8S, and H8 family compiler packages support the map function.

M16C, M32C, and M32R family compiler package supports the map function only when the version of the HEW included in the compiler package is V.4.01 or later.

- **To view the information on sections and symbols**

To view the information on sections and symbols, the following setting is required in advance:

1. Select [**Build ->Tool**] to open the build options (SuperH RISC engine Standard Toolchain) dialog box.
2. Select **List** for **Category** in the **Link/Library** tab.
3. Select the **Generate list file** checkbox. This allows output of information on sections.
4. Click the **Enable all** button. This allows output of information on symbols.
5. Select [**Build-> Build**].



Category : List

☑ Generate list file

Contents :

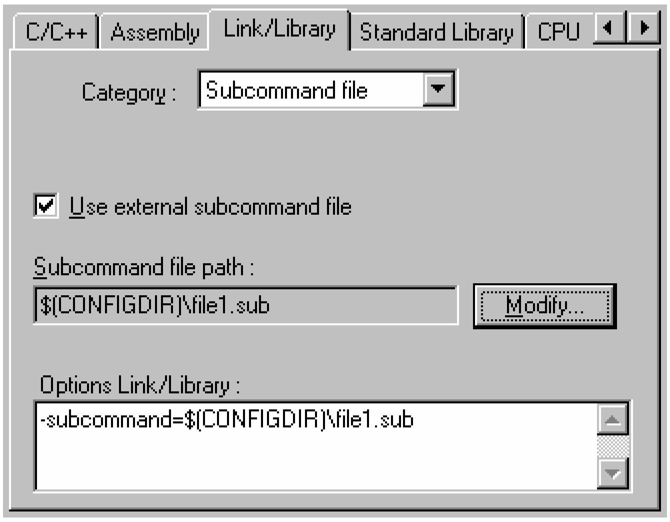
☑ Show symbol
☑ Show reference
☑ Show section
☑ Show cross reference

Options Link/Library :

```
-noprelink -rom=D=R -nomessage
-list="$(CONFIGDIR)\$(PROJECTNAME).map"
-show=symbol,reference,xreference -nooptimize
```

Note:

If a subcommand file is specified as shown below, the section setting information of the linkage editor will not be shown in the **Map Section Information** window.



Category : Subcommand file

☑ Use external subcommand file

Subcommand file path : $(CONFIGDIR)\file1.sub

Options Link/Library :

```
-subcommand=$(CONFIGDIR)\file1.sub
```

13.1 Managing section settings

You can add, modify, or delete information on sections on the GUI through the **Map Section Information** window. It is also easy to check the section settings after modification.

13.1.1 Opening the Map Section Information window

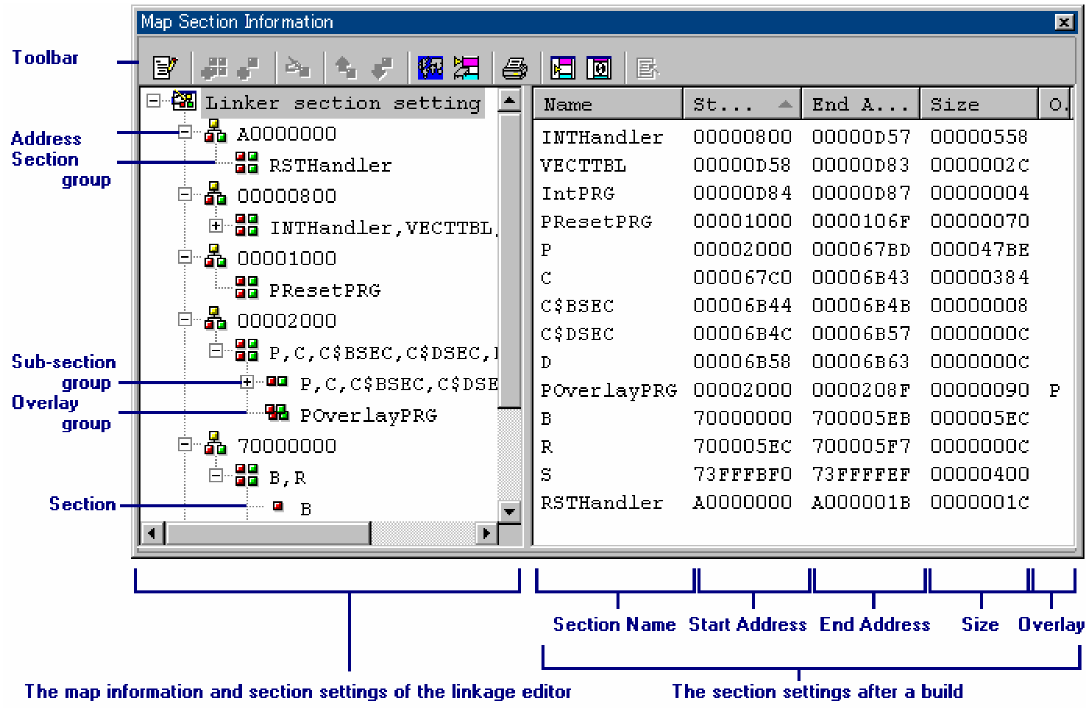
To open the Map Section Information window

1. Select one of the following operations:

- Click the **Map** toolbar button (MAP), **OR**
- Select [**View->Map**].

2. The **Select Map Window Type** dialog box opens.
3. Select "Map Section Information" in the **Map** drop-down list.

Window configuration



```
Linker section setting
  A0000000
    RSTHandler
  00000800
    INTHandler,VECTTBL,
  00001000
    PResetPRG
  00002000
    P,C,C$BSEC,C$DSEC,I
      P,C,C$BSEC,C$DSE
      POverlayPRG
  70000000
    B,R
      B
```

Name	St...	End A...	Size	O.
INTHandler	00000800	00000D57	00000558	
VECTTBL	00000D58	00000D83	0000002C	
IntPRG	00000D84	00000D87	00000004	
PResetPRG	00001000	0000106F	00000070	
P	00002000	000067BD	000047BE	
C	000067C0	00006B43	00000384	
C$BSEC	00006B44	00006B4B	00000008	
C$DSEC	00006B4C	00006B57	0000000C	
D	00006B58	00006B63	0000000C	
POverlayPRG	00002000	0000208F	00000090	P
B	70000000	700005EB	000005EC	
R	700005EC	700005F7	0000000C	
S	73FFFBF0	73FFFFEF	00000400	
RSTHandler	A0000000	A000001B	0000001C	

- The left pane shows the map information and section settings of the linkage editor.

- Section settings of the linkage editor are displayed.

- While in the edit mode, locations of subsection groups, overlay groups, and sections can be changed by drag and drop. If anything is dropped into a different section group, this item is added to the last of other items in the section group. A section dropped into a section group, subsection group, or overlay group is added to the next of the item where dropped.

- The right pane shows the section settings after a build.

- Nothing is displayed in this pane if there is no linkage list file (.map) output by the linkage editor.

- You can adjust the column width by dragging the mouse on a column header. Moreover, right-clicking the column header displays a pop-up menu. A tick mark right next to an entry indicates that this column is displayed. Clicking an entry will switch showing/hiding the column.

- Clicking the column header sorts the listed items. When the window is first opened, the items are sorted by the value of "Start Address".

- The "Overlay" column shows the name of the primary section.

Window options

- Map information and section settings of the linkage editor (left pane)

Right-clicking displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up menu options	Toolbar button	Function
Section Edit Mode		Enter the edit mode.
Add Section Group		Add the section group.
Add Section		Add the section.
Add Overlay Group *1		Add the overlay group.
Edit Selected Item		Edits the selected item.

Cut	✂	Cuts the section information.
Copy	⧉	Copies the section information.
Paste	📋	Pastes the section information.
Delete	✗	Deletes the section information.
Move up	⇧	Moves up the section information.
Move Down	⇩	Moves down the section information.
Change Primary Section		Changes primary section.
Memory Map Setting *2		Sets a memory map.
Memory Resource Auto Allocate *2		Automatically allocation the memory resource.
Print	🖨	Prints section setting tree view.
Toolbar display	-	Shows or hides the toolbar.
Customize toolbar	-	Customizes toolbar buttons.

***1**. This pane cannot be used when the linkage editor does not support overlay.

***2**. Support for this function depends on the selected debugging platform. In HEW V.4.03, SuperH RISC engine and H8SX, H8S, and H8 family simulator debuggers support this function.

- Section settings after a build (right pane)

Right-clicking displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up menu options	Toolbar button	Function
Show Unallocated Area		Shows the unallocated area.
Show No Size Section		Shows the no size section.
View source		Views the source code for the address in the selected line.
Print	🖨	Prints section list view.

13.1.2 Entering/exiting the edit mode

To edit the section information shown in the left pane, the HEW must enter the edit mode. By default, the edit mode is not selected.

To enter the edit mode

1. Right-click within the left pane to open a pop-up menu.
2. Select **Section Edit Mode**. This allows the HEW to enter the edit mode.
3. There is a tick mark on **Section Edit Mode**.

While the HEW is in the edit mode, the options listed in the table below are available.

Right-click on an item in the left pane. The following options with "✓ " in the table can be selected.

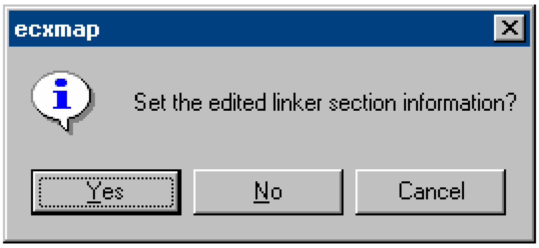
Pop-up menu options	Toolbar button	Item					
		Root	Address	Section group	Subsection group	Overlay group	Section
Add Section Group		✓	✓	✓	✓	✓	✓
Add Section			✓	✓	✓	✓	✓
Add Overlay Group *1				✓	✓		✓ *2
Edit Selected Item			✓	✓	✓	✓	✓
Cut					✓	✓	✓
Copy					✓	✓	✓
Paste			✓	✓	✓	✓	✓
Delete			✓	✓	✓	✓	✓
Move up							✓
Move Down							✓
Change Primary Section *1						✓	

***1**. This pane cannot be used when the linkage editor does not support overlay.

***2**. Will be supported by SuperH RISC engine family C/C++ compiler package V.9.00 Release 04 and H8, H8S, and H8SX family C/C++ compiler package V.6.01 Release 02 or later.

To exit the edit mode

1. Right-click within the left pane to open a pop-up menu.
2. Select **Section Edit Mode**.
3. The confirmation dialog box shown below opens. To reflect the changes to the section settings of the linkage editor, select Yes.



ecxmap

Set the edited linker section information?

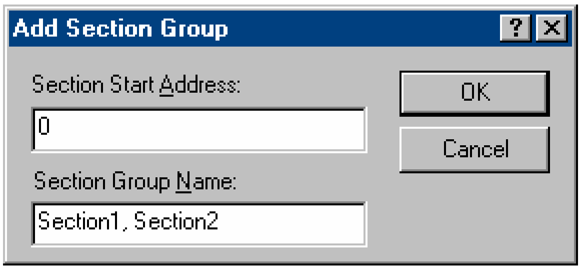
4. The HEW exits the edit mode. The tick mark on **Section Edit Mode** has disappeared.

13.1.3 Adding a section group

In the edit mode, you can add a section group.

To add a section group

1. Right-click within the left pane to open a pop-up menu.
2. Select **Add Section Group**.
3. The **Add Section Group** dialog box appears.



0

Section1, Section2

4. Enter the start address of a section in **Section Start Address**.
5. Enter a section group name in **Section Group Name**. If you specify section names separated by a comma, each of them will be added as a section. In the example above, “Section1” and “Section2” are added as sections.

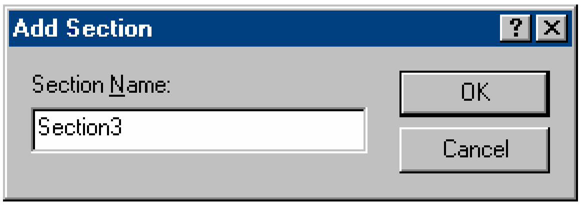
In the tree of the left pane, the new section group is added under the address.

13.1.4 Adding a section

In the edit mode, you can add a section under a section group, subsection group, or overlay group.

To add a section

1. Right-click on a section group, subsection group, or overlay group to open a pop-up menu.
2. Select **Add Section**.
3. The **Add Section** dialog box appears.



Section3

4. Enter a section name in **Section Name**.

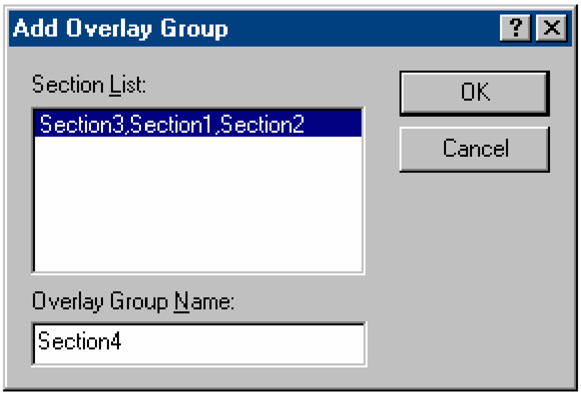
In the tree of the left pane, the new section is added under the section group, subsection group, or overlay group.

13.1.5 Adding an overlay group

In the edit mode, you can add an overlay group (*1) under a section group, subsection group, or section (*2).

To add an overlay group

1. Right-click on a section group, subsection group, or section to open a pop-up menu.
2. Select **Add Overlay Group**.
3. The **Add Overlay Group** dialog box appears.



Section3,Section1,Section2

Section4

4. Enter an overlay group name in **Overlay Group Name**.

In the tree of the left pane, the new overlay group is added under the section group, subsection group, or section.

***1**. This item cannot be used when the linkage editor does not support overlay.

***2**. Will be supported by SuperH RISC engine family C/C++ compiler package V.9.00 Release 04 and H8, H8S, and H8SX family C/C++ compiler package V.6.01 Release 02 or later.

13.1.6 Editing a selected item

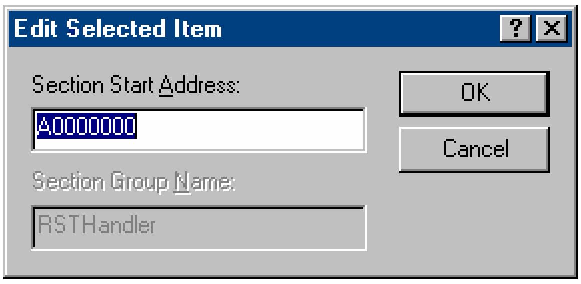
In the edit mode, you can also modify the address value and the name of a section group, subsection group, overlay group, or section.

To modify the selected item

1. Select one of the following operations:

- Right-click on an item in the left pane to open a pop-up menu. Select **Edit Selected Item**, **OR**
- Double-click Modify a value or name by in-place editing.

2. The **Edit Selected Item** dialog box appears.
3. Enter a string or integer value into an edit field.



Section Start Address: A0000000

Section Group Name: RSTHandler

13.1.7 Setting the primary section

In the edit mode, you can set a selected overlay group as the primary section.*

To set the primary section

1. Right-click on an overlay group in the left pane to open a pop-up menu.
2. Select **Change Primary Section**.

In the tree of the left pane, the selected overlay group is set as the primary section.

*. This function cannot be used when the linkage editor does not support overlay.

13.1.8 Setting a memory map

While the HEW is connected with a debugging platform, you can set a memory map.

To set a memory map

1. Right-click within the left pane to open a pop-up menu.
2. Select **Memory Map Setting**.
3. The memory map setting dialog box is displayed. The current mapping of the address space is shown as a list. Set a memory map or memory resource as necessary.
4. Click the OK button.

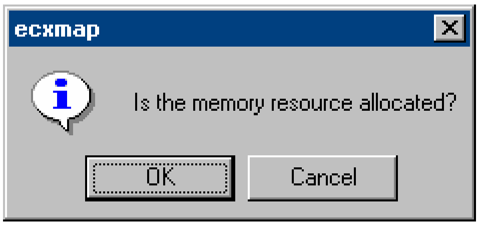
Support for this function depends on the debugging platform.

13.1.9 Automatically allocating the memory resource

When there is a linkage list file (.map) output by the linkage editor, the memory resource can be automatically allocated according to the memory map and linkage map information. While the HEW is connected with a debugging platform, you can allocate a memory resource.

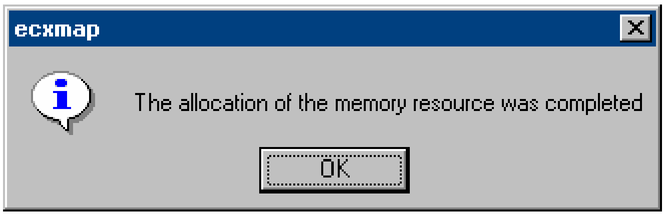
To automatically allocate the memory resource

1. Right-click within the left pane to open a pop-up menu.
2. Select **Memory Resource Auto Allocate**. The dialog box shown below appears.



Is the memory resource allocated?

3. To continue processing, click OK.
4. When allocation is completed, the dialog box shown below appears.



The allocation of the memory resource was completed

Support for this function depends on the debugging platform.

13.1.10 Printing out the section settings tree

You can also print out the section settings tree in the left pane.

To print out the section settings tree

1. Right-click within the right pane to open a pop-up menu.
2. Select **Print**.
3. The standard print formatting and selection dialog box is displayed. From here you can choose your printer and page setup options.

13.1.11 Viewing unallocated areas

You can view areas where no section is allocated. By default, this function is not selected.

To view unallocated areas

1. Right-click within the right pane to open a pop-up menu.
2. Select **Show Unallocated Area**.
3. There is a tick mark on **Show Unallocated Area**. The right pane shows all areas where no section is allocated, which is indicated by “Unallocated Area” in the Name column.

13.1.12 Viewing sections of size 0

You can view sections of size 0. By default, this function is not selected.

To view sections of size 0

1. Right-click within the right pane to open a pop-up menu.
2. Select **Show No Size Section**.
3. The right pane shows all sections of size 0. At this time, the “End Address” column does not show the end addresses.

13.1.13 Viewing the source code for the address

The source file for the address in the selected line will be opened in the editor window.

To view the source code for the address

Select one of the following operations:

- Right-click on the section-list line in the right pane to open a pop-up menu. Then select **View Source**, **OR**
- Double-click on the section-list line in the right pane.

13.1.14 Printing out the section list

You can also print out the section list in the right pane.

To print out the section list

1. Right-click within the right pane to open a pop-up menu.
2. Select **Print**.
3. The standard print formatting and selection dialog box is displayed. From here you can choose your printer and page setup options.

13.2 Viewing symbols

You can view symbols in each of the sections in the **Map Symbol Information** window.

Even if there are too many symbols, the search or filtering function can be used to view necessary information only.

13.2.1 Opening the Map Symbol Information window

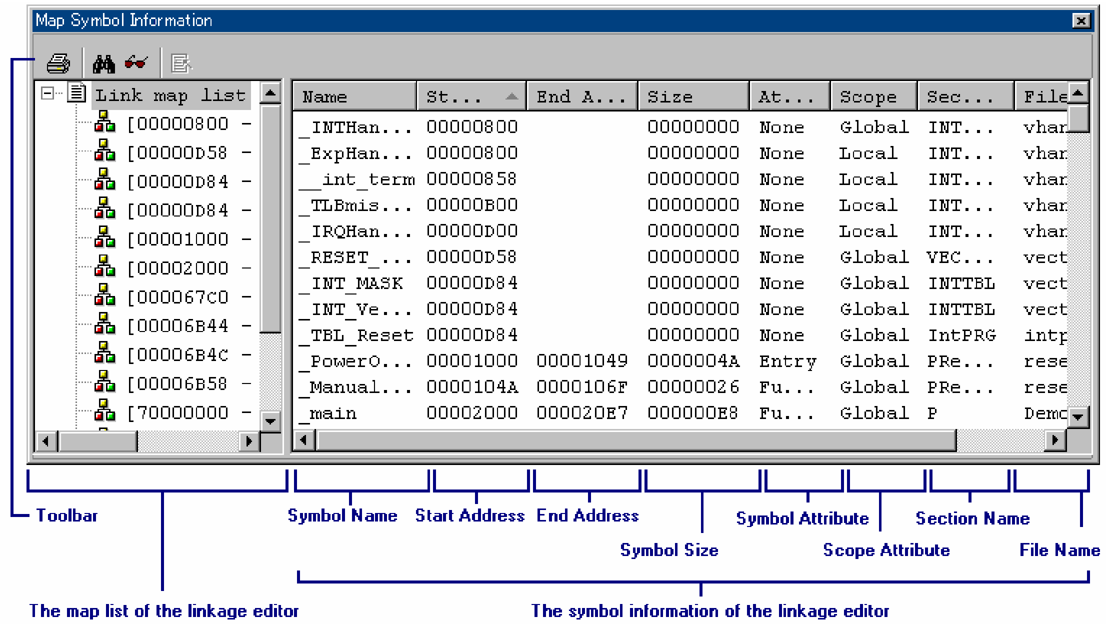
To open the Map Symbol Information window

1. Select one of the following operations:

- Click the **Map** toolbar button (MAP), **OR**
- Select [**View->Map**].

2. The **Select Map Window Type** dialog box opens.
3. Select "Map Symbol Information" in the **Map** drop-down list.

Window configuration



Map Symbol Information

- Link map list
  - [00000800 –
  - [00000D58 –
  - [00000D84 –
  - [00000D84 –
  - [00001000 –
  - [00002000 –
  - [000067C0 –
  - [00006B44 –
  - [00006B4C –
  - [00006B58 –
  - [70000000 –

Name	St...	End A...	Size	At...	Scope	Sec...	File
_INTHan...	00000800		00000000	None	Global	INT...	vhar
_ExpHan...	00000800		00000000	None	Local	INT...	vhar
__int_term	00000858		00000000	None	Local	INT...	vhar
_TLBmis...	00000B00		00000000	None	Local	INT...	vhar
_IRQHan...	00000D00		00000000	None	Local	INT...	vhar
RESET...	00000D58		00000000	None	Global	VEC...	vect
_INT_MASK	00000D84		00000000	None	Global	INTTBL	vect
_INT_Ve...	00000D84		00000000	None	Global	INTTBL	vect
_TBL_Reset	00000D84		00000000	None	Global	IntPRG	intp
_PowerO...	00001000	00001049	0000004A	Entry	Global	PRe...	rese
_Manual...	0000104A	0000106F	00000026	Fu...	Global	PRe...	rese
_main	00002000	000020E7	000000E8	Fu...	Global	P	Demo

- The left pane shows the map list of the linkage editor.

- Nothing is displayed in this pane if there is no linkage list file (.map) output by the linkage editor.

- Selecting the root of the map list ("Linker map list") shows all symbol information in the right pane.

- Selecting a section in the map list only shows symbol information of the selected section in the right pane.

- The right pane shows the symbol information of the linkage editor.

- Clicking the column header sorts the listed items. When the window is first opened, the items are sorted by the value of "Start Address".

- The "Attribute" column shows the symbol attribute.

Entry:	Entry function
Function:	Function name
Data:	Variable name
None:	Undefined (label or assembly symbol)

- The "Scope" column shows the scope attribute.

Global:	Global symbol
Local:	Local symbol

Window options

- Map list of the linkage editor (left pane)

Right-clicking displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up menu options	Toolbar button	Function
Print	🖨	Prints section tree view.
Toolbar display	-	Shows or hides the toolbar.
Customize toolbar	-	Customizes toolbar buttons.

- Symbol information of the linkage editor (right pane)

Right-clicking displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up menu options	Toolbar button	Function
Find	🔍	Finds a symbol.
Find Next	🔍	Finds the next symbol that matches the search text.
Filter	👓	Filters the symbol information.
View source	📝	Views the source code for the address in the selected line.
Print	🖨	Prints section list view.

13.2.2 Printing out the map list

You can also print out the map list in the left pane.

To print out the map list

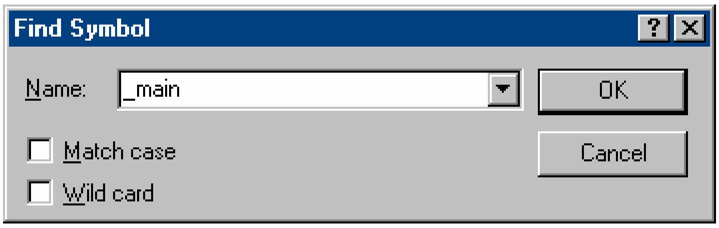
1. Right-click within the left pane to open a pop-up menu.
2. Select **Print**.
3. The standard print formatting and selection dialog box is displayed. From here you can choose your printer and page setup options.

13.2.3 Finding symbols

You can search for the linkage editor symbols.

To search for symbols

1. Right-click within the right pane to open a pop-up menu.
2. Select **Find**.
3. The **Find Symbol** dialog box appears.



4. Enter a symbol name you wish to find in **Name**, or select one from those you have previously searched for in the drop-down list box.
5. To distinguish uppercase and lowercase characters, select the **Match case** checkbox.
6. To use wild-cards (? or *), select the **Wild card** checkbox.

   ?: A character
   *: A string

7. Click OK.

When a symbol is found, a line that contains the symbol is highlighted in the right pane.

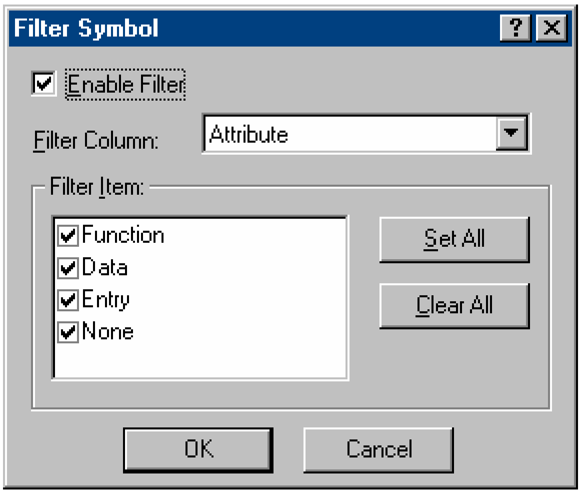
To find the next occurrence of the symbol, select **Find Next**.

13.2.4 Filtering the symbol information

You can view the filtered symbol information of the linkage editor.

To filter the symbol information

1. Right-click within the right pane to open a pop-up menu.
2. Select **Filter**.
3. The **Filter Symbol** dialog box appears.



4. Select the **Enable Filter** checkbox. By default, this checkbox is not selected.
5. Select a column in **Filter Column**.
6. Select a filtering condition in **Filter Item**.

Filter Column	Filter Item	Function
Attribute (symbol attribute)	Function	Filter by functions
	Data	Filter by data symbols
	Entry	Filter by positions to start execution
	None	Filter by other symbols (such as labels)
Scope (scope attribute)	Global	Filter by global symbols
	Normal	Filter by local symbols
Section (sections containing symbols)	Section name	Filter by section names
File (files containing symbols)	File name	Filter by file names

7. Clicking the **Set All** button selects the check boxes of all filtering conditions.
8. Clicking the **Clear All** button de-selects the check boxes of all filtering conditions.
9. Click OK.

Symbol information of the linkage editor shown in the right pane is filtered by the selected condition. For further filtering by combining a different condition, select another column and then select a filtering condition in the column.

To disable filtering, de-select the **Enable Filter** checkbox in the **Filter Symbol** dialog box.

13.2.5 Viewing the source code for the address

The source file for the address in the selected line will be opened in the editor window.

To view the source code for the address

Select one of the following operations:

- Right-click on the symbol-list line in the right pane to open a pop-up menu. Then select **View Source**, **OR**
- Double-click on the symbol-list line in the right pane.

13.2.6 Printing out the symbol information

You can also print out the symbol information in the right pane.

To print out the symbol information

1. Right-click within the right pane to open a pop-up menu.
2. Select **Print**.
3. The standard print formatting and selection dialog box is displayed. From here you can choose your printer and page setup options.

14. Using the Command Line

The High-performance Embedded Workshop (HEW) Command Line Interpreter allows the user to control the debugging platform by sending text-based commands instead of the window menus and commands. It is especially useful if a series of predefined commands need to be sent to the debugging platform by calling them from a batch file and, optionally, recording the output in a log file.

Note:

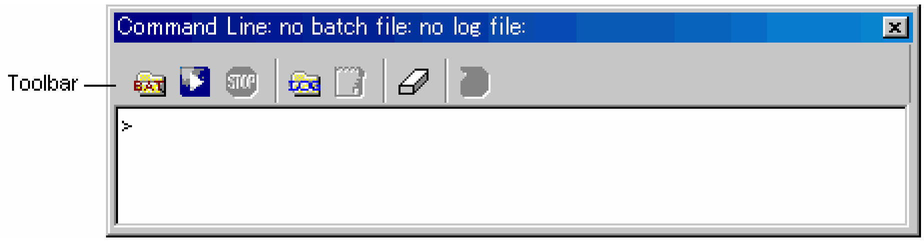
To specify a file in the command line, use a placeholder (excluding TCL). If you wish to specify a directory not included in the placeholder, specify an absolute path. After specifying the absolute path, this file will not be correctly found when it is in another host computer or environment where the path content is different. In such cases, specify the file again.

Example: FILE_LOAD ELF/DWARF2 $(CONFIGDIR)\\demo.abs

14.1 Opening the Command Line window

Choose [**View->Command Line**] or click the **Command Line** toolbar button ( ) to open the Command Line window.

Window configuration



Command Line: no batch file: no log file:

```
>
```

- This window allows the user to control the debugging platform by sending text-based commands.
- A series of predefined command lines can be called from a file and the output can be recorded in a file.
- The command can be executed by pressing the Enter key after the command is input at the prompt (>) on the last line. For information about the available commands, see Reference 3, List of Commands, and the on-line help.
- If available, the window title displays the current batch and log file names separated by colons.
- Pressing the CTRL + or CTRL + ↓ keys on the last line displays the previously executed command line.
- The HEW command and TCL commands can be input in this window.

Option

Right-clicking displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

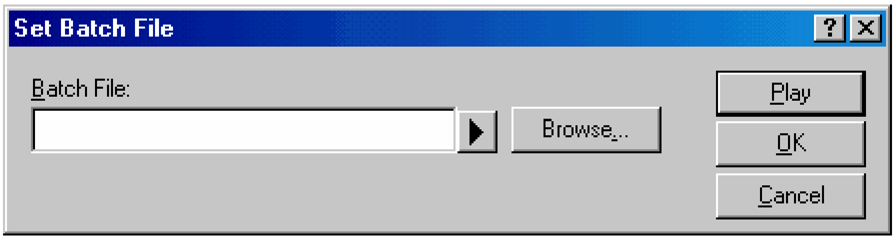
The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up menu options		Toolbar button	Macro Recording	Function
Set Batch File		BAT	-	Specifies a command file.
Play		▶	-	Plays the current command file.
Stop		STOP	-	Stops execution of the current command file.
Set Log File		LOG	-	Specifies a log file.
Logging		📝	-	Starts or stops output to log file.
Browse		🔍	-	Enters a full path to the file.
Placeholder	Configuration directory	Configuration Directory ▼	-	Pastes the $(CONFIGDIR) placeholder.
	Configuration name		-	Pastes the $(CONFIGNAME) placeholder.
	Project directory		-	Pastes the $(PROJDIR) placeholder
	Project name		-	Pastes the $(PROJECTNAME) placeholder.
	Workspace directory		-	Pastes the $(WORKSPDIR) placeholder.
	Workspace name		-	Pastes the $(WORKSPNAME) placeholder.
	HEW Installation directory		-	Pastes the $(HEWDIR) placeholder.
Select All		✎	-	Selects (i.e. highlights) the entire contents of the active window.
Copy		⧉	-	Places a copy of the highlighted text into the Windows® clipboard.
Cut		✂	-	Removes the selected text block and copy them onto the Windows® clipboard.
Paste		📋	-	Copies the contents of the Windows® clipboard into the active window at the position of the insertion cursor.
Clear Window		⌫	-	Clears the contents of the window.
Undo		↶	-	Undoes the last operation.
-		▣	-	Once an opening bracket "[" or "{" is input and the nesting follows, this button is visible. (This button selection is impossible.)
Toolbar display		-	-	Shows or hides the toolbar.
Customize toolbar		-	-	Customizes toolbar buttons.

14.2 Specifying a command file

It is useful to use a command file when a series of predefined command lines need to be executed. Create a command file by a text editor and write necessary command lines. The default extension of a command file is .hdc.

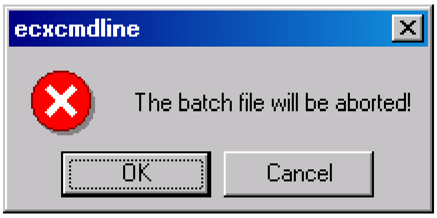
Choose **Set Batch File** from the pop-up menu to open the Set Batch File dialog box, in which the name of a command file (*.hdc) can be specified. Clicking the OK button displays the specified command file name as the window title. Clicking the Cancel button closes the dialog box without modifying the setting.



14.3 Executing a command file

Click the **Play** button in the **Set Batch File** dialog box or choose **Play** from the pop-up menu to execute the command file. The **Play** menu is displayed in gray while the file is running and can be used when the command file execution stops and control returns to the user.

The message box shown below appears when a user attempts to close the Command Line window during execution of a command file.



The batch file will be aborted!

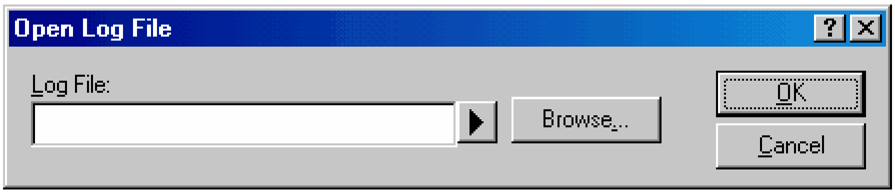
To terminate the execution of the command file and close the Command Line window, click OK. To continue the execution, click Cancel. This does not close the Command Line window.

14.4 Stopping command execution

Choose **Stop** from the pop-up menu to stop command execution. The **Stop** menu becomes valid during command execution.

14.5 Specifying a log file

Choose **Set Log File** from the pop-up menu to open the **Open Log File** dialog box, in which a log file to store the command execution results can be specified.



Enter the name of a log file (*.log). The logging option is automatically set and the name of the file is shown on the window title bar.

Opening a previous log file will ask the user if they wish to append or overwrite the current log.

14.6 Starting or stopping logging

Choose **Logging** from the pop-up menu to toggle logging to file on and off. When logging is active, the button becomes effective. Note that the contents of the log file cannot be viewed until logging is completed, or temporarily disabled by clearing the check box. Re-enabling logging will append to the log file.

14.7 Entering a full path to the file

It is recommended that the full path to a file is specified as a file name in the Command Line window because the current directory can be moved. However, care must be taken to enter the correct full path to a file when it is entered from the keyboard. To save this trouble, a full path can be easily specified by browsing through files.

Choose **Browse** from the pop-up menu to open the Browse dialog box. Select a file and click **Open** to paste the full path to the selected file to the cursor location. This option can only be used when the cursor is located on the last line.

14.8 Pasting a placeholder

Select a placeholder from the **Placeholder** submenu in the pop-up menu to paste the selected placeholder to the cursor location. This function is only available when the cursor is located on the last line.

Placeholder sub-menu	Placeholder
Configuration directory	$(CONFIGDIR)
Configuration name	$(CONFIGNAME)
Project directory	$(PROJDIR)
Project name	$(PROJECTNAME)
Workspace directory	$(WORKSPDIR)
Workspace name	$(WORKSPNAME)
HEW Installation directory	$(HEWDIR)

14.9 Selecting all the window contents

Choose **Select All** from the pop-up menu to select all contents in the Command Line window.

14.10 Copying the selection onto the clipboard

Choose **Copy** from the pop-up menu to copy the selected text block onto the Windows® clipboard. Only available if the test block is selected.

14.11 Cutting out the selection to the clipboard

Choose **Cut** from the pop-up menu to remove the selected text block and copy them onto the Windows® clipboard. This option is available only when you have selected the text block currently being input at the latest cursor position.

14.12 Pasting the contents of the clipboard

Choose **Paste** from the pop-up menu to insert the content of the Windows® clipboard at the current cursor position. This option can only be used when the cursor is at the last line.

14.13 Clearing the window contents

Choose **Clear** from the pop-up menu to clear the contents of the Command Line window.

14.14 Undoing the last operation

Choose **Undo** from the pop-up menu to undo the last operation at the latest cursor position (on the text currently being input).

14.15 Checking brace matching

While the **Brackets Opened Indicator** toolbar button (■) is displayed, users can see if brackets match. The button cannot be operated.

Once an opening bracket “[” or “{” is input and the nesting follows, the **Brackets Opened Indicator** toolbar button is visible (🟨) until “]” or “}” is input as the closing bracket.

Example of a TCL Function Input in the Command Line Window	State	Toolbar Button is Visible
>set bit 1	None	■
1	None	■
>set value 1	None	■
1	None	■
>set r 1	None	■
1	None	■
>if {$bit & $value} {	Opening brace is input	■
> set r 0$r	None	🟨
>} else {	Opening and closing braces are input	🟨
> set r 1$r	None	🟨
>}	Closing brace is input	🟨
> 01	None	■

15. Using the Macro-Recording Support Facility

The macro-recording support facility allows you to record operations, which are associated with the HEW system applications *1, build *2, and debugging *3, as HEW command-line commands or to execute these recorded commands.

Files to record the operations (HEW macro files) are command line batch files that have “hdc” as the extension and can be modified. These files are stored in the “Macro” directory under the HEW installation directory. By default, the file Default.hdc is created.

Note, however, that HEW is not capable of recording all operations in the HEW system corresponding to the HEW command-line commands. For the operations that can be recorded, a macro record icon (●) is shown in the "Macro Recording" column of the menu list. This indicates that this operation can be recorded into a HEW macro file. Also see section 15.5, Functions that can be recorded into macro files, where the HEW command-line commands to be recorded are listed.

*1. Changing a project, session, or configuration

*2. Compilation and build. Support for this function depends on the debugging platform.

*3. Downloading a module, changing a memory value or register value, setting/deleting a software breakpoint, and running a program

The macro-recording support facility is available in the **Tools** menu and on the **Macros** toolbar.

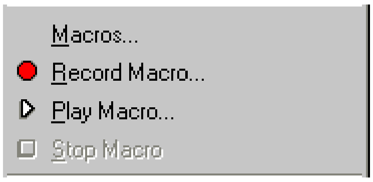
The procedure from selection of **Record Macro** (●) to that of **Stop Macro** consists one macro and one HEW macro file can record multiple macros (a set of HEW command-line commands). A macro includes multiple HEW command-line commands.

15.1 Macro menu and toolbar

The macro recorder has both a menu and toolbar integrated into the HEW.

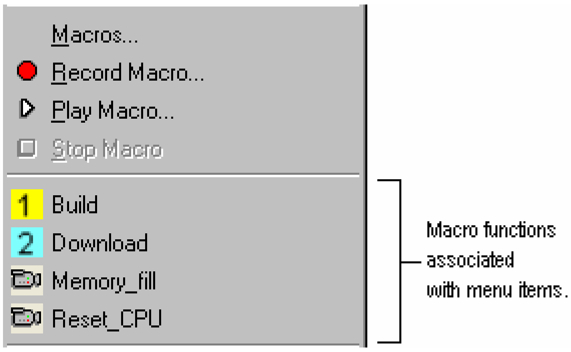
The macro menu is available on the **Tools** menu.

It has the following standard menu items:



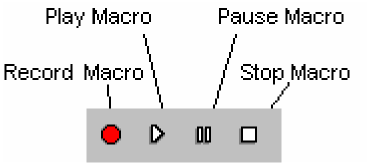
The **Macros** menu item opens the **Macro** dialog box. This allows management of the current macro files and functions. The other menus items are the same as those seen on the **Macros** toolbar.

It is also possible to associate macro functions with a menu, when you do this additional menu items are added underneath the standard macro menu items. This is shown below.



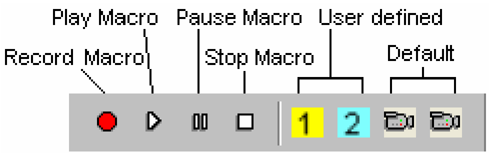
Items that also have an associated toolbar have their icons displayed alongside the menu items.

The standard buttons on the **Macros** toolbar are shown below.



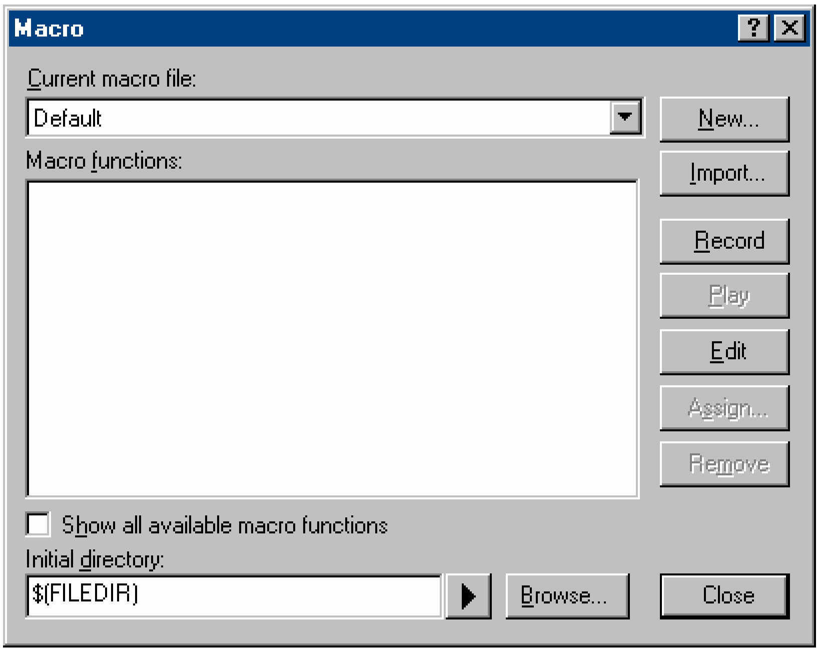
- The **Record Macro** button starts the macro recording process. After this button is clicked debugging and HEW management operations are recorded. This operation is the same as the **Record macro** menu on the Tools menu.
- The **Play Macro** button starts a macro function playback. If more than one macro is defined, the select macro function dialog is displayed to ask you to select specific macro for playback.
- The **Pause Macro** button is only enabled when recording or playing back a macro function. This pauses the current operation so that you can return to it later.
- The **Stop Macro** button is only enabled when recording or playing back a macro function. This halts the current macro operation that is being carried out.

In the same way you can associate macros with a menu you can associate them with a toolbar. When you associate the macro you can decide to use a user defined toolbar icon or the default HEW macro function icon. Once added the associated macros appear to the right of the standard macro buttons. Clicking these buttons then automatically launches the associated macro function.



15.2 Using the Macro dialog box

The **Macro** dialog box allows you to access and manager the macro files currently in use in the HEW system. The **Macro** dialog box is shown below.



The **Current macro file** drop-down list holds all of the macro files currently in use in the HEW system. These files are stored in the HEW installation directory in the "Macro" folder. Any files that are copied into this directory are automatically added to the **Current macro file** drop-down list.

The macros functions in the **Current macro file** selection are shown in the **Macro functions** list box. Changing the **Current macro file** selection will automatically update this list. If you check the **Show all available macro functions** check box then all macros currently defined are displayed in the list. When selected this option ignores the value of the **Current macro file** drop-down list box.

It can be useful to use multiple macro files to logically hold related areas. For example you might like to create a different macro file for use with different debugger targets.

To execute macro files that have relative paths, specify the initial directory in **Initial directory**. Change the directory in the same way as using the change-directory command and execute macro files in the initial directory.

To create a new blank macro file

1. Select the [**Tools->Macros**] menu option. The **Macro** dialog box is displayed.
2. Click the **New** button.
3. Enter the new macro file name.
4. Click OK. The new file name is added to the **Current macro file** drop-down list.

To import an existing macro file

1. Select the [**Tools->Macros**] menu option. The **Macro** dialog box is displayed.
2. Click the **Import** button.
3. Browse to the existing macro file.
4. Click **Select**. The new file name is added to the **Current macro file** drop-down list.
5. When you select the macro file name in the list the available macros are displayed.

It is possible to manually edit any of the macro files. The files are text based HEW command batch files.

To edit an existing macro file

1. Select the [**Tools->Macros**] menu option. The **Macro** dialog box is displayed.
2. Select the macro file, the function exists in.
3. Select the macro in the function list you wish to edit.
4. Click the **Edit** button.
5. The **Macros** dialog box is dismissed and the file is opened in the editor.

You can also drag and drop a macro file into the HEW window to open this file (in the same way as opening an ordinary text file).

To remove the currently selected macro from the existing macro file

1. Select the [**Tools->Macros**] menu option. The **Macro** dialog box is displayed.
2. Select the macro file, the function exists in.
3. Select the macro in the function list you wish to remove.
4. Click the **Remove** button.

Pressing the **Close** button closes the **Macro** dialog box. It is not possible to cancel operations on this dialog box.

15.3 Importing a macro file with existing macros

If you want to import an existing HEW macro file from another machine which already has a number of macros this is easily achieved by the following operations.

One method is to simply copy the new macro file into the **Macros** folder in the HEW root directory. HEW will automatically detect the new file and add its information to the HEW macros dialog box. The new file will not be the default file for use so you will then have to enter the macro dialog box to switch the default macro file to the newly copied one.

Another method for importing macro files is from the macro dialog. The instructions are shown below:

To import an existing macro file into HEW

1. Select the [**Tools->Macros**] menu option. The **Macro** dialog box is displayed.
2. Click the **Import** button.
3. A standard file browser is opened for you to select the existing macro file you wish to import. Select the file and click OK.
4. The file is now imported. It automatically becomes the default macro file in the Current macro file drop-down list.
5. The file is also copied into the "Macro" folder in the HEW root directory.
6. Then click **Close** on the **Macro** dialog box and the macro file is ready for use.

15.4 Recording a macro

There are a number of ways to record a macro file in the HEW, these options are discussed below. Note that not all functions in the HEW system can be recorded currently. For HEW V.4.03 the recording function is limited to workspace, project and session management, some build features and all debugging features for targets that support the feature.

To record a macro from the toolbar or menu

1. Click the **Record Macro** button (●) or menu item on the **Macros** toolbar or the **Tools** menu. To indicate that a recording operation is taking place the mouse cursor is modified to include the record icon (↖●). When the macro is being recorded the Pause Macro (⏸) and Stop Macro (□) buttons and menus are enabled. When the macro is being recorded the pause and stop buttons (↖● ⏸) and menus are enabled.
2. If you are using a debugging platform that does not support the macro facility a warning is displayed that states the following "Warning: The target does not support macros. Recording will be limited. Do you wish to continue?". If you click yes the record operation will continue.
3. Recording will continue until you click **Stop Macro**. For details, see section 15.5, Functions that can be recorded into macro files.
4. When **Stop Macro** is clicked a dialog is displayed that allows you to enter the macro name.
5. Clicking OK saves the name and adds it to the current macro file. Clicking cancel dismisses the dialog box and loses the macro recording.

Another method is to launch the recording process via the Macro dialog box.

To record a macro from the macro dialog box

1. Select the [**Tools->Macros**] menu option. The **Macro** dialog box is displayed.
2. Click the **Record** button. The Macro dialog box is dismissed and the recording operation begins.

15.5 Functions that can be recorded into macro files

HEW is not capable of recording all operations in the HEW system into macro files as scripts. For the operations that can be recorded, “Record Macro” in the menu list shows a macro record icon. This indicates that the operation can be recorded into a HEW macro file.

- Recordable functions (common to all HEW products)
- Recordable functions (dependent on the debugging platform)

15.5.1 Recordable functions (common to all HEW products)

Macros of recordable HEW functions common to all HEW products included in tool packages are recorded into macro files as scripts when the operations listed below are performed.

- **Handling menus, shortcut keys, and toolbar buttons**

If you handle a menu, shortcut key, or toolbar button of those listed below while recording a macro, the command in "HEW command-line commands recorded into macros" will be recorded into a macro file.

Menu	Menu Option		Keyboard Shortcuts	Toolbar Button	Recordable HEW Command-Line Commands
File	Open Workspace		-	-	OPEN_WORKSPACE
	Save Workspace		-	-	SAVE_WORKSPACE
	Close Workspace		-	-	CLOSE_WORKSPACE
	New Session		-	-	CHANGE_SESSION
	Import Session		-	-	CHANGE_SESSION
	Save Session		-	-	SAVE_SESSION
	Refresh Session		-	-	REFRESH_SESSION
	Download A New Module		-	-	See Dialog boxes.
	Recent Workspaces		-	-	OPEN_WORKSPACE
	Recent Downloaded Modules		-	-	FILE_LOAD
Edit	Toggle Breakpoint		F9	✋	SET_DISASSEMBLY_SOFT_BREAK /SET_SOURCE_SOFT_BREAK
	Enable/Disable Breakpoint		CTRL+F9	✋	STATE_DISASSEMBLY_SOFT_BREAK /STATE_SOURCE_SOFT_BREAK
View	Workspace		ALT+K		See Windows.
	Disassembly		CTRL+D		
	CPU	Registers	CTRL+R	R1	
		Memory	CTRL+M		
		IO	CTRL+I	I/O	
Project	Set Current Project		-	-	CHANGE_PROJECT
	Insert Project		-	-	CHANGE_PROJECT
	Edit Project Configuration ***2**		-	-	SAVE_SESSION
Build ***2**	Build File		CTRL+F7		BUILD_FILE
	Build		F7		BUILD
	Build All		-		BUILD_ALL
	Build Multiple		-	-	BUILD_MULTIPLE /CLEAN
	Clean Current Project		-	-	CLEAN
	Clean All Project		-	-	CLEAN
	Build Configurations		-	-	See Dialog boxes.
Debug	Debug Sessions		-	-	
	Reset CPU		-		RESET
	Go		F5		GO
	Reset Go		SHIFT+F5		GO_RESET
	Go to Cursor		-		GO_TILL
	Set PC to Cursor		-	I PC	REGISTER_SET ***1**

	Run		-	-	GO_TILL
	Step In		F11	{}	STEP
	Step Over		F10	{}	STEP_OVER
	Step Out		SHIFT+F11	{}	STEP_OUT
	Step		-	-	See Dialog boxes.
	Step Mode	Auto	-	-	STEP_MODE
		Assembly	-	-	STEP_MODE
		Source	-	-	STEP_MODE
	Halt Program		-	STOP	HALT
	Initialize		-	-	INITIALIZE
	Connect *2		-		CONNECT
	Disconnect *2		-		DISCONNECT
	Save Memory		-	-	FILE_SAVE
	Verify Memory *2		-	-	FILE_VERIFY
	Download Modules	<File name of the download module>	-	-	FILE_LOAD
		All Download Modules	-	-	FILE_LOAD_ALL
	Unload Modules	<File name of the download module>	-	-	FILE_UNLOAD
		All Downloaded Modules	-	-	FILE_UNLOAD_ALL
Setup	Radix	Hex	-	16	RADIX
		Decimal	-	10	RADIX
		Oct	-	8	RADIX
		Bin	-	2	RADIX

*1. Refer to the help of an emulator or simulator.

*2. Support for this function depends on the debugging platform.

- **Windows**

While recording a macro, if you perform in-place edit, double-click a column, click on a pop-up menu, or make any setting in a dialog box opened by clicking the pop-up menu as shown in the table below, the command in "HEW command-line commands recorded into macros" will be recorded into a macro file.

To open the editor window, double-click a file in the Projects tab of the workspace window.

Window	Target	Function/Handling	Recordable HEW Command-Line Commands
Projects tab of the Workspace window	Pop-up menu of a workspace	Clean All Projects *2	CLEAN
	Pop-up menu of a project	Build *2	BUILD
		Build All *2	BUILD_ALL
		Clean Current Project *2	CLEAN
		Set as Current Project	CHANGE_PROJECT
	Pop-up menu of a project file	Build <File name> *2	BUILD_FILE
	Pop-up menu of a Download modules folder	Download all module	FILE_LOAD
		Download A New Module	See Dialog boxes.
	Pop-up menu of a download module	Download module	FILE_LOAD
		Download module (debug data only)	FILE_LOAD
		Unload module	FILE_UNLOAD
		Download A New Module	See Dialog boxes.
	<File name of the download module>	Download module by double-clicking	FILE_LOAD
Editor and Disassembly (source mode)	Pop-up menu	Build "<File name>" *2	BUILD_FILE
		Toggle Breakpoint	SET_SOURCE_SOFT_BREAK
		Enable/Disable Breakpoint	STATE_SOURCE_SOFT_BREAK

		Go To Cursor	GO_TILL
		Set PC Here	REGISTER_SET ***1**
	S/W Breakpoints column	Insert/delete a breakpoint by double-clicking	SET_SOURCE_SOFT_BREAK
Editor and Disassembly (mixed/disassembly mode)	Pop-up menu	Go To Cursor	GO_TILL
		Set PC Here	REGISTER_SET ***1**
		Toggle Breakpoint	SET_DISASSEMBLY_SOFT_BREAK
		Enable/Disable Breakpoint	STATE_DISASSEMBLY_SOFT_BREAK
	S/W Breakpoints - ASM column	Insert/delete a breakpoint by double-clicking	SET_DISASSEMBLY_SOFT_BREAK
Register	Value	In-place edit	REGISTER_SET ***1**
	Flag register ***2**	Edit by clicking	REGISTER_SET ***1**
	Pop-up menu	Edit	REGISTER_SET ***1**
Memory	Value	In-place edit	MEMORY_EDIT
	Pop-up menu/toolbar button	Set	MEMORY_EDIT
		Fill	MEMORY_FILL
		Move	MEMORY_MOVE
		Compare ***2**	MEMORY_COMPARE
		Save	FILE_SAVE
		Load	FILE_LOAD
IO	Value	In-place edit	MEMORY_EDIT
		Open the edit dialog box by double-clicking	MEMORY_EDIT

***1**. Refer to the help of an emulator or simulator.

***2**. Support for this function depends on the debugging platform.

- **Dialog boxes**

If you make any setting in a dialog box or change the selection on a toolbar as shown in the table below, the command in "HEW command-line commands recorded into macros" will be recorded into a macro file.

Dialog box	**Toolbar**	**Function/Handling**	**Recordable HEW Command-Line Commands**
Download Module	-	Enter Offset, Format, Filename, Access size, and Perform memory verify during download option	FILE_LOAD
Build Multiple ***1**	-	Click the Build button or the Build All button	BUILD_MULTIPLE
		Click the Clean button	CLEAN
Build Configurations	SimDebug_SH-3	Change the Current configuration drop-down list	CHANGE_CONFIGURATION
Debug Sessions	SimSessionSH-3	Change the Current session drop-down list	CHANGE_SESSION
Step Program	-	Select the Step over calls checkbox	STEP_OVER
		Do not select the Step over calls checkbox	STEP

***1**. Support for this function depends on the debugging platform.

15.5.2 Recordable functions (dependent on the debugging platform)

Macros of the recordable HEW function dependent on the debugging platform, which is included in tool packages, are recorded into macro files as scripts when the operation shown in the table below is performed. Support of this function depends on the debugging platform in use. For details, refer to the user’s manual or help files for the emulator or the simulator.

- **The emulator/simulator debugger for the SuperH RISC engine family; and the H8SX, H8S, H8 families of MCUs**

While recording a macro, if you perform in-place edit, double-click a column, click on a pop-up menu, or make any setting in a dialog box opened by clicking the pop-up menu as shown in the table below, the command in "HEW command-line commands recorded into macros" will be recorded into a macro file.

Menu	Menu Option		Keyboard Shortcuts	Toolbar Button	Recordable HEW Command-Line Commands
View	Symbol	Label	CTRL+Shift+A		See Windows.
		Watch	CTRL+W		
		Locals	CTRL+Shift+W		

- **Windows**

Window	Target	Function/Handling		Recordable HEW Command-Line Commands
Label	Pop-up menu/toolbar button	Add		SYMBOL_ADD
		Delete		SYMBOL_CLEAR
		Delete All		SYMBOL_CLEAR
		Load		SYMBOL_LOAD
	BP column	Insert/delete a breakpoint by double-clicking		SET_DISASSEMBLY_SOFT_BREAK
Watch	Pop-up menu/toolbar button	Auto Update		WATCH_AUTO_UPDATE
		Auto Update All		WATCH_AUTO_UPDATE
		Delete Auto Update		WATCH_AUTO_UPDATE
		Delete Auto Update All		WATCH_AUTO_UPDATE
		Record Update Value	Start Recording	WATCH_RECORD
			Stop Recording	WATCH_RECORD
		Add Watch		WATCH_ADD
		Edit Value		WATCH_EDIT
		Delete		WATCH_DELETE
		Delete All		WATCH_DELETE
		Radix	Hexadecimal	WATCH_RADIX
			Decimal	WATCH_RADIX
			Octal	WATCH_RADIX
			Binary	WATCH_RADIX
	+/- mark	Expand/collapse an array by clicking		WATCH_EXPAND
	Value	In-place edit		WATCH_EDIT
Locals	Pop-up menu/toolbar button	Edit		WATCH_ADD WATCH_EDIT WATCH_DELETE
	Value	In-place edit		WATCH_ADD WATCH_EDIT WATCH_DELETE

- **The simulator debugger for the SuperH RISC engine family; and the H8SX, H8S, H8 families of MCUs**

While recording a macro, if you perform in-place edit, double-click a column, click on a pop-up menu, or make any setting in a dialog box opened by clicking the pop-up menu as shown in the table below, the command in "HEW command-line commands recorded into macros" will be recorded into a macro file.

Menu	Menu Option		Keyboard Shortcuts	Toolbar Button	Recordable HEW Command-Line Commands
View	CPU	Simulated I/O	CTRL+Shift+I		See Windows.
	Code	Coverage	CTRL+Shift+O		
		Trace	CTRL+T		
		Eventpoints	CTRL+E		

- **Windows**

Window	Target	Function/Handling		Recordable HEW Command-Line Commands
Simulated I/O	Pop-up menu/toolbar button	Erase All		SIMULATEDIO_CLEAR
Coverage	Pop-up menu/toolbar button	Enable All		COVERAGE
		Clear All		COVERAGE
		Add Range		COVERAGE_RANGE
		Edit Range		COVERAGE_RANGE
		Enable		COVERAGE
		Clear Data		COVERAGE
		Save Data		COVERAGE_SAVE
		Load Data		COVERAGE_LOAD
Trace	Pop-up menu/toolbar button	Acquisition		TRACE_ACQUISITION
Event - Software Break	Pop-up menu/toolbar button	Add	PC Breakpoint	BREAKPOINT
			Break Access	BREAK_ACCESS
			Break Data	BREAK_DATA
			Break Register	BREAK_REGISTER
			Break Sequence	BREAK_SEQUENCE
			Break Cycle	BREAK_CYCLE
		Edit	PC Breakpoint	BREAK_CLEAR BREAKPOINT
			Break Access	BREAK_CLEAR BREAK_ACCESS
			Break Data	BREAK_CLEAR BREAK_DATA
			Break Register	BREAK_CLEAR BREAK_REGISTER
			Break Sequence	BREAK_CLEAR BREAK_SEQUENCE
			Break Cycle	BREAK_CLEAR BREAK_CYCLE
		Enable		BREAK_ENABLE
		Disable		BREAK_ENABLE
		Delete		BREAK_CLEAR
		Delete All		BREAK_CLEAR
Event - Software Event	Pop-up menu/toolbar button	Add	PC Breakpoint	BREAKPOINT
			Break Access	BREAK_ACCESS
			Break Data	BREAK_DATA
			Break Register	BREAK_REGISTER
			Break Sequence	BREAK_SEQUENCE
			Break Cycle	BREAK_CYCLE
		Edit	PC Breakpoint	BREAK_CLEAR BREAKPOINT
			Break Access	BREAK_CLEAR BREAK_ACCESS
			Break Data	BREAK_CLEAR BREAK_DATA
			Break Register	BREAK_CLEAR BREAK_REGISTER
			Break Sequence	BREAK_CLEAR BREAK_SEQUENCE
			Break Cycle	BREAK_CLEAR BREAK_CYCLE
		Enable		BREAK_ENABLE
		Disable		BREAK_ENABLE
		Delete		BREAK_CLEAR
		Delete All		BREAK_CLEAR

15.6 Playing a macro

There are a number of ways to play a macro file in the HEW, these options are discussed below.

To play a macro from the toolbar or menu

1. Click the **Play Macro** button (▷) or menu item on the **Macros** toolbar or the Tools menu.
2. If there is only one macro defined it is automatically played. If multiple macro functions exist in the current default macro file a dialog is displayed that asks you to choose which macro you wish to execute.
3. When playing the mouse cursor is modified to include the play icon (↖▷).
4. Click OK and the macro is executed. When the macro is being executed the pause and stop buttons (⏸⏹) and menus are enabled.

It is also easy to run a macro from the macro toolbar, menu or via a keyboard shortcut if you have set up macro function assignments in the Macro dialog box. For information on this area read the "Assigning a macro" information.

Another method is to launch the macro function via the Macro dialog box.

To play a macro from the macro dialog box

1. Select the [**Tools->Macros**] menu option. The **Macro** dialog box is displayed.
2. Select the macro file which contains the function you want to play.
3. Select the macro function in the function list that you want to play.
4. Click the **Play** button.
5. The Macro dialog box is dismissed and the macro function executed.

15.7 Editing a macro

The Macros are stored in the "Macros" folder in the HEW root folder. These files are just text based TCL files that can also be manually edited in the HEW editor window. Any changes made will automatically be picked up by the HEW next time the macro is executed.

It is also possible to edit macros in the following way.

To edit a macro

1. Select the [**Tools->Macros**] menu option. The **Macro** dialog box is displayed.
2. Select the macro file which contains the function you want to edit.
3. Select the macro function in the **Macro functions** that you want to edit.
4. Click the **Edit** button.
5. The **Macro** dialog box is dismissed and the file opened in the HEW editor.

15.8 Assigning a macro

It is possible to assign a macro to either a custom menu item, toolbar or keyboard shortcut. This allows you to quickly access macros which you often use. It is possible to assign a single macro to a **Tools** menu, **Macros** toolbar and keyboard shortcut all at the same time if you want. A custom menu and a toolbar button are shown in alphabetical order.

To assign a macro to a toolbar button

1. Select the [**Tools->Macros**] menu option. The **Macro** dialog box is displayed.
2. Select the macro file which contains the function you want to assign.
3. Select the macro function in the **Macro functions** list that you want to assign.
4. Click the **Assign** button. The **Macro assign** dialog box is displayed.
5. Select the **Assign to Toolbar** check box.
6. You can now modify the macro description in the **Description** field. This will be used as the toolbar tool tip if you assign the macro to a toolbar button.
7. Now you can decide whether you will use the default macro toolbar button () or specify your own toolbar button image from an existing bitmap file. (*.bmp)
8. To specify your own image simply click the browse button to open a standard file browser. This will allow you to manually locate the file on your machine.
9. Click OK and the toolbar is added to the **Macros** toolbar.

To assign a macro to a menu

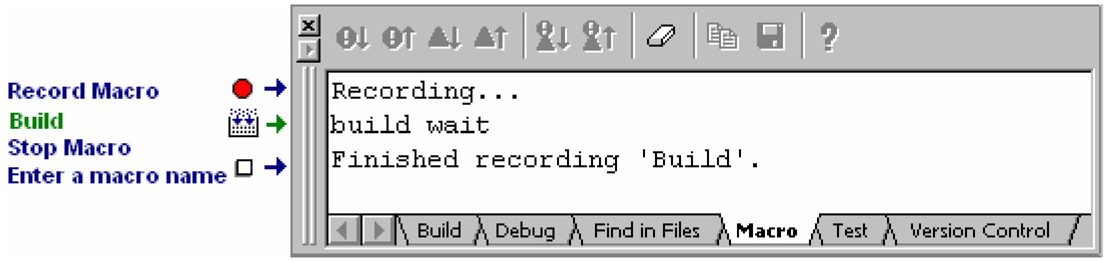
1. Select the [**Tools->Macros**] menu option. The **Macro** dialog box is displayed.
2. Select the macro file which contains the function you want to assign.
3. Select the macro function in the **Macro functions** list that you want to assign.
4. Click the **Assign** button. The **Macro assign** dialog box is displayed.
5. Select the **Assign to Menu Name** check box.
6. You can now modify the macro description in the **Description** field. This will be used as the menu tool tip if you assign the macro to menu.
7. Enter the menu name as you want it to appear on the tools menu.
8. Click OK and the menu is added to the **Tools** menu.

To assign a macro to a keyboard shortcut

1. Select the [**Tools->Macros**] menu option. The **Macro** dialog box is displayed.
2. Select the macro file which contains the function you want to assign.
3. Select the macro function in the **Macro functions** list that you want to assign.
4. Click the **Assign** button. The **Macro assign** dialog box is displayed.
5. Select the **Assign to Keyboard Shortcut** check box.
6. Select the keyboard shortcut you wish to assign to the macro in the drop-down list.
7. Click OK and the shortcut is now available for use.

15.9 Configuring the Macro tab of the Output window

Shows the current records of macros. You can view information such as HEW command-line commands recorded into a HEW macro file from execution of [**Tools->Macro Recording**] to [**Tools->Stop Macro**]. It is also possible to view this information while recording.



```
Recording...
build wait
Finished recording 'Build'.
```

Right-clicking displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up Menu Options	Toolbar button	Function
Clear Window	⌫	Clears the contents of the window.
Save	💾	Saves the contents of the window into a text file.
Copy	📋	Copies the selected contents onto the Windows® clipboard.
Toolbar display	-	Shows or hides the toolbar.
Customize toolbar	-	Customizes toolbar buttons.

16. Using the Test Support Facility

The High-performance Embedded Workshop (HEW) provide regression testing facilities for your application by the test support facility. Regression test is re-execution of a previous test after modification of the program to verify the result. The test tool allows HEW macro and batch files to be executed and the system can then be compared to see if it matches the original captured data.

It is possible to create many tests and scripts and automate the testing procedure. Then once the testing has been completed the results are displayed in an easy to understand format in the test browser window. This can significantly reduce the work needed to execute many regression tests.

The test tool uses the TCL command line batch files as its method of executing the features and functions of HEW. These functions can be created easily using the macro recorder feature or by manually editing them yourself.

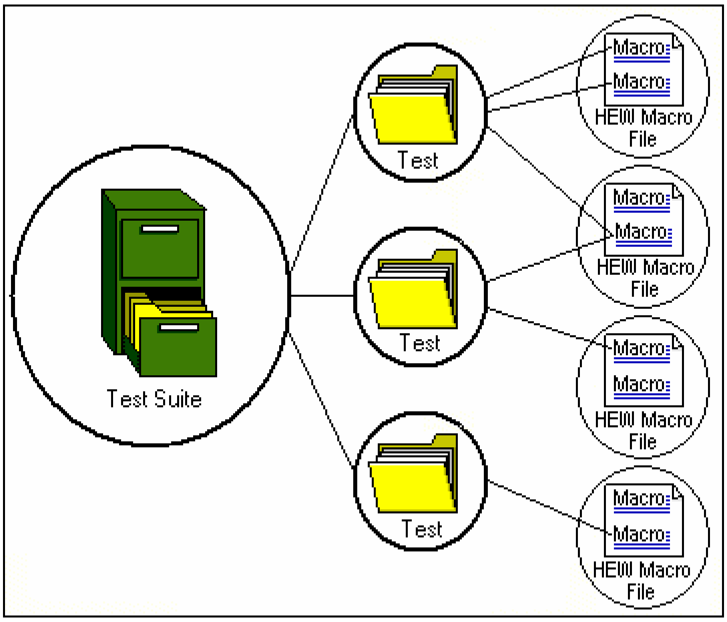
If you need to automate the execution of the actual tests the test tool also supports command line operation. Many commands are available that allow you to open and close suites, run and compare test data.

The first step towards using the test tool is to create a test suite. This is available on the main test menu. Then you must edit the test suite to create some actual tests. Then you can execute the tests using the "Run tests" menu item.

Note, however, that HEW is not capable of acquiring test-image information of all functions in the HEW system. For the items from which test-image information can be acquired, see section 16.6, Functions that can be saved as test-image data into test-image files.

16.1 Creating a test suite

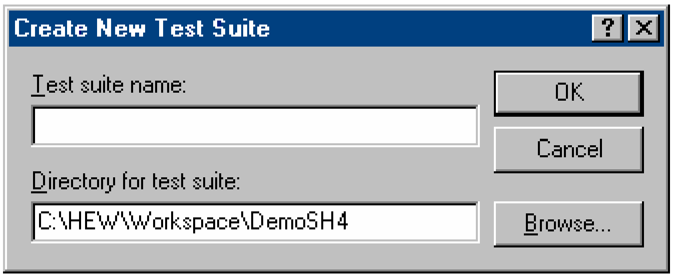
A test suite is a set of tests. A test suite has a very similar concept to a workspace. The test suite can contain may tests which in turn can contain many macros to be executed.



Test suites are independent to the current HEW workspace. This allows your test suite to work with multiple workspaces and projects to test different situations in one test suite. The "open_workspace" and "change_project" commands all work with the test suite feature so allow you to control which workspace is currently in use.

To create a test suite

1. Select the [**Test->Create New Test Suite**] menu option. The **Create New Test Suite** dialog box is displayed.



C:\HEW\Workspace\DemoSH4

2. Enter the test suite name.
3. Initially the workspace directory is shown in **Directory for test suite**. This can be modified as required.
4. Clicking OK.
5. The test suite is then created. This then enables a number of other items on the **Test** menu.

Once the test suite is created the Test tab of the workspace window has the test suite added to it. This tab allows quick navigation around your test suite and fast access to the tests.

A file is located in the destination location with the filename ".HTS" (HEW test suite).

16.2 Opening and closing test suites

Once a test suite is created it is saved to a file with the extension ".HTS" (HEW test suite). To open this file again after you have created it previously you can use the following operation.

To open a test suite

1. Select the [**Test->Open Test Suite**] menu option. The Open Test Suite dialog box is displayed.
2. Select the test suite file. Then click Select. The test suite is loaded.
3. This then enables a number of other items on the **Test** menu and opens the test suite contents in the Test tab of the workspace window.

When test suites are opened the filename of the suite is added to the **Recently used test suite** list on the **File** menu. This allows access to open your test suites after the initial **Open test Suite** menu operation.

To close a test suite

Select the [**Test->Close Test Suite**] menu option. Once clicked the current test suite is closed and all items are removed from the workspace tab.

There are TCL command line commands for these operations that can be used via the command line window. These commands are named "open_test_suite" and "close_test_suite".

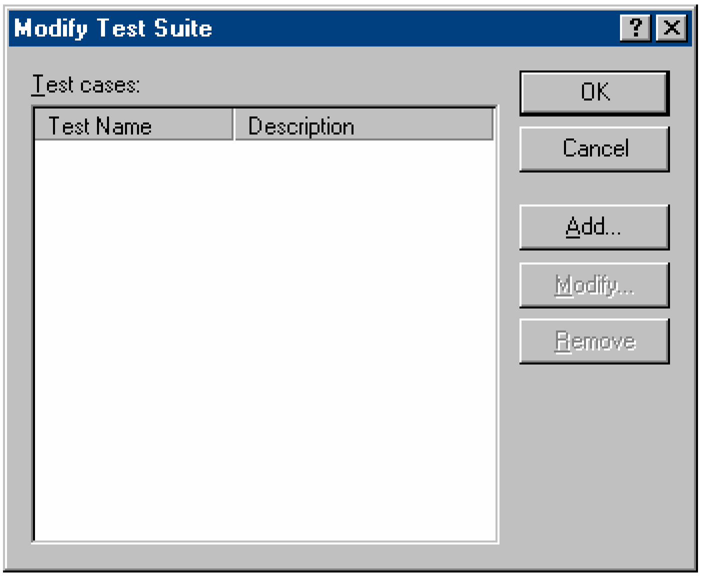
You can also close a test suite from the workspace window pop-up menu.

16.3 Editing a test suite

Once your test suite has been created the next step is to add some tests. This operation allows you to access the **Modify Test Suite** dialog box and will allow you to add tests which will then be executable.

To edit a test suite

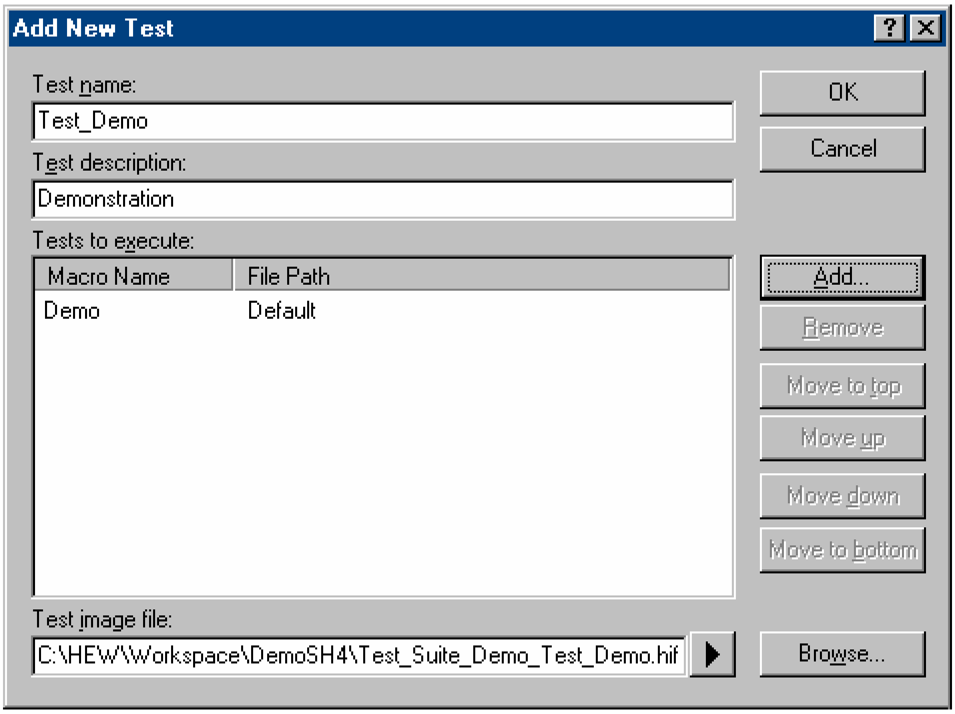
1. Select the [**Test->Edit Test Suite**] menu option. The **Modify Test Suite** dialog box is displayed.



Test Name	Description

The **Modify Test Suite** dialog box displays the currently defined test cases. Each test case has a name and can have a detailed description that you have defined to explain the tests purpose.

Clicking the **Add** button on this dialog box displays the **Add New Test** dialog box, this is shown below.



Test name: Test_Demo

Test description: Demonstration

Tests to execute:

Macro Name	File Path
Demo	Default

Test image file: C:\HEW\Workspace\DemoSH4\Test_Suite_Demo_Test_Demo.hif

Selecting the test and clicking modify allows you to edit currently defined details for the selected test. These details are also shown in the same dialog as that used for "**Add new test**". If you click **Remove** the test is removed.

The **Add New Test** dialog box allows you to configure tests for execution later. This is discussed in "Adding tests to the test suite".

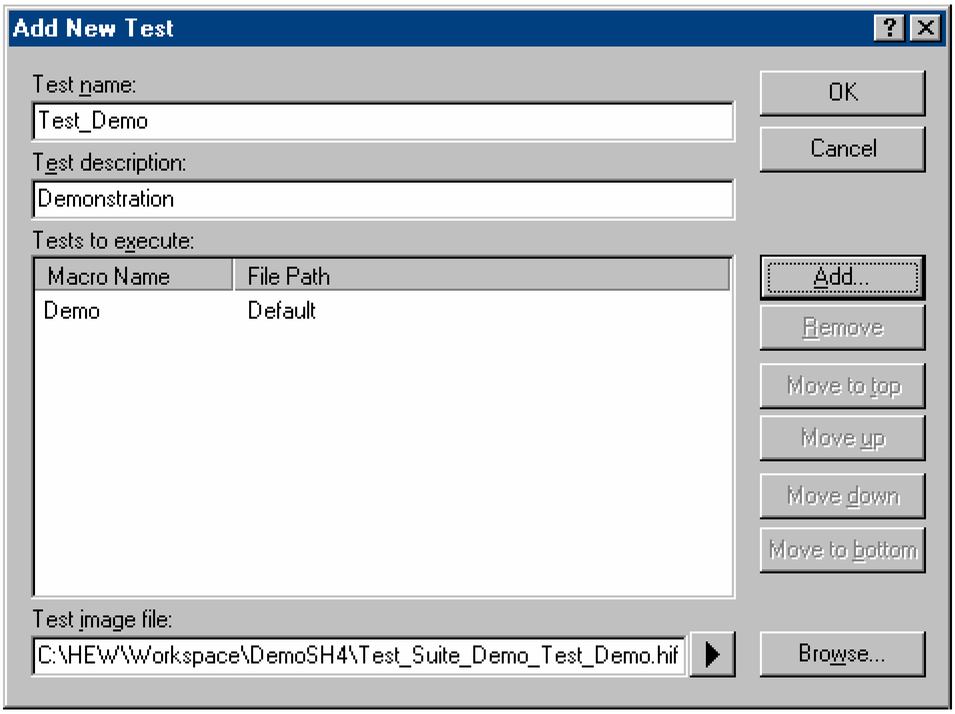
You can also edit a test suite from the workspace window pop-up menu.

16.4 Adding tests to the test suite

Once your test suite has been created the next step is to add some tests. This operation allows you to access the **Modify Test Suite** dialog box and will allow you to add tests which will then be executable.

To add tests to a test suite

1. Select the [**Test->Edit Test Suite**] menu option. The **Modify Test Suite** is then displayed.
2. Click **Add**. The **Add New Test** dialog box is displayed.



Test name: Test_Demo

Test description: Demonstration

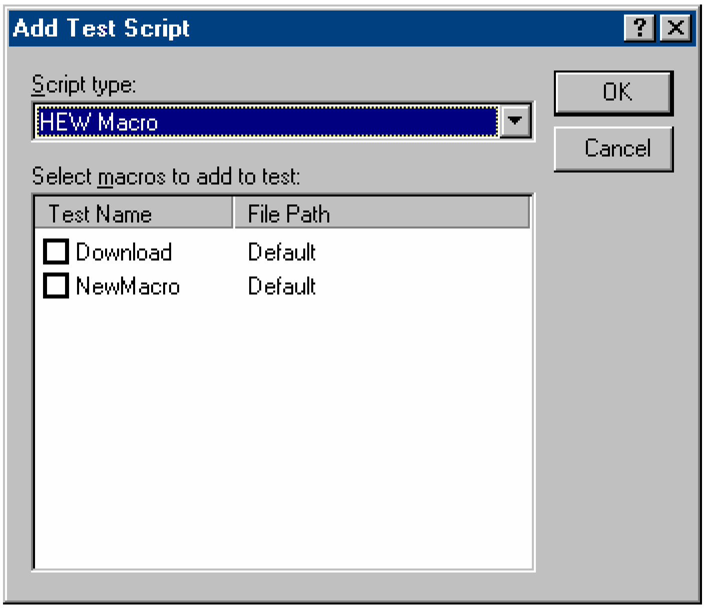
Tests to execute:

Macro Name	File Path
Demo	Default

Test image file: C:\HEW\Workspace\DemoSH4\Test_Suite_Demo_Test_Demo.hif

To add a new test you should setup the following data

1. Enter the test name. There can be no spaces in this name.
2. Enter the test description. This should describe the test in a verbose way so you will understand it at a later date.
3. To setup the actual tests to execute you should click the **Add** button.
4. The **Add New Test Script** dialog box is displayed (See below).
5. The **Script type** box allows you to select a HEW macro ("HEW macro") or a HEW command line batch file ("TCL command line batch file").
6. Selecting **HEW Macro** in the **Script type** drop-down list shows all of the registered macros in the **Select macros to add to test** list.
7. You can select multiple macro files to be executed for each test. This is achieved by selecting the checkbox next to the macro names you wish to use.
8. Selecting "TCL command line batch file" in the drop-down list changes the **Add Test Script** dialog box so you can define the file to execute when the test is executed.
9. Clicking OK stores the results and adds it to the **Add New Test** dialog box.
10. Finally you must set the **Debug system for test image comparison** file. This file stores the comparison data for the system to be compared to after the tests have executed. This file can be created by using the [**Test->Create Test Image File**] menu option.
11. Clicking OK adds the test to the **Modify Test Suite** dialog box.



Add Test Script

Script type: HEW Macro

Select macros to add to test:

Test Name	File Path
□ Download	Default
□ NewMacro	Default

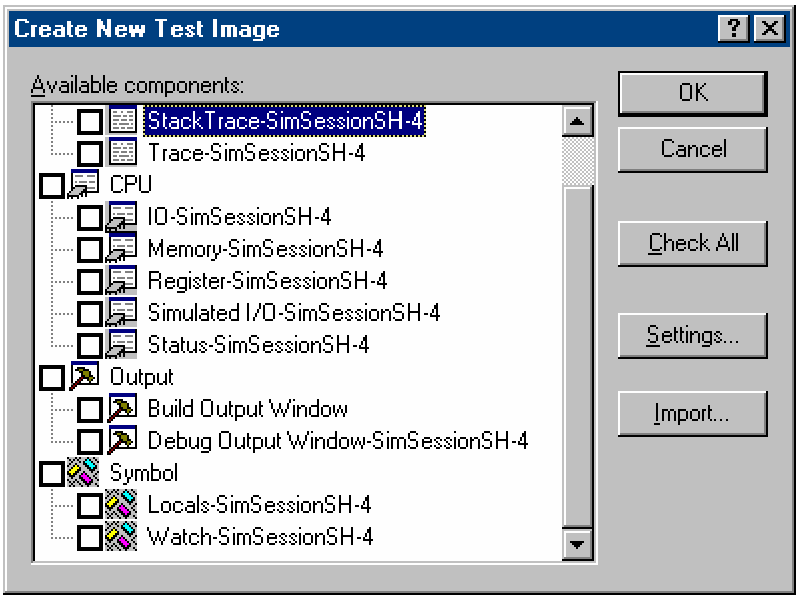
16.5 Creating a test image file

The test image data is what is used for any comparison in the test system. Only items that are saved in the test file can be compared.

For example you may only be interested in comparing data that is located in a small area memory not the memory for the entire device. The more items that you compare for each test will slow down the comparison and this can make a big difference if you are executing many tests.

To create test image data to be saved into a test image file

1. Select the [**Test->Create New Test Image File**]. The **Create New Test Image** dialog box is then displayed.



Create New Test Image

Available components:

- □ StackTrace-SimSessionSH-4
- □ Trace-SimSessionSH-4
- □ CPU
  - □ IO-SimSessionSH-4
  - □ Memory-SimSessionSH-4
  - □ Register-SimSessionSH-4
  - □ Simulated I/O-SimSessionSH-4
  - □ Status-SimSessionSH-4
- □ Output
  - □ Build Output Window
  - □ Debug Output Window-SimSessionSH-4
- □ Symbol
  - □ Locals-SimSessionSH-4
  - □ Watch-SimSessionSH-4

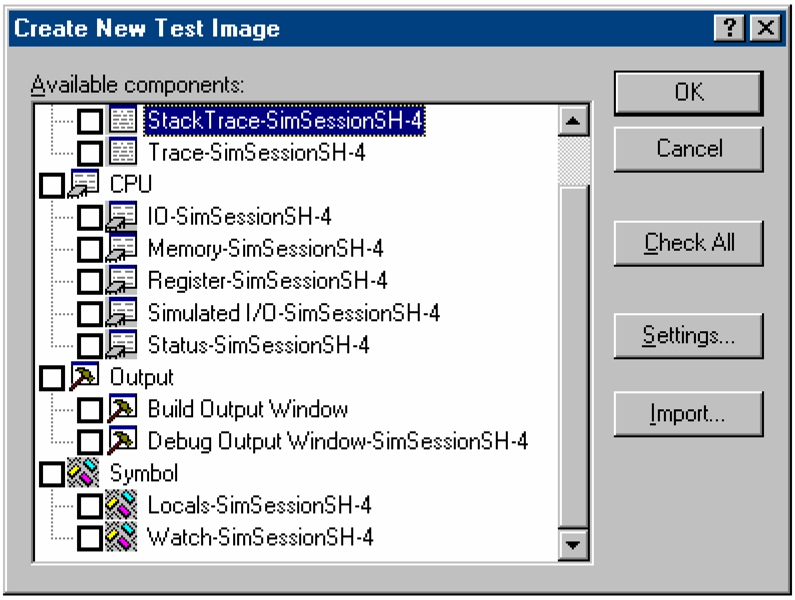
2. Clicking the **Check All** button selects all checkboxes, while clicking the **Uncheck All** button deselects all checkboxes.

3. Click the **Import** button to open the **Import the Test Image File** dialog box. Browse to the HIF file location. The settings of an existing test image file is imported.

4. Each component is listed in the dialog box. Check the checkbox next to the component name to save data for that component.
5. Select the component in the dialog and then click the **Settings** button. A dialog box will be displayed that is dependent on the selected component. This will allow you to customize the data that is saved to the file. For details, see section 16.6, Functions that can be saved as test-image data into test-image files.
6. Clicking OK will dismiss the dialog and store the changes that you have set up.
7. The **Save Test Image File** dialog box is displayed.
8. Save the data into the HEW test-image file selected in the **Add New Test** dialog box or a new HEW test-image file. Only test-image data of the selected test items (with tick marks in checkboxes) will be saved into the file.

You can also edit a test image file from the workspace window pop-up menu.

16.6 Functions that can be saved as test-image data into test-image files



HEW is not capable of saving all HEW functions as test-image data. For details on the functions that can be saved into test-image files, see the topics below. These topics also include information about failed test items (FAIL) shown after test or at comparison of test-image files.

- Functions that can be saved into test-image files (common to all HEW products)
- Functions that can be saved into test-image files (dependent on the debugging platform)

In some cases it is necessary to make detailed setting specific to each of the test items before saving test-image data. If you double-click on a test item in the **Create New Test Image** dialog box, a further dialog box for detailed setting opens. When a test item has any detailed setting, it is possible to customize test-image data of this test item before saving. On the other hand, when a test item does not allow detailed setting and the checkbox for this test item is selected, all test-image data of this item will be saved into the file.

A save file icon (💾) in the "Saving into Test-Image File" column of an item listed in the **View** menu indicates that this data can be saved into a test-image file.

16.6.1 Functions that can be saved into test-image files (common to all HEW products)

Among the functions common to all HEW products, the test items listed below can be saved into test-image files. The following table shows these test items in the **Create New Test Image** dialog box.

Menu	Menu Option		Keyboard Shortcuts	Toolbar Button	Window	Create New Test Image dialog box		
						Test group	Test item	Whether detailed setting specific to the test item can be made
View	Output		AIL+O		Build tab of the output window	Output	Build Output Window	No
					Debug tab of the output window		Debug Output Window	
	CPU	Registers	CTRL+R	R1	Register	CPU	Register *	
		IO	CTRL+I	I/O	IO		IO *	Yes
		Status	CTRL+U		Status		Status *	
		Memory	CTRL+M		Memory		Memory *	
	Code	Stack Trace	CTRL+K		Stack Trace	Code	Stack Trace *	

*. Test items cannot be selected when the HEW is not connected to a debugging platform.

When detailed setting specific to the test item cannot be made

- Output - Build Output Window

Test-image data to be saved into test-image files	All information in the Build tab of the output window.	
Test result details	Original	Content of the test-image file.
	New	Data in the current HEW system or in another test-image file to be compared with.
	Example	**Original**: **New**: Building - New - Debug

- Output - Debug Output Window

Test-image data to be saved into test-image files	All information in the Debug tab of the output window.	
Test result details	Original	Content of the test-image file.
	New	Data in the current HEW system or in another test-image file to be compared with.
	Example	**Original**: Connected **New**: Step Normal End

- Output - Register

Test-image data to be saved into test-image files	All information in the Register window.	
Test result details	Failed at register	Name of the unmatched register.
	Src	Content of the test-image file.
	Dest	Data in the current HEW system or in another test-image file to be compared with.
	Example	**Failed at register** R11, **Src** = 0x00000000, **Dest** = 0x00000fff

When detailed setting specific to the test item can be made

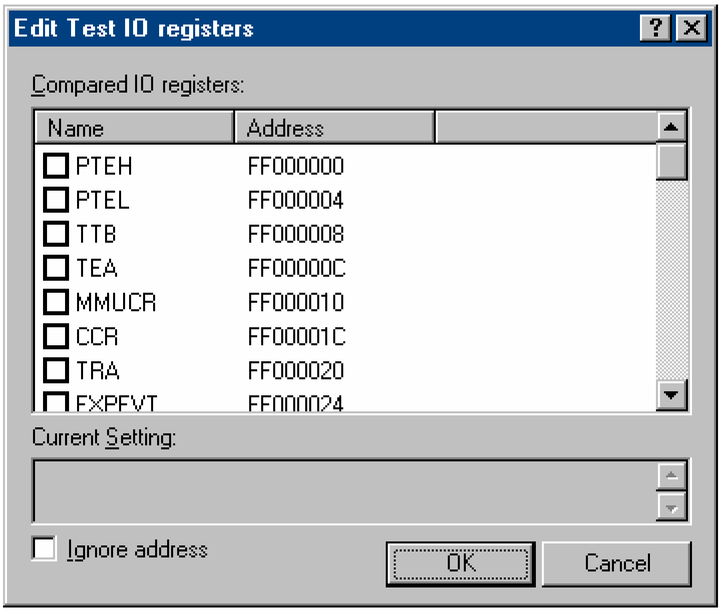
If you double-click on a test item in the **Create Test Image** dialog box, a further dialog box for detailed setting opens.

- CPU - IO

Test-image data to be saved into test-image files	Values set as detailed information and the range of data in the **IO** window acquired with this setting. No I/O register is specified by default.	
Test result details	Failed at	Name of the unmatched I/O register.
	Src	Content of the test-image file.
	Dest	Data in the current HEW system or in another test-image file to be compared with.
	Example	**Failed at** IPRC, register value is different: **Src** = 0000, **Dest** = FFFF

To make detailed setting

1. Opens the **Edit Test IO registers** dialog box.



Name	Address
☐ PTEH	FF000000
☐ PTEL	FF000004
☐ TTB	FF000008
☐ TEA	FF00000C
☐ MMUCR	FF000010
☐ CCR	FF00001C
☐ TRA	FF000020
☐ EXPEVT	FF000024

2. Selecting the checkbox for an I/O register in the **Compared I/O registers** list allows this I/O register to be saved into the test-image file. These checkboxes are not selected by default. The selected I/O register is shown in **Current Setting**.

3. If you do not wish to compare the address of the selected register, select the **Ignore address** checkbox. This checkbox is not selected by default.

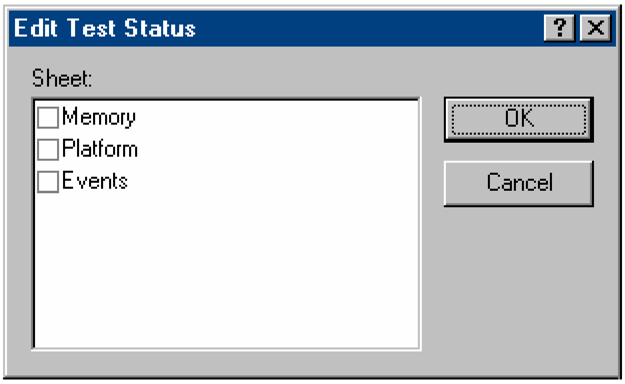
4. Click OK.

- CPU - Status

Test-image data to be saved into test-image files	Values set as detailed information and all data in the **Status** window.	
Test result details	Sheet	Name of the unmatched sheet
	Line	Unmatched line number
	Src	Content of the test-image file
	Dest	Data in the current HEW system or in another test-image file to be compared with.
	Example	Status differs. **Sheet** = Platform **Line** = 6 **Src** = Execute From, Pipeline Reset **Dest** = Execute From, EX Stage

To make detailed setting

1. Opens the **Edit Test Status** dialog box.



Edit Test Status

Sheet:

□ Memory
□ Platform
□ Events

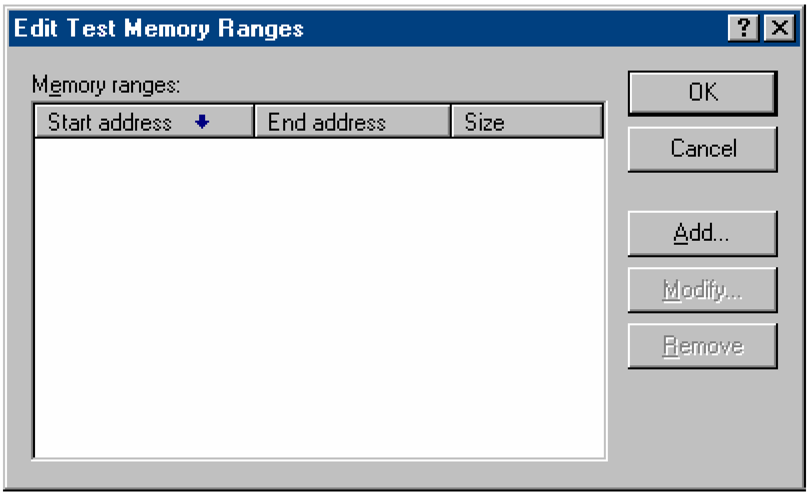
2. To specify the sheet to be compared in the test-image file, select the **Memory**, **Platform**, or **Events** checkbox. These checkboxes are not selected by default.
3. Click OK.

- CPU - Memory

Test-image data to be saved into test-image files	Values set as detailed information and the range of data in the **Memory** window acquired with this setting. No memory range is specified by default.	
Test result details	Failed at address	Unmatched address.
	Src	Content of the test-image file.
	Dest	Data in the current HEW system or in another test-image file to be compared with.
	Example	**Failed at address** 0x70000014, **Src** = 0x00002f5a, **Dest** = 0x00002704

To make detailed setting

1. Opens the **Edit Test Memory Range** dialog box.

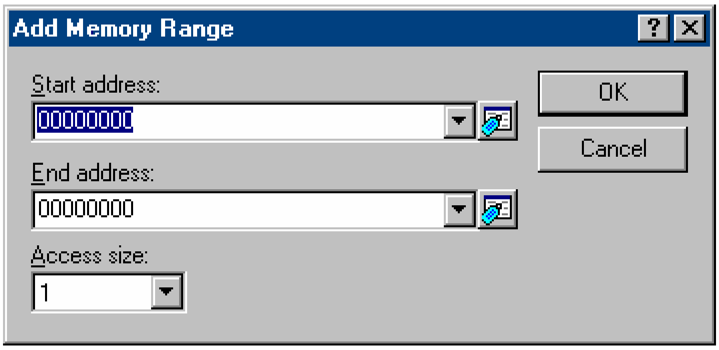


Edit Test Memory Ranges

Memory ranges:

Start address	End address	Size

2. Click **Add** button in the Edit Test Memory Ranges dialog box.
3. The **Add Memory Range** dialog box is displayed.



Add Memory Range

Start address: 00000000

End address: 00000000

Access size: 1

4. Specify the memory range by **Start address**, **End address**, and **Access size**.
5. Click OK.

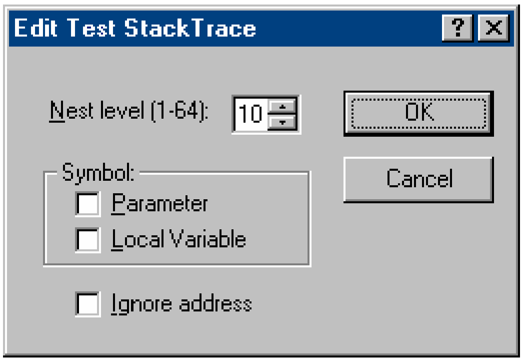
- Code - StackTrace

Test-image data to be saved into test-image files	Values set as detailed information and data within nested function calls in the **Stack Trace** window. The default value in **Nest level** is 10.	
Test result details	Src	Content of the test-image file
	Dest	Data in the current HEW system or in another test-image file to be compared with.

	Example	Value differs.
		Src = F, PowerON_Reset_PC(), { 0000080E }
		Dest = F, PowerON_Reset_PC(), { 0000081C }

To make detailed setting

1. Opens the **Edit Test StackTrace** dialog box.



Edit Test StackTrace

Nest level (1-64): 10

Symbol:

☐ Parameter

☐ Local Variable

☐ Ignore address

2. The range of stack trace information to be saved into the test-image file is determined by the number of nest levels specified in **Nest level**.
3. To compare parameters or local variables, select checkbox **Parameter** or **Local Variable**. These checkbox are not selected by default.
4. If you do not wish to compare the address of the selected register, select the **Ignore address** checkbox. This checkbox is not selected by default.
5. Click OK.

16.6.2 Functions that can be saved into test-image files (dependent on the debugging platform)

Among the HEW functions dependent on the debugging platform, the test items listed below can be saved into test-image files.

- **The emulator/simulator debugger for the SuperH RISC engine family; and the H8SX, H8S, H8 families of MCUs**

The following table shows these test items in the **Create New Test Image** dialog box. Test items cannot be selected when the HEW is not connected to a debugging platform.

Menu	Menu Option		Keyboard Shortcuts	Toolbar Button	Window	Create New Test Image dialog box		
						Test group	Test item	Whether detailed setting specific to the test item can be made
View	Code	Trace	CTRL+T		Trace	Code	Trace	Yes
	Symbol	Locals	CTRL+Shift+W		Locals	Symbol	Locals	
		Watch	CTRL+W		Watch		Watch	

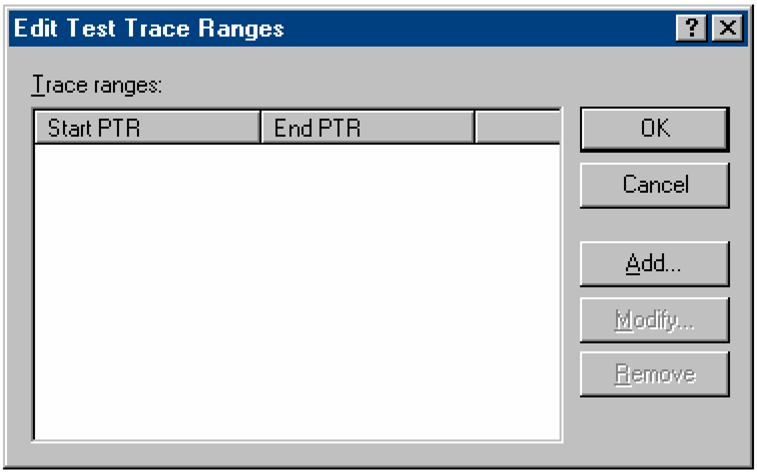
If you double-click on a test item in the **Create Test Image** dialog box, a further dialog box for detailed setting opens.

- Code - Trace

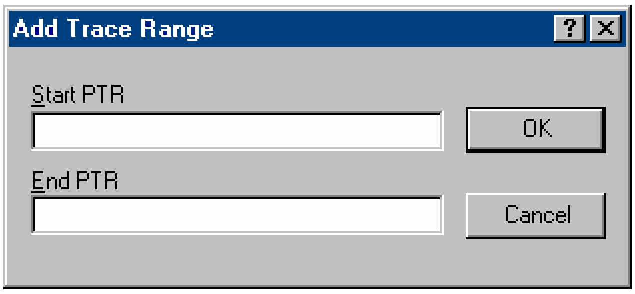
Test-image data to be saved into test-image files	Values set as detailed information and the range of data in the **Trace** window acquired with this setting. No trace range is specified by default.	
Test result details	Comparing PTR	Compared trace range.
	Trace type	Trace type
	Trace data is not matching. PTR	Selected pointer.
	Src	Content of the test-image file.
	Dest	Data in the current HEW system or in another test-image file to be compared with.
	Example	**Comparing PTR** -3 to -1. **Trace type** : Trace. **Trace data is not matching. PTR**:-3 to PTR:-1 Trace data at the beginning of difference. PTR:-3 **Src** : 0000001879 0000107C F->DEMMW MOV.L @R15+, R1 R1<-00000010 **Dest**: 0000001887 00000818 FD<E JSR @R5 PC<-00001000

To make detailed setting (with one trace type)

1. Opens the **Edit Test Trace Ranges** dialog box.



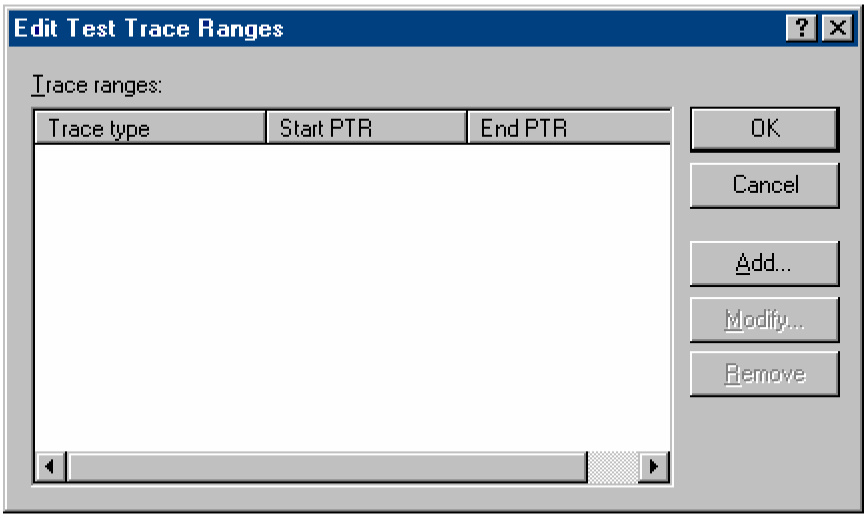
2. Click **Add** button in the Edit Test Trace Ranges dialog box.
3. The **Add Trace Range** dialog box is displayed.



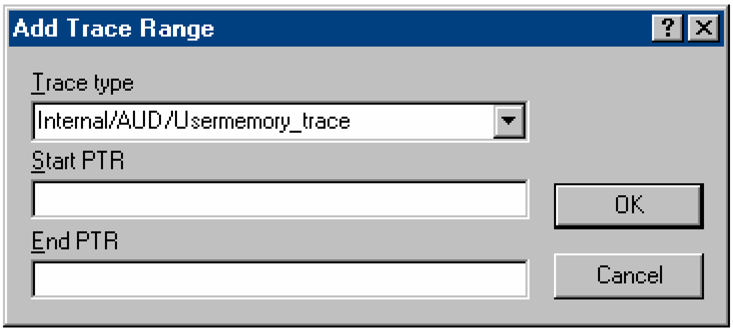
4. Specify the trace range by **Start PTR**, and **End PTR**.
5. Click OK.
6. Select the **Edit** or **Remove** button in the **Edit Test Trace Range** dialog box as required.
7. Click OK.

To make detailed setting (with two or more trace types)

1. Opens the **Edit Test Trace Ranges** dialog box.



2. Click **Add** button in the Edit Test Trace Ranges dialog box.
3. The **Add Trace Range** dialog box is displayed.



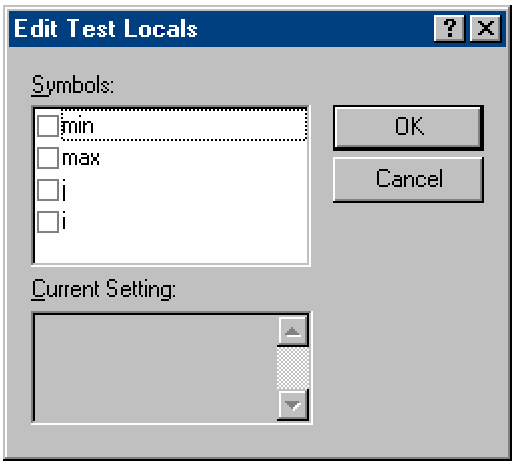
4. Specify the trace range by **Trace type**, **Start PTR**, and **End PTR**.
5. Click OK.
6. Select the **Edit** or **Remove** button in the **Edit Test Trace Range** dialog box as required.
7. Click OK.

- Symbol - Locals

Test-image data to be saved into test-image files	Values set as detailed information and the range of data in the **Locals** window acquired with this setting. Checkboxes for all symbols are blank by default.	
Test result details	Failed at symbol	Unmatched symbol name.
	Src	Content of the test-image file.
	Dest	Data in the current HEW system or in another test-image file to be compared with.
	Example	**Failed at symbol** i, **Src** = (int)H'00000001{R14}, **Dest** = (int)H'00000000{R14}

To make detailed setting

1. Opens the **Edit Test Locals** dialog box.



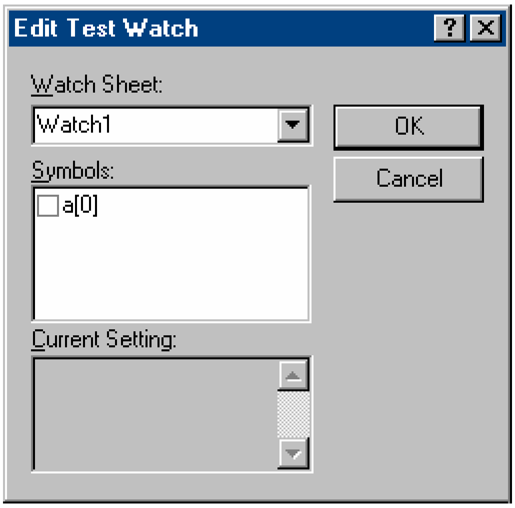
2. To specify a symbol to be saved into the test-image file, select the checkbox for that symbol. The selected symbol is shown in **Current Setting**.
3. Click OK.

- Symbol - Watch

Test-image data to be saved into test-image files	Values set as detailed information and the range of data in the **Watch** window acquired with this setting. Checkboxes for all symbols are blank by default.	
Test result details	Failed at symbol	Unmatched symbol name.
	Src	Content of the test-image file.
	Dest	Data in the current HEW system or in another test-image file to be compared with.
	Example	**Failed at symbol** a[6], **Src** = (long)H'00002704{70000018}, **Dest** = (long)H'00000daa{70000018}

To make detailed setting

1. Opens the **Edit Test Watch** dialog box.



Edit Test Watch

Watch Sheet: Watch1

Symbols:
□ a[0]

Current Setting:

2. To specify a symbol to be saved into the test-image file, select **Watch Sheet** and the checkbox for that symbol. The selected symbol is shown in **Current Setting**.
3. Click OK.

- **The simulator debugger for the SuperH RISC engine family; and the H8SX, H8S, H8 families of MCUs**

The following table shows these test items in the **Create New Test Image** dialog box. Test items cannot be selected when the HEW is not connected to a debugging platform.

Menu	**Menu Option**		**Keyboard Shortcuts**	**Toolbar Button**	**Window**	**Create New Test Image dialog box**		
						Test group	**Test item**	**Whether detailed setting specific to the test item can be made**
View	CPU	Simulated I/O	Ctrl+Shift+I	▣	Simulated I/O	CPU	Simulated I/O	Yes
	Code	Coverage	Ctrl+Shift+O	▦	Coverage	Code	Coverage	

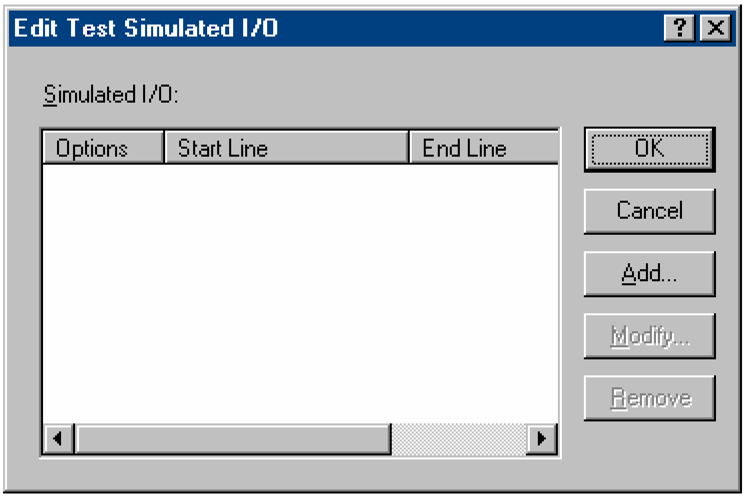
If you double-click on a test item in the **Create Test Image** dialog box, a further dialog box for detailed setting opens.

- CPU - Simulated I/O

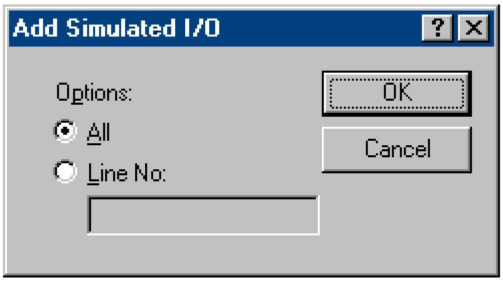
Test-image data to be saved into test-image files	Values set as detailed information and the range of data in the **Simulated I/O** window acquired with this setting. No I/O simulation range is specified by default.	
Test result details	Failed at Line	Unmatched line number
	Src	Content of the test-image file.
	Dest	Data in the current HEW system or in another test-image file to be compared with.
	Example	**Failed at Line** 1: **Src** = ### Data Input ### **Dest** = a[1]=21468

To make detailed setting

1. Opens the **Edit Test Simulated I/O** dialog box.



2. Click **Add** button in the Edit Test Simulated I/O dialog box.
3. The **Add Simulated I/O** dialog box is displayed.



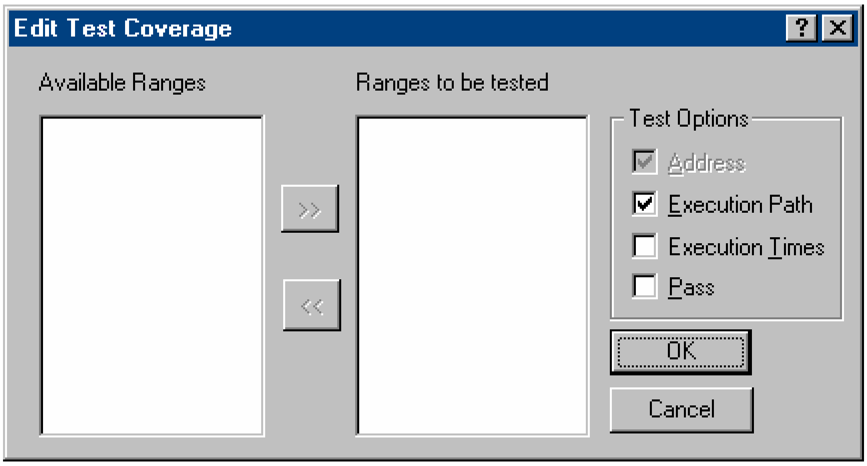
4. Specify the I/O simulation range by **All** or **Line No**.
5. Click OK.
6. Select the **Edit** or **Remove** button in the Edit Test Simulated I/O dialog box as required.
7. Click OK.

- Code - Coverage

Test-image data to be saved into test-image files	Values set as detailed information and the range of data in the **Simulated I/O** window acquired with this setting. No I/O simulation range is specified by default.	
Test result details	Range	Coverage range.
	Instruction Execution	Unmatched test option.
	mismatch at	Unmatched address
	Image	Content of the test-image file.
	System	Data in the current HEW system or in another test-image file to be compared with.
	Example	**Range** DemoSH4.c: **Instruction Execution** Time **mismatch at** 0x20E4 -- **Image**: 1 **System**: 0

To make detailed setting

1. Opens the **Edit Test Coverage** dialog box.



2. **Available Ranges** shows the coverage ranges that are currently available.

3. If you select coverage ranges in **Available Ranges** and click the >> button, these coverage ranges then appear in **Ranges to be tested**.

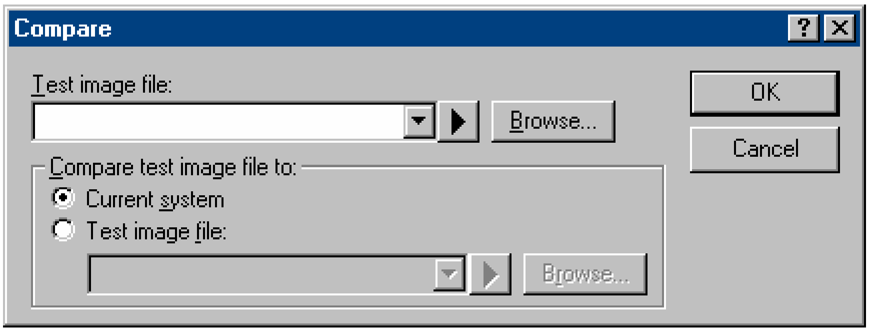
4. The **Address** checkbox among test options is always ON because addresses must be saved whenever coverage ranges are saved into test-image files (users cannot control this selection). To set other test options **Execution Path**, **Execution Times**, and **Pass**, select the checkboxes for these options. The **Execution Path** checkbox is selected by default. Other checkboxes are not selected by default.

5. Click OK.

16.7 Comparing a test image file

To compare a test image file

1. Select the [**Test->Compare Test Image File**] menu option. The **Compare** dialog box is displayed.



2. Enter the test file location that you want to compare in **Test image file** field. This is a "*.HIF" (HEW test image file) file and contains details about the test image data.

3. Then select what you want to compare the file you have selected with. You can choose either the Current system or another previously saved test image file. The **Current system** option is useful if you have manually executed a script and want to check the current test image data with some you saved previously.

4. Click OK.

5. The results will be loaded into the test browser.

It is possible to compare test image file separately to test execution. This allows you to compare manually the current system or two test image files at a later date to when the tests were actually executed.

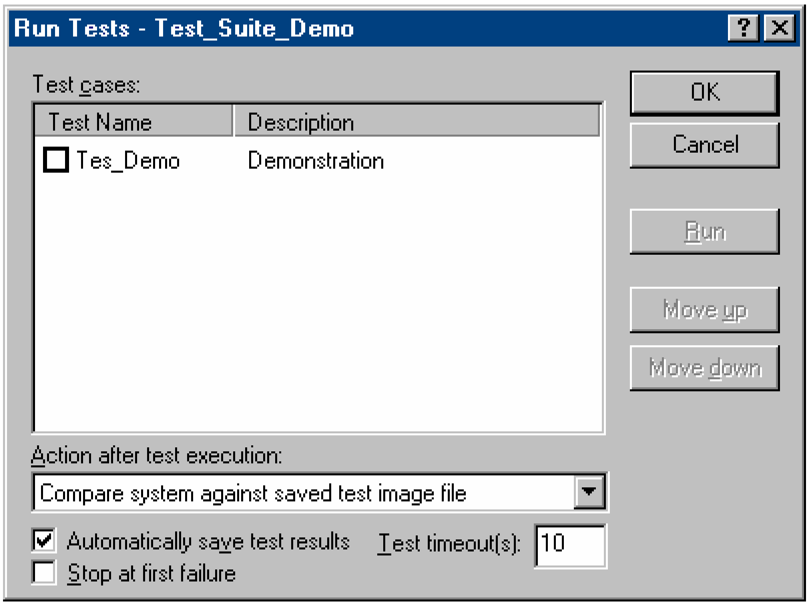
You can also compare test image file from the workspace window pop-up menu.

16.8 Running tests

This allows you to select tests and automate their execution.

To run tests

1. Select the [**Test->Run Tests**] menu option. The **Run Tests** dialog box is displayed.



Run Tests - Test_Suite_Demo

Test cases:

Test Name	Description
□ Tes_Demo	Demonstration

Action after test execution: Compare system against saved test image file

☑ Automatically save test results

Test timeout(s): 10

□ Stop at first failure

2. All tests currently defined in the HEW test system are listed in the **Test cases** list.
3. Clicking the check box selects the test for execution in this test run.
4. It is also possible to modify the order a test is executed in by selecting the test and clicking the **Move up** and **Move down** buttons.
5. When one or more tests have been selected the **Run** button is enabled. Clicking this **Run** button starts the test run. Information will be shown in **Test** tab of the output window as the tests are executed.
6. Once completed the test browser will be displayed and will show the test results for all tests executed.

The **Action after test case execution** drop-down list allows you two options.

The "Compare system against saved test image file" is the normal operation and allows the current HEW system to be compared with the test image file (*.HIF) file that was attached to the relevant test case. These results will then be added to the test browser and will give pass, fail and failure details information to you.

The "Refresh test image file" option simply runs all of the test cases and updates the test image file (*.HIF) file with the system image after each test.

The **Automatically save test results** check box allows you to automatically save the results of each test execution to a text file in the same directory as the test suite. The filename used is made up from the current test suite and the date of time of the test execution.

The **Stop at first failure** will stop the automated test execution when the first test failure is met. This avoids running many tests when perhaps the first test causes all others to fail anyway.

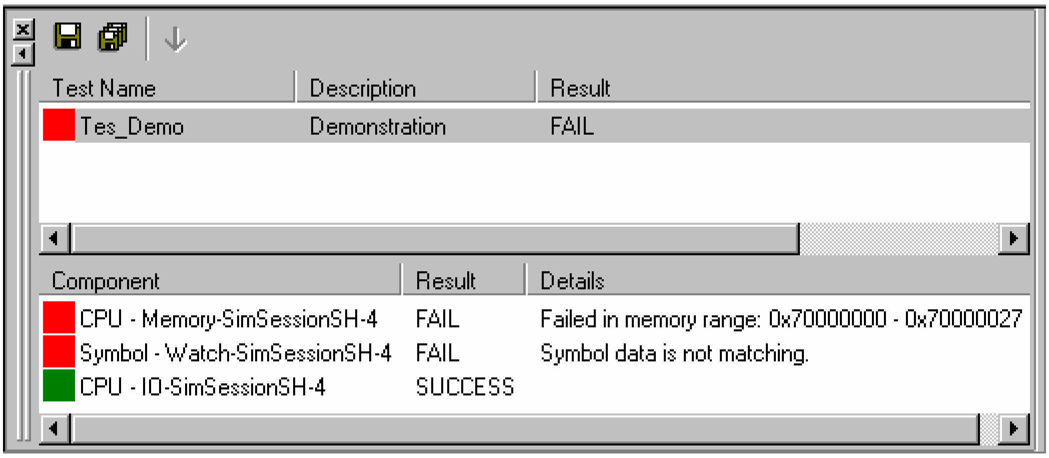
The **Test time out** is for use when you may be executing user code. In some cases if there is a bug the code many execute and never finish during the tests execution. If any one test takes longer than the number of seconds in this box the test is terminated and is flagged as a failure.

You can also run test from the workspace window pop-up menu.

16.9 Using the test browser

The test browser displays the results of the last test execution.

The test browser can be easily opened by selecting the [**Test->Test Results Browser**] menu option.



	Test Name	Description	Result
🟥	Tes_Demo	Demonstration	FAIL

	Component	Result	Details
🟥	CPU - Memory-SimSessionSH-4	FAIL	Failed in memory range: 0x70000000 - 0x70000027
🟥	Symbol - Watch-SimSessionSH-4	FAIL	Symbol data is not matching.
🟩	CPU - IO-SimSessionSH-4	SUCCESS	

The top pane of the test browser lists all of the tests that were executed during the last test run. A green icon to the left of the test name indicates the test was successful. A red icon indicates the test failed.

Selecting a test in the top pane shows the test details in the pane below. The bottom pane of the test browser lists all of the components that were checked. A green icon to the left of the component name indicates the test was successful. A red icon indicates the test failed. If a test fails the details are shown in this window.

Double clicking on the component displays more information on why the test data comparison failed. This can also be viewed by selecting the test and then clicking the details button on the toolbar.

The results of the tests can be exported to file. The two options are **Export** and **Export All**, **Export** only saves the results of the currently selected test. **Export All** saves the results of all the tests executed.

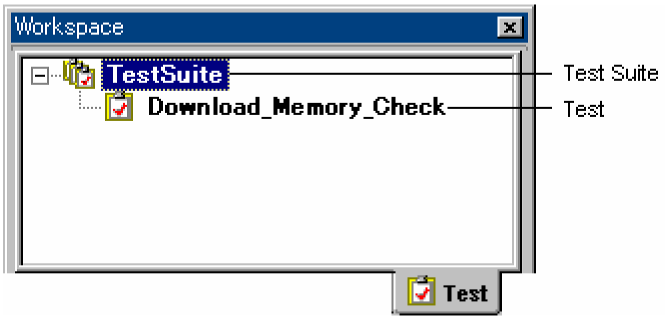
The results can be saved as text or as a comma delimited file for importing into other tools for analysis.

You can also display **Test Browser** from the workspace window pop-up menu.

16.10 Configuring the Test tab of the Workspace window

The **Test** tab of the workspace tab is located alongside the other tabs for **Workspace**, **Navigation** and **Templates**. This tab was created to allow fast access to the tests in your test suite.

When the test suite is opened the test suite is added to the Test tab of the workspace window and the tests in the test suite are added underneath them. This is shown below.



Right-clicking on either the test suite name or the test gives fast access to the test facilities features.

Right-clicking on the test suite icon () allows you the following options:

Pop-up Menu Options	Function
Edit Test Suite	Edits the current test suite. Allows you to add and remove tests to the HEW test system.
Close Test Suite	Closes the current test suite.
Compare Test Image File	Compare test image file allows you to compare a test image file with the current HEW system or with another test image file already on your disk. The results are then displayed in the test browser.
Run Tests	Run tests allows you to run multiple tests that you have defined in the test suite and see the results of the comparisons in the test browser. There are various options to configure the test run execution. You can also use this option to refresh the test image files automatically if you make a minor modification and they need to be updated.
Test Results Browser	This menu displays the test results for one or more test executions. It shows the pass and fail results and the detailed reason why the test failed.
Properties	The Test Suite Properties dialog box is displayed. It displays the test suite name, test suite description, test suite location and the last modified date.

Right-clicking on a test icon () allows you the following options:

Pop-up Menu Options	Function
Run Test Case	Executes the selected test case.
Edit Test Case	Modifies the selected test case.
Remove Test Case	Deletes the selected test case.
Create New Test Image File	Creates a test-image file for the selected test case.
Compare Test Image File	Compares the test-image file with data in the current system.
Edit Test Image File	Modifies the test-image file.
Save Test Image File	Saves the current setting in the HEW system into the test image file.
Save Test Image File As	Saves the current setting in the HEW system as another test image file.
Properties	The Test properties dialog box is displayed. It displays the test name, test description, test location and the last modified date.

16.11 Configuring the Test tab of the Output window

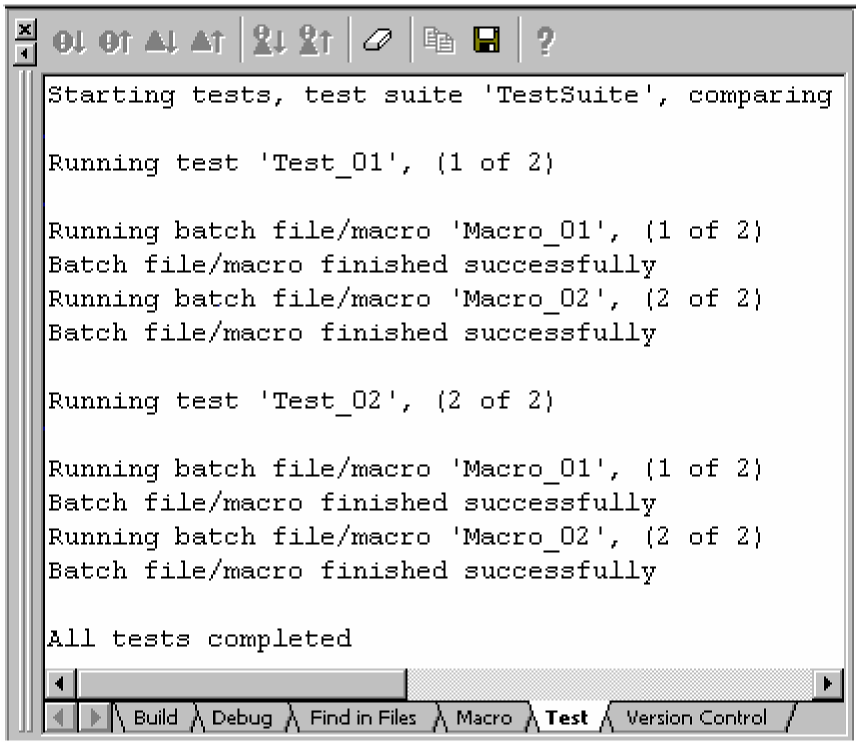
The Test tab of the output window displays the results and progress of the current test execution. The test execution progress shows the current test being executed and the number of tests remaining. If errors occur then these are displayed in this window.

Starting tests, test suite <Test Suite Name>, <Compare Conditions>, <Save Conditions>, continue on failure, timeout=10

Running test <Test Name>, (X of Y)

Running batch file/macro <Macro Name>, (X of Y)

Classification	Item	Run Tests dialog box		Example (Figure below)
Starting tests, test suite	<Test Suite Name>	-	-	'TestSuite'
	<Compare Conditions>	Action after test execution drop-down list box	Compare system against saved test image file	comparing results
			Refresh test image file	refreshing results
	<Save Conditions>	Automatically save test results	☑ (Default)	saving results
			□	not saving results
	<Error Condition>	Stop at first failure	☑	stop on failure
			□ (Default)	continue on failure
	<Time out>	Test timeout	Optional (Default = 10)	timeout=10
Running test	<Test Name>	-	-	'Test_01' 'Test_02'
	(X of Y) X is the current test number and Y is the total number of tests.	-	-	(1 of 2) (2 of 2)
Running batch file/macro	<Macro Name>	-	-	'Macro_01' 'Macro_02'
	(X of Y) X is the current test number and Y is the total number of tests.	-	-	(1 of 2) (2 of 2)



```
Starting tests, test suite 'TestSuite', comparing

Running test 'Test_01', (1 of 2)

Running batch file/macro 'Macro_01', (1 of 2)
Batch file/macro finished successfully
Running batch file/macro 'Macro_02', (2 of 2)
Batch file/macro finished successfully

Running test 'Test_02', (2 of 2)

Running batch file/macro 'Macro_01', (1 of 2)
Batch file/macro finished successfully
Running batch file/macro 'Macro_02', (2 of 2)
Batch file/macro finished successfully

All tests completed
```

Right-clicking displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up Menu Options	Toolbar button	Function
Clear Window	⌫	Clears the contents of the window.
Save	💾	Saves the contents of the window into a text file.
Copy	📋	Copies the selected contents onto the Windows® clipboard.
Toolbar display	-	Shows or hides the toolbar.
Customize toolbar	-	Customizes toolbar buttons.

17. Using the Debugger

This chapter describes the debugging operations and their related windows and dialog boxes.

See Chapter 1, Overview, for the fundamental concepts of the High-performance Embedded Workshop (HEW).

For details on the functions available with the debugging platform in use, refer to the user's manual or help of the emulator or simulator.

17.1 Preparations for debugging

This section describes the preparations for debugging your program. You will learn how to select and configure a debugging platform with which to debug, how to load the user program, and what the debugger sessions are.

17.1.1 Compiling for debug

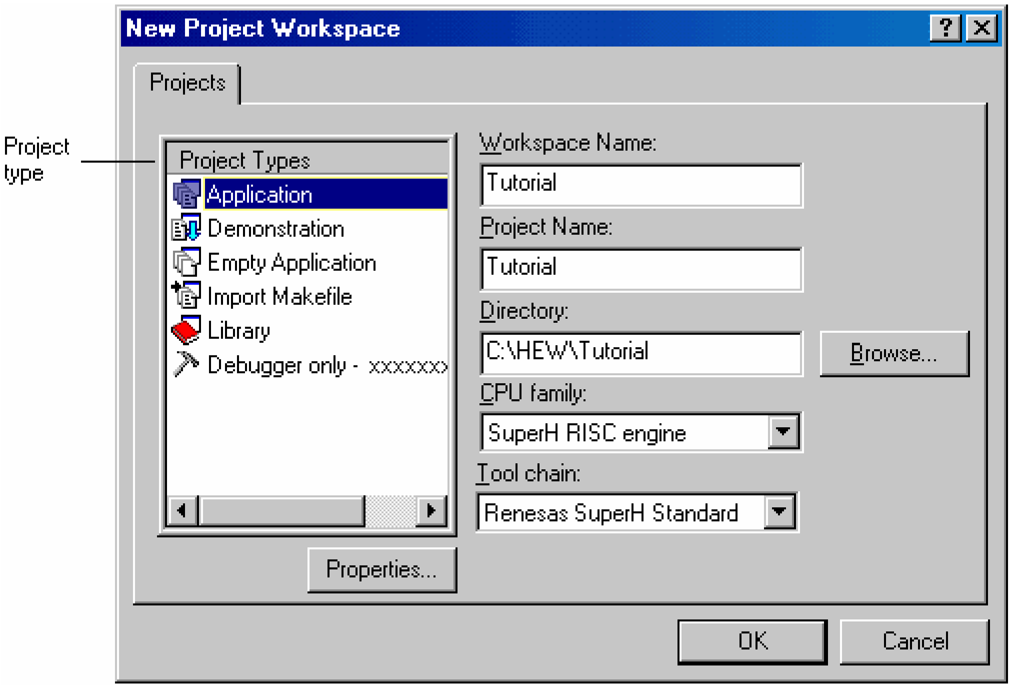
In order to be able to debug your program at C/C++ source level, your C/C++ program must be compiled and linked with the [Debug] option enabled. When this option is enabled, the compiler puts all the information necessary for debugging your C/C++ code into the absolute file or management information file, which are usually called **Debug Object Files**. When you create your project the initial setup will normally be configured for debug.

Notes:

- Make sure you have the debug option enabled on your compiler and linker when you generate an object file for debugging.
- If your debug object file does not contain any debugging information (for example, the S-Record format), then you can still load it into the debugging platform, but you will only be able to debug at the assembly language level.

17.1.2 Selecting a debugging platform

Selecting the debugging platform is very dependent on the installation of the HEW. If the HEW has a toolchain installed then the application project generator will be able to set up both the toolchain and the debugger targets simultaneously. This allows the options for targets and toolchain to be matched closely so that no inconsistencies occur. If there is no toolchain installed you will only be able to select debug-only project types. By default, HEW will display a debug-only project generation type for each CPU family in the **New Project Workspace** dialog box.



Project Types
- Application
- Demonstration
- Empty Application
- Import Makefile
- Library
- Debugger only - xxxxxx

Workspace Name: Tutorial

Project Name: Tutorial

Directory: C:\HEW\Tutorial

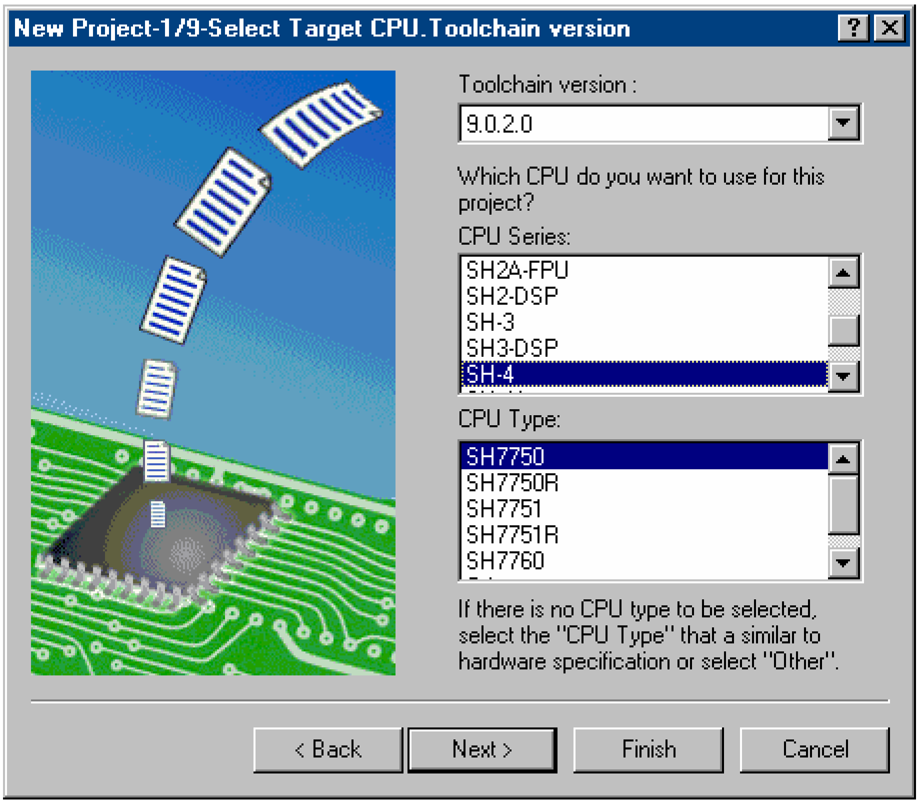
CPU family: SuperH RISC engine

Tool chain: Renesas SuperH Standard

The **New Project Workspace** dialog box allows you to select a project type for generation, which matches your CPU target.

Project Type	Description
Application	Project for generating an execution program that includes the initial routine file written in the C/C++ language.
Assembly Application	Project for generating an execution program that includes the initial routine file written in the assembly language.
Demonstration	Project for generating a demonstration program written in the C language. (SuperH RISC engine Standard Tool chain or H8S,H8/300 Standard Tool chain)
C source startup Application	Project for generating a startup program written in the C language. (M16C Standard Tool chain or M32C Standard Tool chain)
Empty Application	Project for only setting the toolchain environment (no generation file).
Import Makefile	A project to create an executable program by importing an existing makefile.
Library	Project for generating a library file (no generation file).
Debugger only - xxxxxx	A debug-only project (no generation file).

In the example below, the SH-4 simulator/debugger is assumed. For details on the functions available with the debugging platform in use, refer to the user’s manual or help of the debugging platform.



New Project-1/9-Select Target CPU.Toolchain version

Toolchain version : 9.0.2.0

Which CPU do you want to use for this project?

CPU Series: SH2A-FPU, SH2-DSP, SH-3, SH3-DSP, SH-4 (selected)

CPU Type: SH7750 (selected), SH7750R, SH7751, SH7751R, SH7760

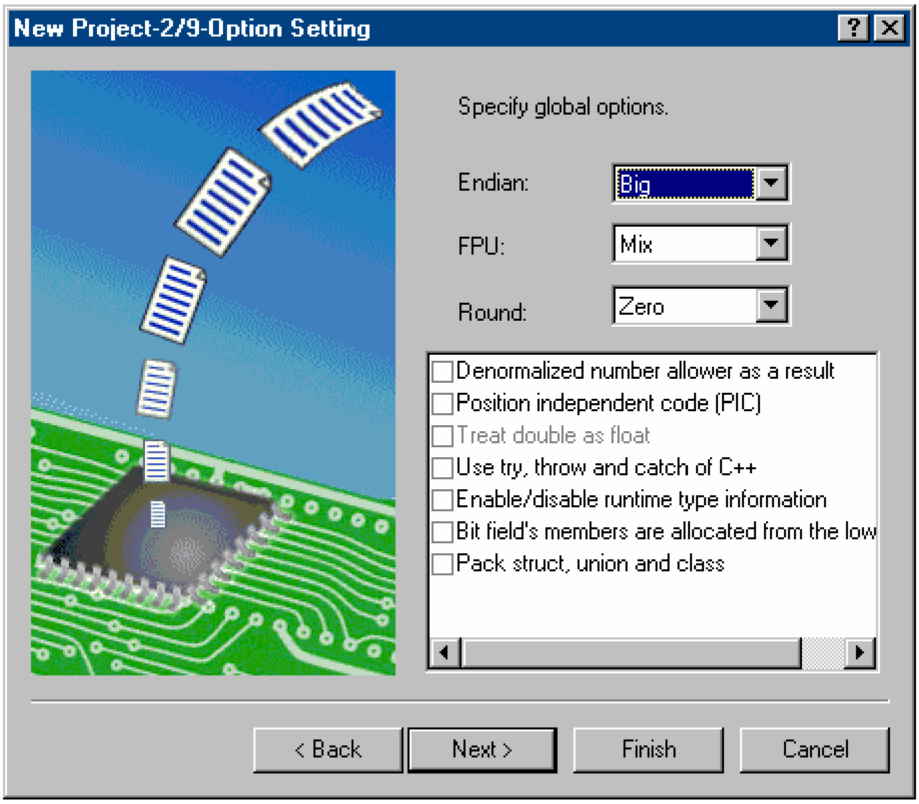
If there is no CPU type to be selected, select the "CPU Type" that a similar to hardware specification or select "Other".

1. Select the CPU and Toolchain version in Step 1. The CPU types (**CPU Type**) are classified according to the CPU series (**CPU Series**). Select the CPU corresponding to the program to be developed because the generation file differs according to the **CPU Series** and **CPU Type** settings. If there is no corresponding CPU, select a CPU with similar hardware specifications or **Other**.

The following buttons at the bottom of the dialog box are the same as those in the **New Project** wizard dialog box.

Next>	Moves to the next display.
<Back	Returns to the previous display.
Finish	Opens the **Summary** dialog box (selections followed by this button are default).
Cancel	Returns to the **New Project Workspace** dialog box.

To move to Step 2, click the **Next>** button in Step 1.



New Project-2/9-Option Setting

Specify global options.

Endian: Big

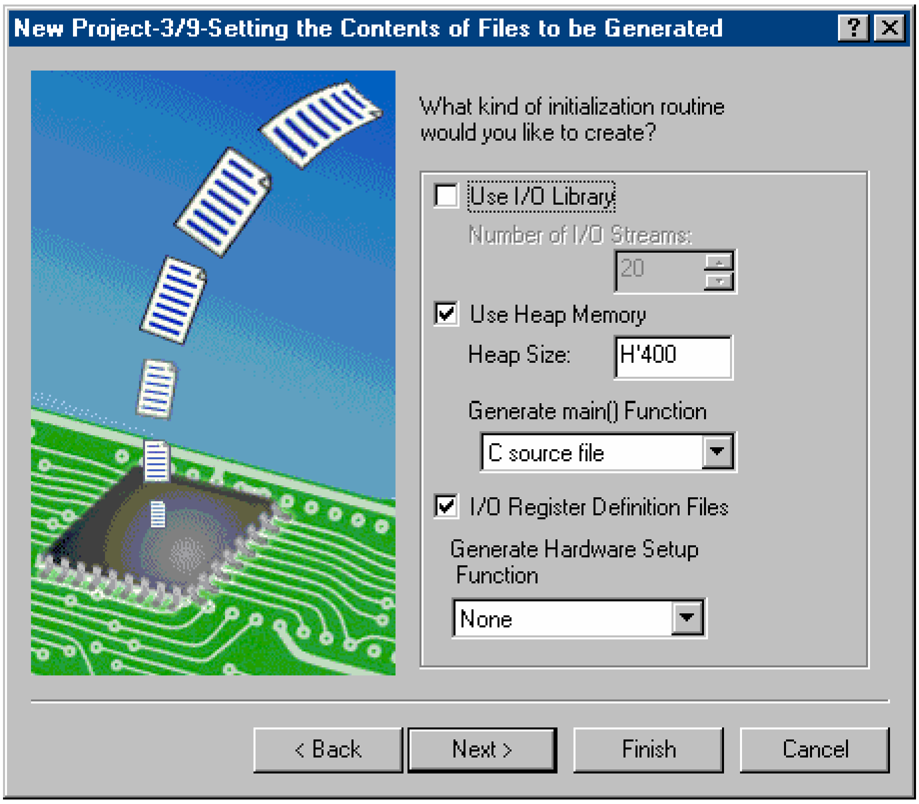
FPU: Mix

Round: Zero

□ Denormalized number allower as a result
□ Position independent code (PIC)
□ Treat double as float
□ Use try, throw and catch of C++
□ Enable/disable runtime type information
□ Bit field's members are allocated from the low
□ Pack struct, union and class

2. Specify the options common to all project files in Step 2. The specifiable items depend on the CPU selected in Step 1.

To move to Step 3, click the **Next>** button in Step 2.



New Project-3/9-Setting the Contents of Files to be Generated

What kind of initialization routine would you like to create?

□ Use I/O Library

Number of I/O Streams: 20

☑ Use Heap Memory

Heap Size: H'400

Generate main() Function: C source file

☑ I/O Register Definition Files

Generate Hardware Setup Function: None

3. Specify the generation file in Step 3.

Use I/O Library — Checking enables use of standard I/O libraries.

- Number of I/O Streams — Specifies the number of I/O streams that can be used simultaneously.

Use Heap Memory — Checking enables use of the heap area management function sbrk( ).

- Heap Size — Specifies the unit of the size of the heap area to be managed.
- Generate main( ) Function — Selects generation of a model main function. Generates a main function file (Project name).c(cpp).

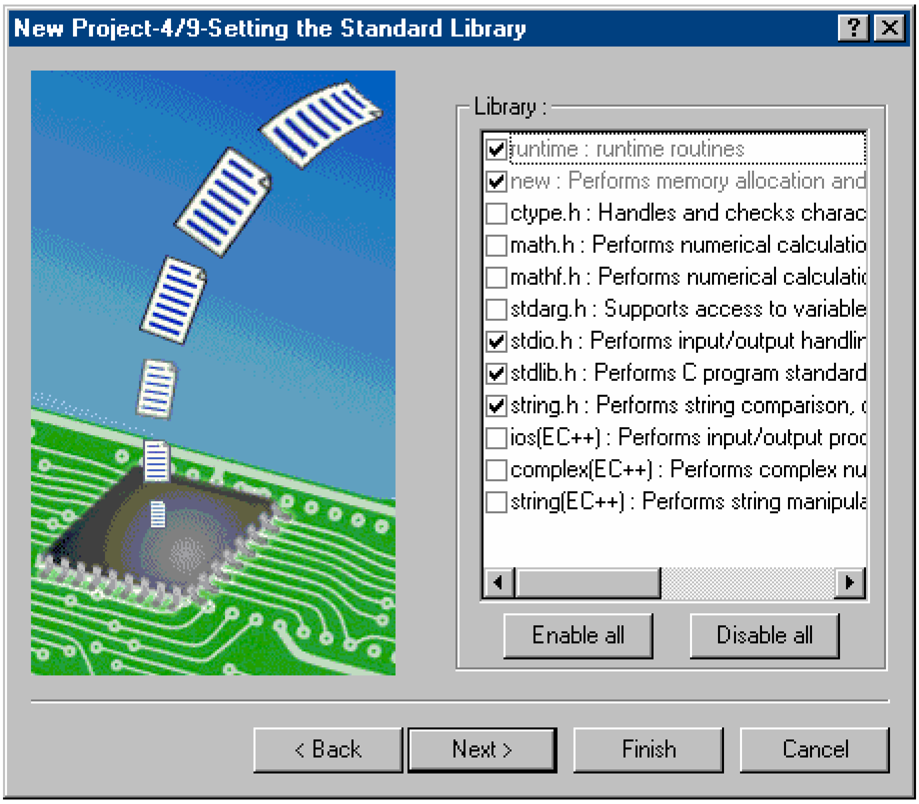
I/O Register Definition Files — Checking generates an I/O register definition file (iodefine.h) written in the C language.

- Generate Hardware Setup Function — Selects generation of a model I/O register initial setting program. Generates a hardware setting file (hwsetup.c(cpp) or hwsetup.src).

Note:

To include a main function that has already been made, select **None** in **Generate main( ) Function** and after making the project, add the file containing the main function to the project. Note that if the name of the function to be included is different, the function calling section in resetprg.c must be modified. Be sure to refer to the hardware manual of the CPU for actual values of the sample file contents, such as the vector table definition and I/O register definition, which are generated by the project generator.

To move to Step 4, click the **Next>** button in Step 3.



New Project-4/9-Setting the Standard Library

Library :

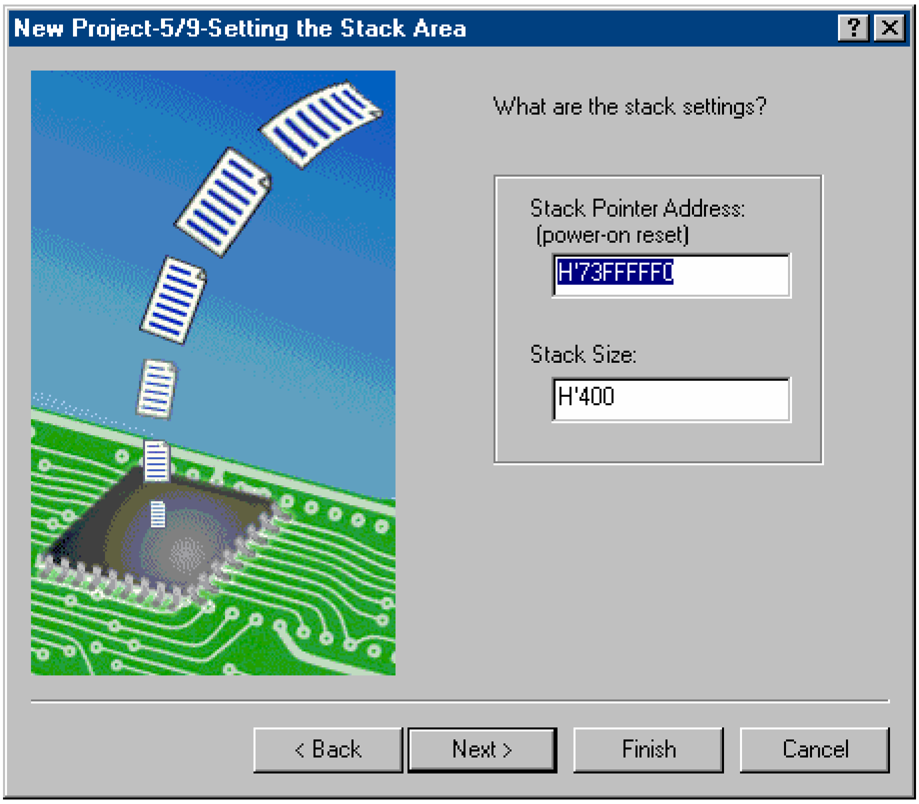
☑ runtime : runtime routines
☑ new : Performs memory allocation and
□ ctype.h : Handles and checks charac
□ math.h : Performs numerical calculatio
□ mathf.h : Performs numerical calculati
□ stdarg.h : Supports access to variable
☑ stdio.h : Performs input/output handlir
☑ stdlib.h : Performs C program standard
☑ string.h : Performs string comparison, c
□ ios(EC++) : Performs input/output proc
□ complex(EC++) : Performs complex nu
□ string(EC++) : Performs string manipula

4. Specify the configuration of the standard libraries used by the C/C++ compiler in Step 4.

The functions defined in the checked items and the runtime functions are included.

Enable all	Selects all standard library functions.
Disable all	Does not select all standard library functions. Note that only the minimum required functions, runtime and new, are selected.

To move to Step 5, click the **Next>** button in Step 4.



New Project-5/9-Setting the Stack Area

What are the stack settings?

Stack Pointer Address:
(power-on reset)
H'73FFFFF0

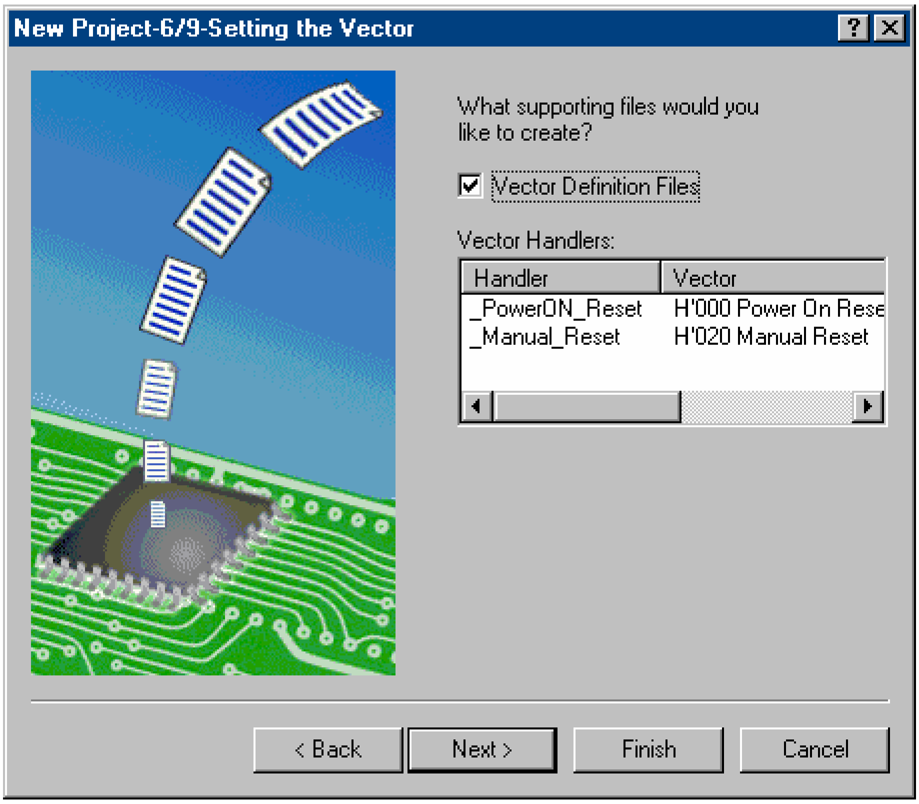
Stack Size:
H'400

5. Specify the stack area in Step 5. This is done by setting the initial value of the stack pointer and the stack size. The initial value of the stack areas depends on the CPU selected in Step 1.

Note:

The stack area is defined by stacksct.h which is generated by the HEW. If stacksct.h has been modified by an editor, it cannot be modified from [**Project -> Edit Project Configuration**] in the HEW.

To move to Step 6, click the **Next>** button in Step 5.



New Project-6/9-Setting the Vector

What supporting files would you like to create?

☑ Vector Definition Files

Vector Handlers:

Handler	Vector
_PowerON_Reset	H'000 Power On Rese
_Manual_Reset	H'020 Manual Reset

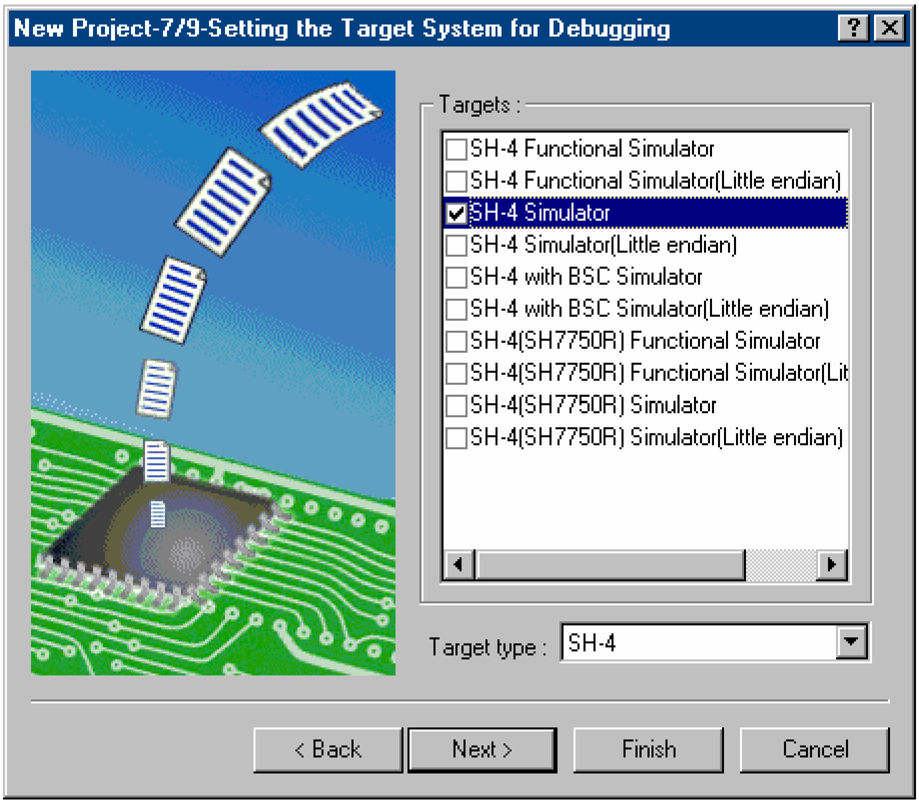
6. Specify the vector in Step 6.

Vector Definition Files	Checking generates a vector definition file and a vector table setting function definition file.	
Vector Handlers	Handler	Displays the handler program name of the reset vector. To modify the handler program, after selecting the handler program name by clicking on it, enter the new handler program name. Note that if the handler program is modified, a reset program (resetprg.c) is not generated.
	Vector	Displays a description of the vector.

Note:

Since the generated reset program, interrupt functions, reset vector handlers, and interrupt source register definitions are samples, be sure to refer to the CPU hardware manual.

To move to Step 7, click the **Next>** button in Step 6.



New Project-7/9-Setting the Target System for Debugging

Targets :

- □ SH-4 Functional Simulator
- □ SH-4 Functional Simulator(Little endian)
- ☑ SH-4 Simulator
- □ SH-4 Simulator(Little endian)
- □ SH-4 with BSC Simulator
- □ SH-4 with BSC Simulator(Little endian)
- □ SH-4(SH7750R) Functional Simulator
- □ SH-4(SH7750R) Functional Simulator(Lit
- □ SH-4(SH7750R) Simulator
- □ SH-4(SH7750R) Simulator(Little endian)

Target type : SH-4

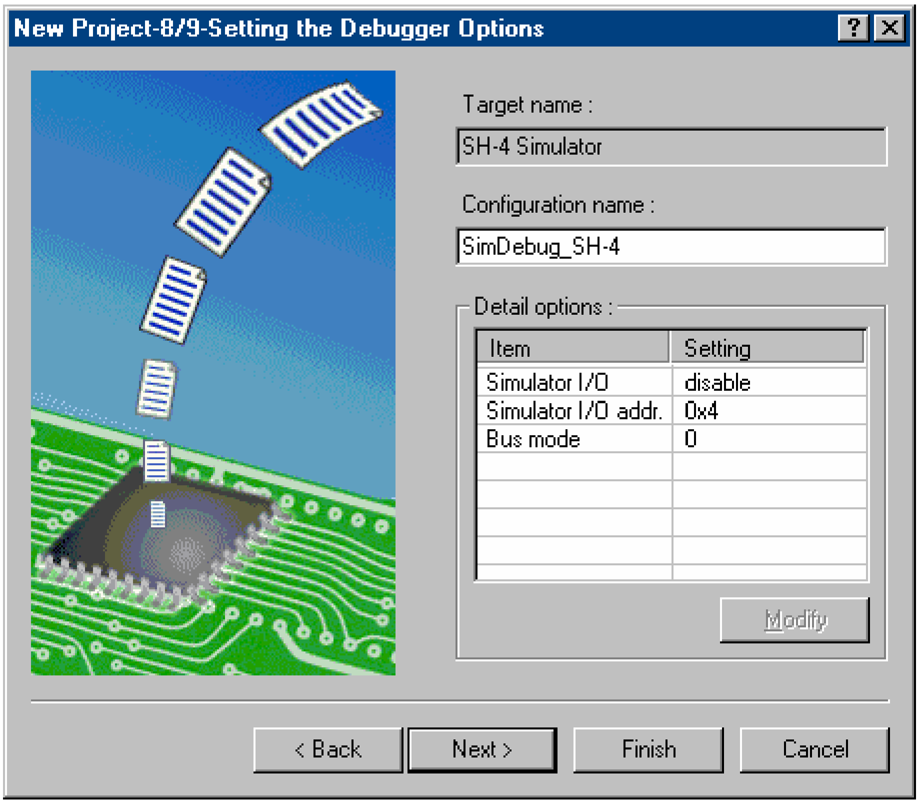
7. Specify the debugger targets in Step 7.

Targets	Sets the debugger targets. Select (by checking) the debugger targets. No selection or a selection of more than one target is possible.
Target Type	Specifies the type of the targets displayed in **Targets**.

Note:

The endian type selected in step 2 will be applied to the compiler settings. This is separate from the endian type of the debugger target selected in step 7.

To move to Step 8, click the **Next>** button in Step 7.



New Project-8/9-Setting the Debugger Options

Target name : SH-4 Simulator

Configuration name : SimDebug_SH-4

Detail options :

Item	Setting
Simulator I/O	disable
Simulator I/O addr.	0x4
Bus mode	0

8. Set the options for the debugger targets selected in Step 8.

Configuration name — By default, the HEW generates two configurations: **Release** and **Debug**. If a debugger target is selected, a configuration for the selected target is also generated (an abbreviation including the target name). This configuration name can be changed in **Configuration name**.

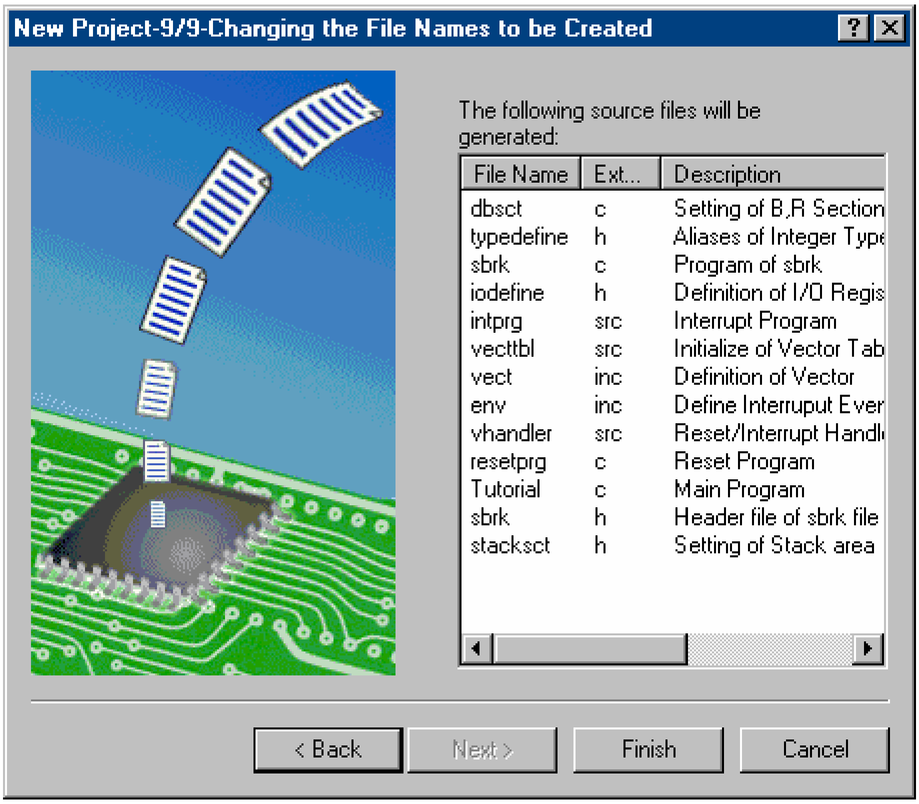
Detail options — Sets the debugger target options. To modify an option, select **Item** and click **Modify**. If the selected item cannot be modified, **Modify** remains gray even when **Item** is selected.

Simulator I/O — System call for standard I/O or file I/O from the user program is enabled (**Enable**) or disabled (**Disable**).

Simulator I/O addr. — Address for above system call.

Bus mode — Currently cannot be used by the simulator/debugger.

To move to Step 9, click the **Next>** button in Step 8.



New Project-9/9-Changing the File Names to be Created

The following source files will be generated:

File Name	Ext...	Description
dbsct	c	Setting of B,R Section
typedefine	h	Aliases of Integer Type
sbrk	c	Program of sbrk
iodefine	h	Definition of I/O Regis
intprg	src	Interrupt Program
vecttbl	src	Initialize of Vector Tab
vect	inc	Definition of Vector
env	inc	Define Interruput Ever
vhandler	src	Reset/Interrupt Handl
resetprg	c	Reset Program
Tutorial	c	Main Program
sbrk	h	Header file of sbrk file
stacksct	h	Setting of Stack area

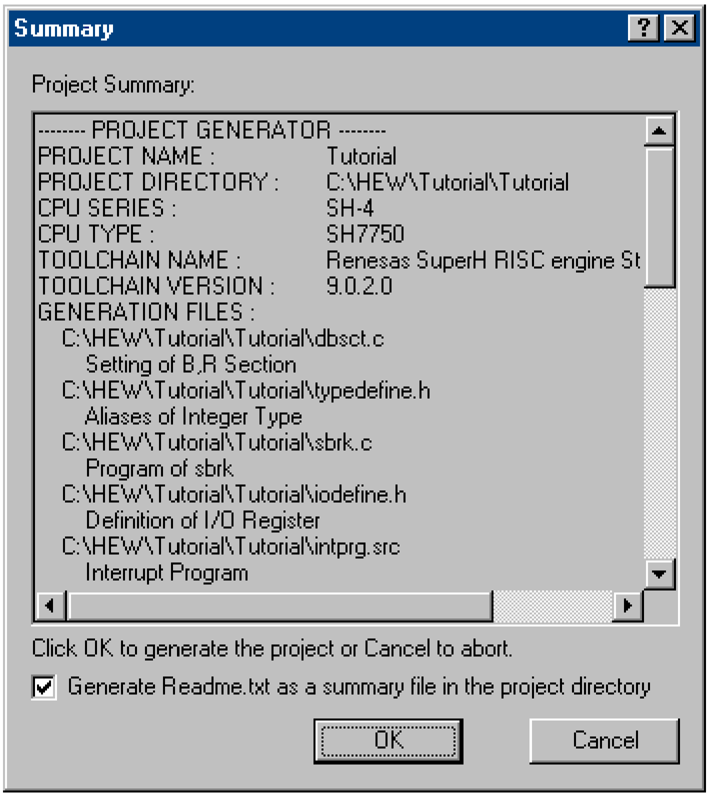
9. The files to be generated by the HEW based on the settings made so far is displayed as a list in Step 9.

File Name — File name
To change a file name, after selecting the file name by clicking on it, enter the new file name.

Extension — File extension

Description — Description of the file

Clicking the **Finish** button in Step 9 displays the **Summary** dialog box.



```
-------- PROJECT GENERATOR --------
PROJECT NAME :          Tutorial
PROJECT DIRECTORY :     C:\HEW\Tutorial\Tutorial
CPU SERIES :            SH-4
CPU TYPE :              SH7750
TOOLCHAIN NAME :        Renesas SuperH RISC engine St
TOOLCHAIN VERSION :     9.0.2.0
GENERATION FILES :
  C:\HEW\Tutorial\Tutorial\dbsct.c
    Setting of B,R Section
  C:\HEW\Tutorial\Tutorial\typedefine.h
    Aliases of Integer Type
  C:\HEW\Tutorial\Tutorial\sbrk.c
    Program of sbrk
  C:\HEW\Tutorial\Tutorial\iodefine.h
    Definition of I/O Register
  C:\HEW\Tutorial\Tutorial\intprg.src
    Interrupt Program
```

Click OK to generate the project or Cancel to abort.

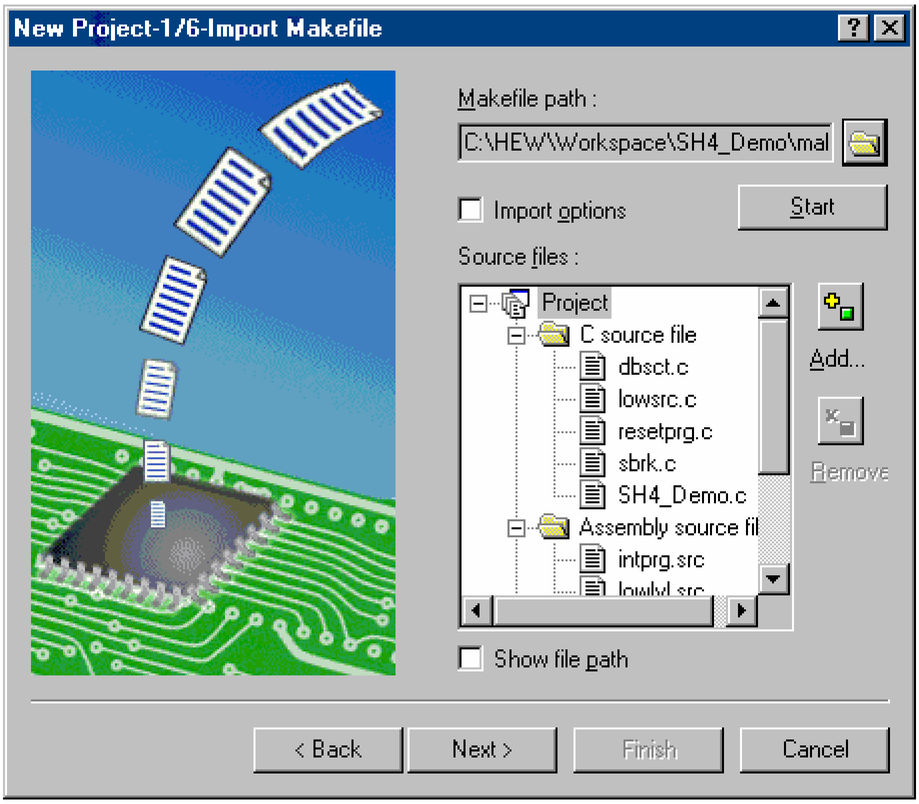
☑ Generate Readme.txt as a summary file in the project directory

10. The project generator displays information on the project to be generated in the **Summary** dialog box. After confirming the display contents, click the OK button. Clicking Cancel returns to the **New Project** wizard dialog box. Checking **Generate Readme.txt as a summary file in the project directory** will save the project information displayed in the **Summary** dialog box as a text file named Readme.txt in the project directory.

(1) To create a new project having information from makefile

HEW can analyze GNU make format and Hmake format (HEW generated) and create a workspace which has file information from makefile.

Open the **New Project Workspace** dialog box and select the **Import Makefile** as project type. After supplying some fields (e.g.: Workspace Name) and pressing OK button, **New Project – Import Makefile** dialog box is appeared.



New Project-1/6-Import Makefile

Makefile path : C:\HEW\Workspace\SH4_Demo\mal

□ Import options

Source files :

```
Project
  C source file
    dbsct.c
    lowsrc.c
    resetprg.c
    sbrk.c
    SH4_Demo.c
  Assembly source fil
    intprg.src
    lowlvl.src
```

□ Show file path

When a makefile is selected by **Makefile path**, **Source files** shows source files in the makefile. To view the source files in the makefile again, click **Start**.

To apply toolchain options such as the compiler, select the **Import options** check box.

If you want to remove a file from the project, you can remove it by selecting the file and pressing **Remove** button. And if you want to add a file to the project, you can add it by pressing **Add** button.

Selecting the **Show file path** check box shows the full path of the file.

17.1.3 Editing project configuration

If you are using the SuperH or H8 toolchains then it is possible to configure the simulator again using the project generator. This feature is not enabled for the demonstration project type.

1. Click on the [**Project->Edit Project Configuration**]. The **Edit Project Configuration** dialog box is displayed.
2. Click on the **Target** tab.
3. Select the target you wish to use and then click OK.

17.1.4 Configuring the debugging platform

Before you can load a program into your debugging platform you must set it up to match your application's system. The items that must be set-up are typically device type, operating mode, clock speed and the memory map. It is particularly important to set-up the memory map, as you must have memory in the debugging platform, into which your user code will be loaded.

In the High-performance Embedded Workshop, the project generation process will have completed much of this work. However if you are using a different configuration of board from the standard types then some customization will be essential.

(1) Setup

To set-up the debugging platform configuration choose the [**Setup->Simulator**] menu option or the [**Setup->Emulator**] menu option. Under this sub menu will be the menus which can be used to configure your debug platform.

In the case of the SuperH family Simulator the available menus are the **System** and **Memory resource**. These options both allow the simulator to be customized and setup to your requirements.

You will be presented with a set-up dialog specific to the debugging platform that you chose in the **Debug Settings** dialog.

For a detailed description of the features available in your debugging platform, please refer to the separate Debugging Platform User Manual.

(2) Memory mapping

For the debugger to correctly represent your user system, the memory map must be set up. It needs to know which areas in the device's address space are RAM, ROM, on-chip registers or areas where there is no memory.

When you select the device type and mode in the project generator, the HEW will automatically set up the map for that device and the mode in which the processor is operating. For example, in a device with internal ROM and RAM, the areas where these are located in the device's memory map will be set by default.

If you are using a device that does not have internal memory, or a device with external memory instead of (or in addition to) the internal memory, then you must tell the debugging platform that you have memory there.

Tip:

If you are trying to debug code with an emulator and need some memory available that does not exist either on-chip or externally (in your hardware), then you can map some *emulation memory* from the emulator to the address space for your application to use.

The details will be specific to the debugging platform that you chose in the new project.

Additional information about the memory mapping can be viewed in the **System Status** view's **Memory** pane. The **Device Configuration** area shows how the memory in the device's address space.

Note:

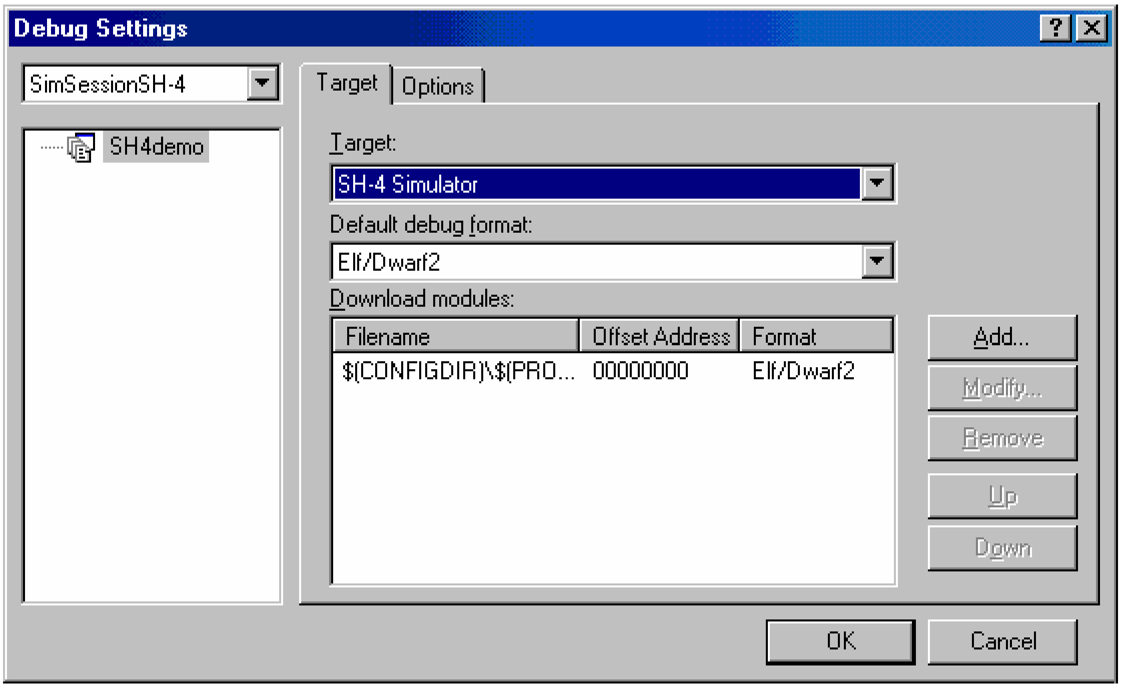
Due to page length limitations in some emulators, the range addresses may not exactly match the entered addresses.

(3) Setting up the HEW debugger

Normal operation of the HEW debugger means that your target and download modules will be automatically configured in the project generation process. However in some cases it may be necessary for you to manually configure your debug session. This is often the case when using old toolchains and project generators that do not support the latest HEW interfaces.

To check your debug session setup display the Debug Settings dialog box. Select one of the following operations

- Select the [**Debug->Debug Settings**] menu option, **OR**
- Right-click on the download module or within the **Download modules** folder on the Projects tab of the workspace window and select **Debug Settings** from the pop-up menu.



Debug Settings

SimSessionSH-4

SH4demo

Target:
SH-4 Simulator

Default debug format:
Elf/Dwarf2

Download modules:

Filename	Offset Address	Format
$(CONFIGDIR)\$(PRO...	00000000	Elf/Dwarf2

Target, Default debug format, or Download modules can be selected on the **Target** tab of the Debug Settings dialog box. Moreover, the **Options** tab provides options of accesses to command-line batch files, accesses with connection to the target, and accesses with download.

To change the target the following operations is necessary

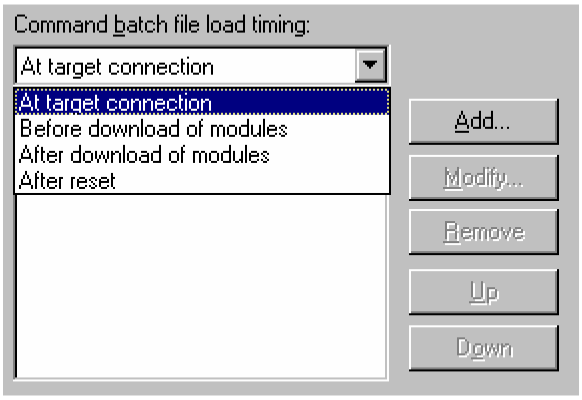
1. Select the project that needs to be changed in the tree on the left of the dialog. It defaults to the current project.
2. Select the session which is to be modified in the drop list above the tree.
3. Change the target using the target drop list control. This removes any target specific setup options that have been previously been set.

(4) Setting the Options of Debug Settings

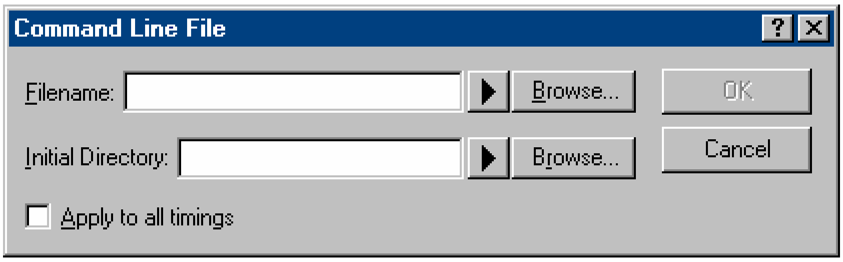
The HEW debugger is tightly integrated with the TCL command line facilities. This means that it is possible to write batch files for the HEW debugger which can be executed automatically at certain times. The **Command batch file load timing** list displays the order in which the files will be executed. It is possible to Add, Modify, Remove, Up and Down the files in this list.

To configure the automatic command line batch file execution

1. Select the [**Debug->Debug Settings**] menu item. The Debug Settings dialog box will be invoked.
2. Select the **Options** tab.
3. Select the **Command batch file load timing**. This can be the following values, at target connection, before downloading the modules, after downloading the modules, and after reset.


Command batch file load timing:
At target connection
At target connection
Before download of modules
After download of modules
After reset
Add...
Modify...
Remove
Up
Down

4. Then click **Add**. The debugger will then display the add **Command Line File** dialog box.



5. Enter the command-batch file name in the **Filename** field. If you wish to insert a placeholder into the **Filename** field, click the placeholder button (▶) and select the placeholder from the pop-up menu. To browse a file, click the Browse button.
6. In the **Initial Directory**] field, enter the name of the directory where you wish to execute the command batch file. By this setting, you can also use command batch files that have relative paths. If you wish to insert a placeholder into the **Initial Directory** field, click the placeholder button (▶) and select the placeholder from the pop-up menu. To browse a directory, click the Browse button.
7. Selecting the **Apply to all timings** check box adds the batch file to every timing. (At target connection, Before downloading the modules, After downloading the modules, and After reset)
8. Click OK to add the batch file.
9. Once added it can moved into the correct place in the order by using the **Up** and **Down** buttons. This is only valid if you are adding multiple command line batch files.

Other options

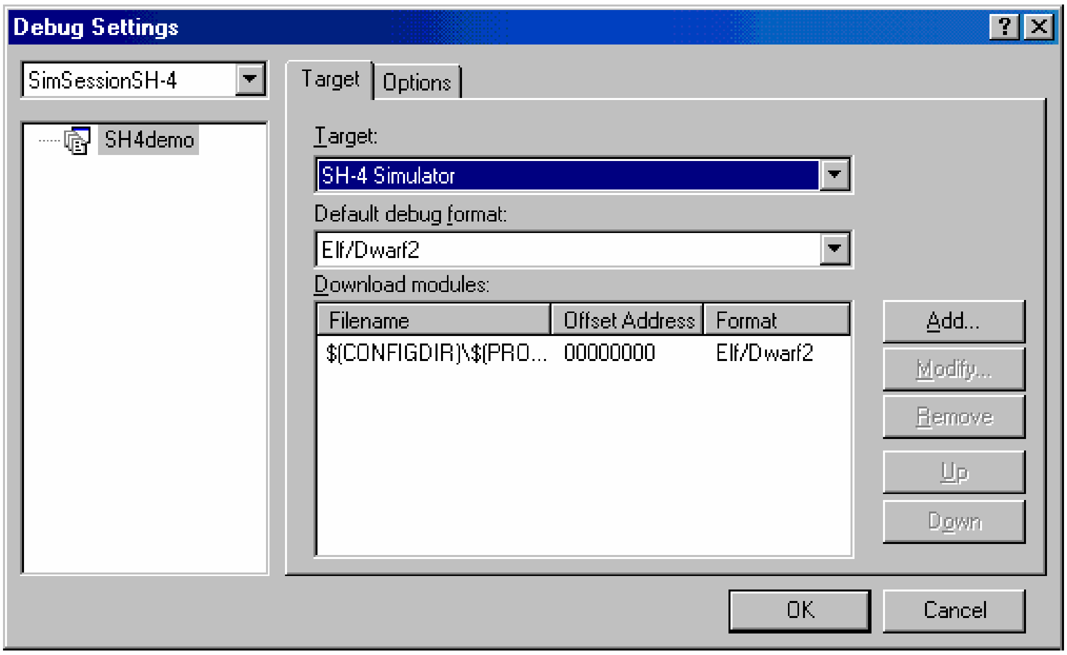
Option	Description
Do not perform automatic target connection ***1**	If this checkbox is selected, the target is not connected until you select [**Debug->Connect**]. By default, this checkbox is not selected.
Download modules after build	If this checkbox is selected, the user program will be automatically downloaded after a build. By default, this checkbox is selected.
Reset CPU after download module ***1**	If this checkbox is selected, the target will be automatically reset after downloading the user program. This checkbox is only enabled if the feature is supported by the selected target. Selecting a new target will reset this option to the default setting for the target. Default for this checkbox depends on the debugging platform.
Remove breakpoints on download	If this checkbox is selected, the breakpoints will be automatically removed after downloading the user program. Default for this checkbox depends on the debugging platform.
Disable memory access until after target connection command file execution	When this checkbox is selected, if a command batch file to be automatically executed at connection to the target has been specified, the memory in the target will not be accessed until execution of the command batch file is finished. This option is useful when initial setting of memory is necessary at connection to the target. By default, this checkbox is not selected.
Limit disassembly memory access	Disables reading data from memory outside the displayed range in disassembly mode. Default for this checkbox depends on the debugging platform.

***1.** Support for this function depends on the debugging platform.

17.1.5 Downloading modules

(1) Setting the downloading a program

Once you have made sure that there is memory in your system in which to download your code, you can then proceed to download a program to debug. The initial selection of download module is automatic with regard to an application generator, as it is the output from the linker. However, with regard to the debug-only project generator, it is possible for you to browse to the module that you wish to download. This is outlined in the section 17.1.2, Selecting a debugging platform.



Target: SH-4 Simulator

Default debug format: Elf/Dwarf2

Download modules:

Filename	Offset Address	Format
$(CONFIGDIR)\$(PRO...	00000000	Elf/Dwarf2

It is also possible to manually choose download modules after the project creation. This is achieved via the **Debug Settings** dialog box. This dialog box allows you to control the debug settings throughout your workspace. The tree on the left of the dialog contains all of the current projects. Selecting a project in this tree will then show you the settings for that project and the configuration selection in the Configuration drop-down list. In this list box, it is possible to select multiple configurations or all configurations. If you select multiple configurations you can choose to modify the settings for one or more configurations at once. The **Debug Settings** dialog box displays the following debug options:

- Current debug target for the current project and configuration selection.
- Download modules for the current project and configuration selection.

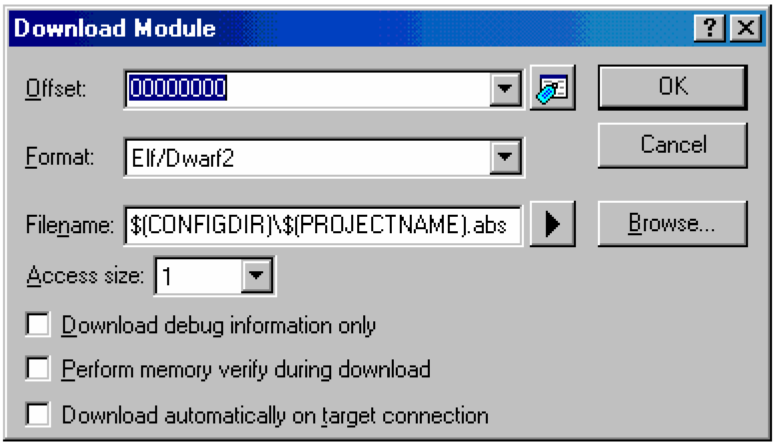
The download module list displays the order in which the files will be downloaded to the target. It is possible to Add, Modify, Remove, Up and Down modules in this list.

To add a new download module

1. Select one of the following operations:

- Select the [**Debug->Debug Settings**] menu option, **OR**
- Right-click on a module or within the Download modules folder on the Projects tab of the workspace window to display a pop-up menu. Select **Debug Settings**.

2. The **Debug Settings** dialog box will be displayed. In the project tree, select the project and configurations to which you want to add a download module.
3. Click the **Add** button. The **Download Module** dialog box is displayed.



Download Module

Offset: 00000000

Format: Elf/Dwarf2

Filename: $(CONFIGDIR)\$(PROJECTNAME).abs

Access size: 1

☐ Download debug information only

☐ Perform memory verify during download

☐ Download automatically on target connection

4. All fields must be setup for the download module to be configured correctly. The first field is **Offset**. This is the memory address offset the module will be loaded at. It defaults to 0. After the module has been added, the offset value is shown on the right of the module name in the [**Debug->Build**] submenu, the **Download modules** folder on the Projects tab of the workspace window, and the [**File->Recently Download Module**] submenu.
5. The **Format** drop-down list box contains a list of supported object format. Note this does not have to match the default object format. However you can only debug modules that match the format specified in the debug settings dialog default object format field.
6. The **Filenames** can be specified with placeholders or as an absolute setting. It is recommended to use placeholders.
7. The **Access size** field specifies the access width when the memory is accessed.
8. The **Download debug information only** checkbox can be used when you wish to download the debugging information.
9. The **Perform memory verify during download** checkbox can be used to do additional checks when downloading the module to ensure it was correctly downloaded to the target device.
10. The **Download automatically on target connection** checkbox can be used to automatically download the module when the target is being connected.
11. When you click the OK button, the debug download module is added to the bottom of the list.

For automatically downloading an existing module to the target when adding it to the Download module list, see section 17.1.5 (2), Downloading modules.

To change the setting of a download module

1. Select one of the following operations:

- Open the **Debug Settings** dialog box and select a module, for which you wish to modify the setting, in the **Download modules** list and click the **Modify** button, **OR**
- Right-click on the download module or within the Download modules folder on the Projects tab of the workspace window and select **Properties** from the pop-up menu.

2. **Download Module** dialog box will be displayed.
3. Change the setting and click the OK button.

To remove download modules

1. Open the **Debug Settings** dialog box.
2. Select a module, which you wish to remove, in the **Download modules** list and click the **Remove** button.

To remove selected download modules using the Projects tab of the Workspace window

1. Select the download modules that you want to remove in the Projects tab of the Workspace window. Multiple download modules can be selected by holding down the SHIFT or CTRL key.
2. Select one of the following operations:

- Select **Remove** from the pop-up menu opened by right-clicking, **OR**
- Press **Delete** key.

3. A confirmation dialog box opens for you to select whether or not to remove the selected download modules from the project. To delete the selected download modules, select **Yes**. Otherwise select **No** or **Cancel**.

If you do not wish to open this confirmation dialog box, select the **Don't ask this question again** checkbox. To open this dialog box again, select [**Setup->Options**] to open the Options dialog box. Select the **Remove download module(s) from project** checkbox on the Confirmation tab. By default, this checkbox is selected.

To change the order of a module to be downloaded to the target

1. Open the **Debug Settings** dialog box.
2. Select a module in the **Download modules** list and click the **Up** or **Down** button.

Any changes made in the **Debug Settings** dialog box are only changed when you click OK.

The default debug format is set to the first download module in the list by default. Only one default debug object format can be specified for each session. All currently installed debugger formats are listed here.

(2) Downloading modules

Download the object program to be debugged.

To download modules

Select one of the following operations:

- Select the module, which you wish to download, from the [**Debug->Download Modules**] submenu, **OR**
- Double-click the module in the Download modules folder on the Projects tab of the workspace window, **OR**
- Right-click on the module in the Download modules folder on the Projects tab of the workspace window to display a pop-up menu. Select **Download** or **Download (Debug Data Only)**.

The HEW shows the most recently downloaded four modules on the submenu of [**File->Recently Download Modules**]. This is useful when you wish to download a module that you recently used.

To download all modules

Select one of the following operations:

- Select [**Debug->Download Modules->All Download Modules**], **OR**
- Right-click within the Download folder on the Projects tab of the workspace window to display a pop-up menu. Select **Download all modules**.

Select the [**Debug->Debug Settings**] menu item. The Debug Settings dialog box will be invoked. The files will be downloaded to the target in order of the **Download modules** list.

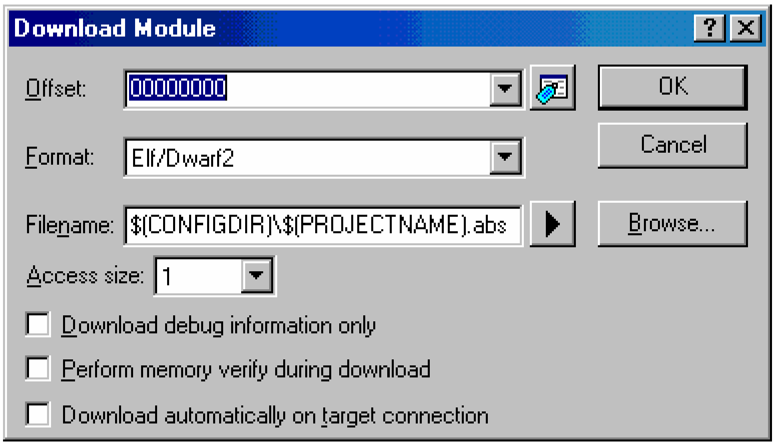
An existing module, which is not yet added to the Download modules list, can be automatically downloaded to the target when it is added to the Download modules list.

To automatically download an existing module to the target when adding it to the Download modules list

1. Select one of the following operations:

- Select [**File->Download A New Module**], **OR**
- Right-click on a module or within the Download modules folder on the Projects tab of the workspace window to display a pop-up menu. Select **Download A New Module**.

2. **Download Module** dialog box will be displayed.



Download Module

Offset: 00000000

Format: Elf/Dwarf2

Filename: $(CONFIGDIR)\$(PROJECTNAME).abs

Access size: 1

□ Download debug information only

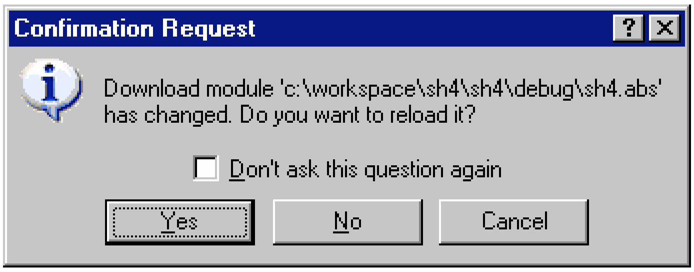
□ Perform memory verify during download

□ Download automatically on target connection

3. All fields must be setup for the download module to be configured correctly. The first field is **Offset**. This is the memory address offset the module will be loaded at. It defaults to 0. After the module has been added, the offset value is shown on the right of the module name in the [**Debug->Build**] submenu, the **Download modules** folder on the Projects tab of the workspace window, and the [**File->Recently Downloaded Modules**] submenu.
4. The **Format** drop-down list box contains a list of supported object format. Note this does not have to match the default object format. However you can only debug modules that match the format specified in the debug settings dialog default object format field.
5. The **Filenames** can be specified with placeholders or as an absolute setting. It is recommended to use placeholders.
6. The **Access size** field specifies the access width when the memory is accessed.
7. The **Download debug information only** checkbox can be used when you wish to download the debugging information.
8. The **Perform memory verify during download** checkbox can be used to do additional checks when downloading the module to ensure it was correctly downloaded to the target device.
9. The **Download automatically on target connection** checkbox can be used to automatically download the module when the target is being connected.
10. Click the **OK** button.

Select the [**Debug->Debug Settings**] menu item. The **Debug Settings** dialog box will be invoked. The module is added to the bottom of the **Download modules** list on the **Target** tab.

If a module previously downloaded is modified outside the HEW, a confirmation dialog box appears asking if you wish to download this module again. To download the module again, select **Yes**. Otherwise select **No** or **Cancel**.



Confirmation Request

Download module 'c:\workspace\sh4\sh4\debug\sh4.abs' has changed. Do you want to reload it?

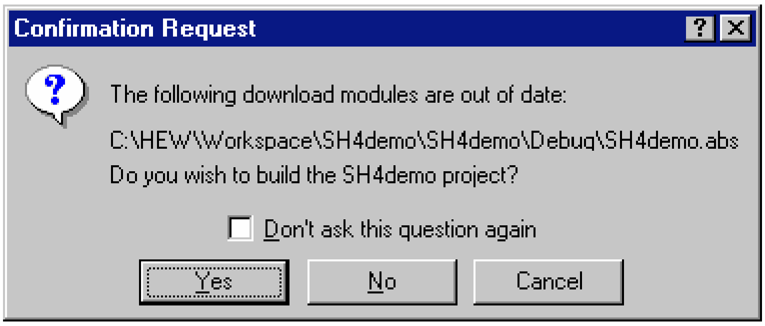
If you do not wish to open this confirmation dialog box, select the **Don't ask this question again** checkbox.

To open the confirmation dialog box again

1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Confirmation** tab.
3. Select the **Reload out-of-date download modules** checkbox. This checkbox is selected by default.
4. Click **OK**.

(3) Check for changed source files before download

The HEW can check to see if any of the source files have changed before the download module is downloaded in the current project. If files have been modified then a confirmation is launched which asks the user if they wish to rebuild the code before the download takes place. To rebuild the code before the download again, select **Yes**. To not rebuild the code before the download again, select **No**.



Confirmation Request

The following download modules are out of date:

C:\HEW\Workspace\SH4demo\SH4demo\Debug\SH4demo.abs

Do you wish to build the SH4demo project?

If you do not wish to open this confirmation dialog box, select the **Don't ask this question again** checkbox.

To open the confirmation dialog box again

1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Confirmation** tab.
3. Select the **Build out-of-date download modules** checkbox. This checkbox is selected by default.
4. Click OK.

(4) Showing a source tree on download (debug-only project)

While a debug-only project (i.e., "Debugger only - xxxxxx") is in use, a tree of source files for a download module are automatically shown in the **Projects** tab of the workspace window once the module has been downloaded.

The files for a download module are retrieved when the module is downloaded.

If the files are not full paths and are relative or short file names, they will automatically be searched for relative to the download module directory.

Files that cannot be found after the initial search will be shown to you in the **Locate Files** dialog box.

However, you can choose not to open the **Locate Files** dialog box at downloading.

To prevent opening of the Locate Files dialog box at downloading

1. Select the [**Setup->Options**] menu item. The **Options** dialog box will be displayed.
2. Select the **Confirmation** tab.
3. Select the **Do not show Debug only workspace browse dialog on download** checkbox. This checkbox is not selected by default.
4. Click **OK**.

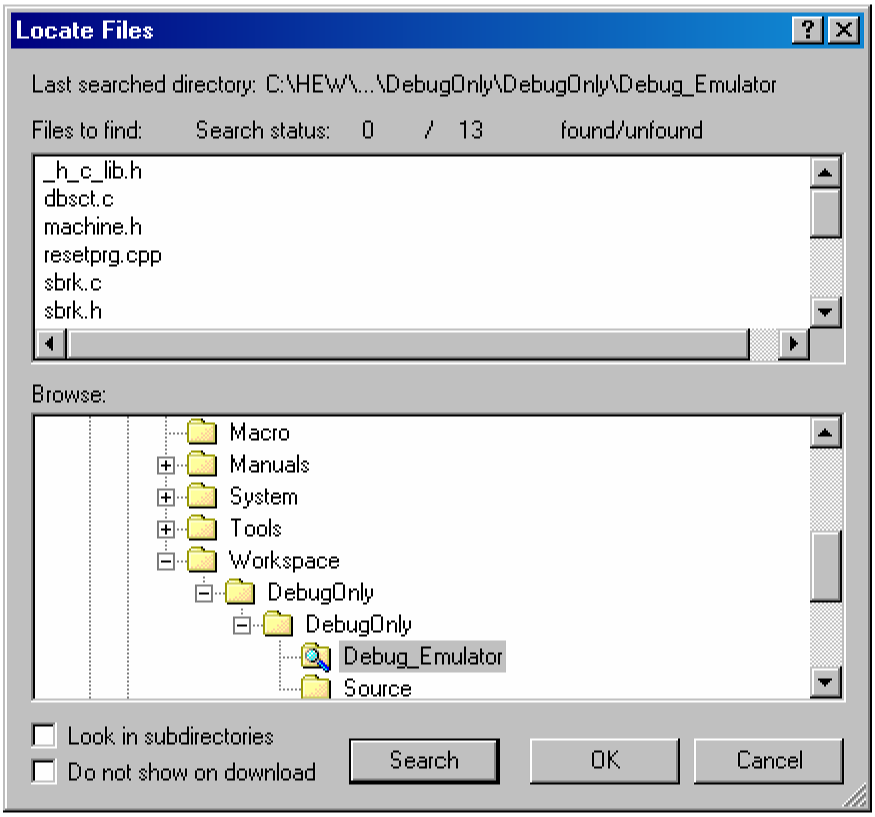
- **Debuggers supporting the feature of showing a source tree**

This feature is only supported by the debuggers that can create debug-only projects. The condition of the support, however, depends on the debugger as shown below.

Debugger	Condition of the Support
SuperH RISC engine or H8SX, H8S, H8 family emulator debugger	The HEW has become V.4.02 or later due to an update.
M16C, M32C, M32R family emulator debugger	The HEW included in the product package is V.4.02 or later.

(a) Locating files

Files that cannot be found or do not exist after the initial search will be shown to the user in the **Locate Files** dialog box (in the **Files to find** list).



Locate Files

Last searched directory: C:\HEW\...\DebugOnly\DebugOnly\Debug_Emulator

Files to find: Search status: 0 / 13 found/unfound

_h_c_lib.h
dbsct.c
machine.h
resetprg.cpp
sbrk.c
sbrk.h

Browse:

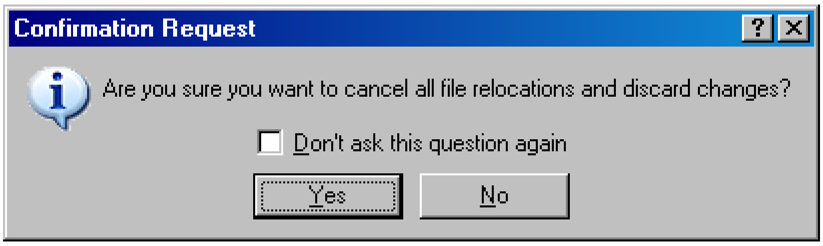
□ Look in subdirectories
□ Do not show on download

To locate files

1. The **Last searched directory** holds the last directory that was searched for these module files.
2. The **Search status** field shows the number of files that were found against the number that have not - this is only relevant to the files in the dialog and do not account for any previous searches.
3. The **Files to find** list holds a list of all the files that could not be found, they will appear in short filename format, and will be searched for in this manner.
4. The **Browse** section allows the user to select a directory in which to search the remaining files for.
5. The **Look in subdirectories** check box will look for the files in all sub-directories for the chosen directory.
6. The **Do not show on download** check box will not bring up this dialog box on download if there are files that can not be located. This checkbox is also present in the **Confirmation** tab of the **Options** dialog box, and can be turned off and on here.
7. Pressing **Search** will look for the files in the **Browse** list in the chosen directory, files that have been found will be removed from the **Files to find** list.
   Any searched directories in the **Browse** section will be shown by a folder and magnifying glass icon (🔍).
   Once the user has selected **Search** this button will be changed to a **STOP** button allowing the user to cancel a search. Any files that had been found before the search was cancelled will return to the unlocated state.
8. Clicking **OK**, then the files that were searched for and found will be located to that directory, accept all searches, and close the dialog box.

9. Clicking **Cancel** completely cancel all searches that have been done, and close the dialog without changing anything.

If you had done some searches and they clicked on cancel a confirmation box will appear as below;



Are you sure you want to cancel all file relocations and discard changes?

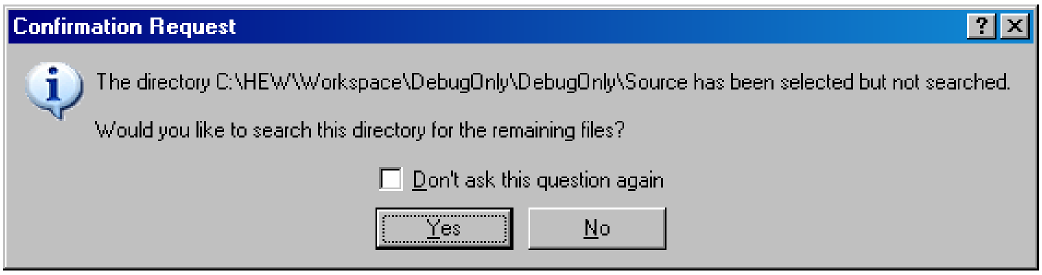
□ Don't ask this question again

If you select **Yes** this will invoke the Locate Files dialog box and close the dialog box. If you select **No** the dialog box will remain open allowing you to select OK. If the confirmation is selected not to be shown again, the default behavior will be **Yes**.

To open the confirmation dialog box again

1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Confirmation** tab.
3. Select the **Show dialog to search on OK for directory browse dialog** checkbox. This checkbox is selected by default.
4. Click **OK**.

If you select the OK button after selecting a directory, and not selecting search, a confirmation box will be shown, as below.



The directory C:\HEW\Workspace\DebugOnly\DebugOnly\Source has been selected but not searched.

Would you like to search this directory for the remaining files?

□ Don't ask this question again

If you select **Yes** a search will be made, and the Locate Files dialog box will remain open, allowing you to stop the search if needed.

Selecting **No** will invoke the message box and close the Locate Files dialog box.

If the confirmation is ticked not to be shown again, the default behavior will be **No**.

To open the confirmation dialog box again

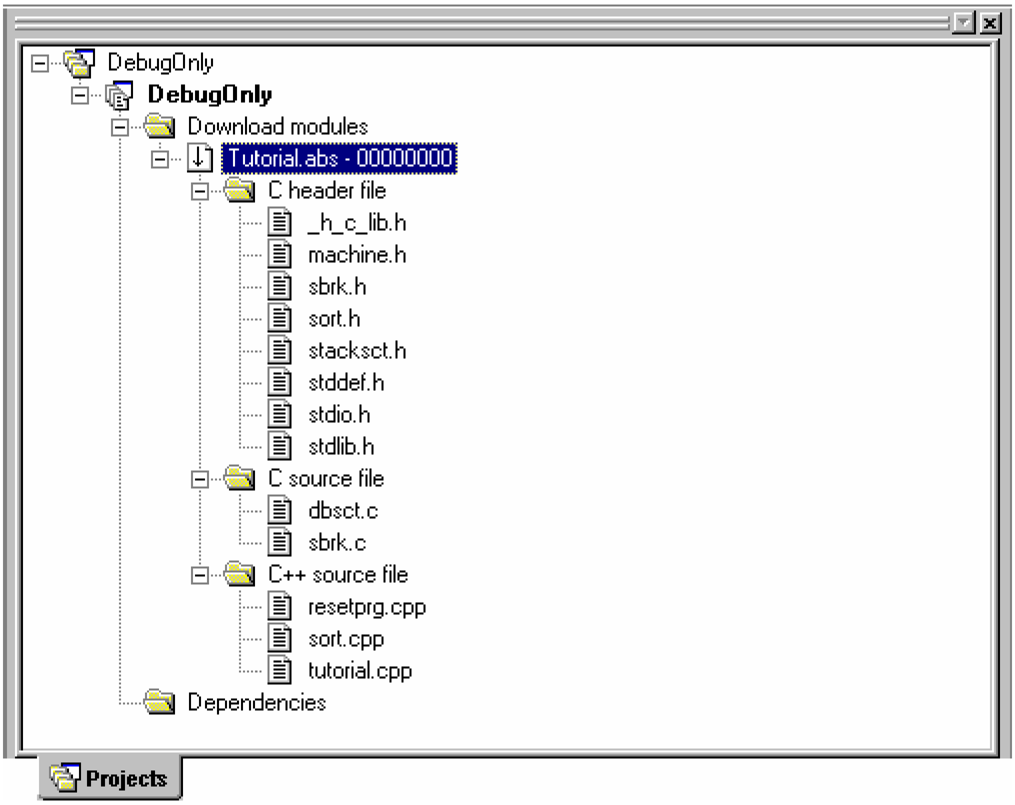
1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Confirmation** tab.
3. Select the **Show message when canceling for directory browse dialog** checkbox. This checkbox is selected by default.
4. Click **OK**.

(b) Showing a source tree

Files retrieved from the download module will be shown in the workspace window. The files will be shown under the download module.

Files will be grouped according to file type and shown under their relevant folder. If no file type for that file exists they will be grouped under their extension folder. When you double-click on a file, the external editor is launched with the file. If you wish to open the file in the editor window, see section 2.5, File extensions and file groups.

Files that could be located will be shown under the download module, shown in a normal icon (🗎).



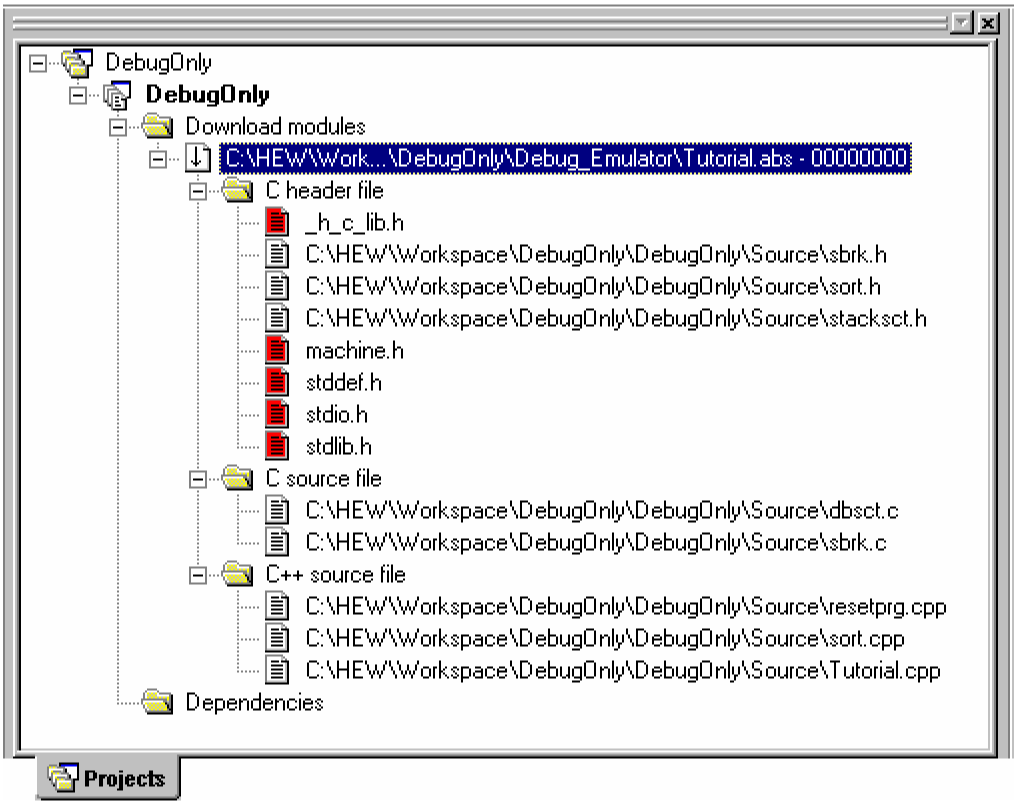
```
DebugOnly
  DebugOnly
    Download modules
      Tutorial.abs - 00000000
        C header file
          _h_c_lib.h
          machine.h
          sbrk.h
          sort.h
          stacksct.h
          stddef.h
          stdio.h
          stdlib.h
        C source file
          dbsct.c
          sbrk.c
        C++ source file
          resetprg.cpp
          sort.cpp
          tutorial.cpp
    Dependencies
```

Files that could not be located will be shown under the download module but will have a red icon (🟥).

To distinguish these files, you can view the full path of the file.

To view the full paths of all files

1. Right-click anywhere inside the **Projects** tab of the **Workspace** window.
2. Select the **Configure View** menu item. The **Configure View** dialog box will be displayed.
3. Select the **Show file paths** check box.
4. Click **OK**.



```
DebugOnly
  DebugOnly
    Download modules
      C:\HEW\Work...\DebugOnly\Debug_Emulator\Tutorial.abs - 00000000
        C header file
          _h_c_lib.h
          C:\HEW\Workspace\DebugOnly\DebugOnly\Source\sbrk.h
          C:\HEW\Workspace\DebugOnly\DebugOnly\Source\sort.h
          C:\HEW\Workspace\DebugOnly\DebugOnly\Source\stacksct.h
          machine.h
          stddef.h
          stdio.h
          stdlib.h
        C source file
          C:\HEW\Workspace\DebugOnly\DebugOnly\Source\dbsct.c
          C:\HEW\Workspace\DebugOnly\DebugOnly\Source\sbrk.c
        C++ source file
          C:\HEW\Workspace\DebugOnly\DebugOnly\Source\resetprg.cpp
          C:\HEW\Workspace\DebugOnly\DebugOnly\Source\sort.cpp
          C:\HEW\Workspace\DebugOnly\DebugOnly\Source\Tutorial.cpp
    Dependencies
```

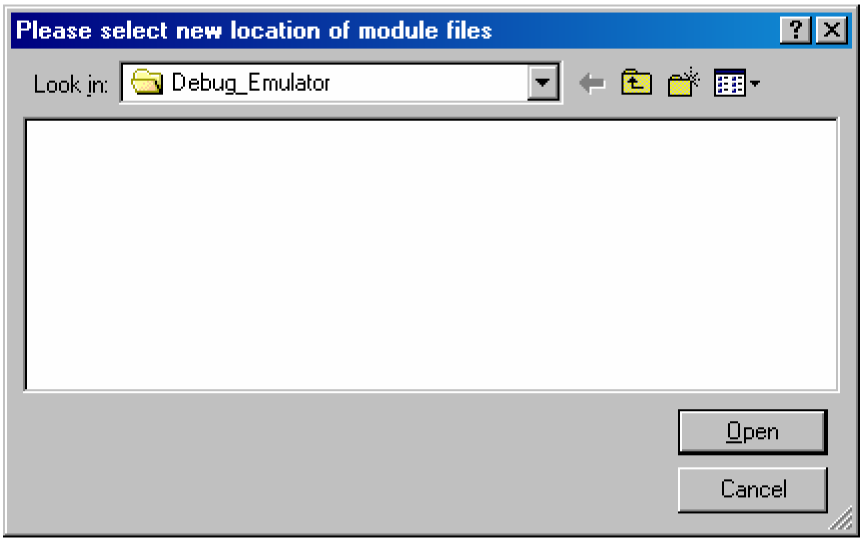
If you wish to relocate files in the download module to a different directory this is possible via pop-up menu.

(c) Relocating files to a different directory

If you wish to relocate the whole module i.e. want to relocate all files in the module to a different directory this is possible via the download module pop-up menu.

To relocate all files in the module to a different directory

1. Right-click on a download module in the **Projects** tab of the workspace window.
2. Select the **Relocate Module** menu item. Selecting this will bring up a standard windows directory open dialog box.



Please select new location of module files

Look in: Debug_Emulator

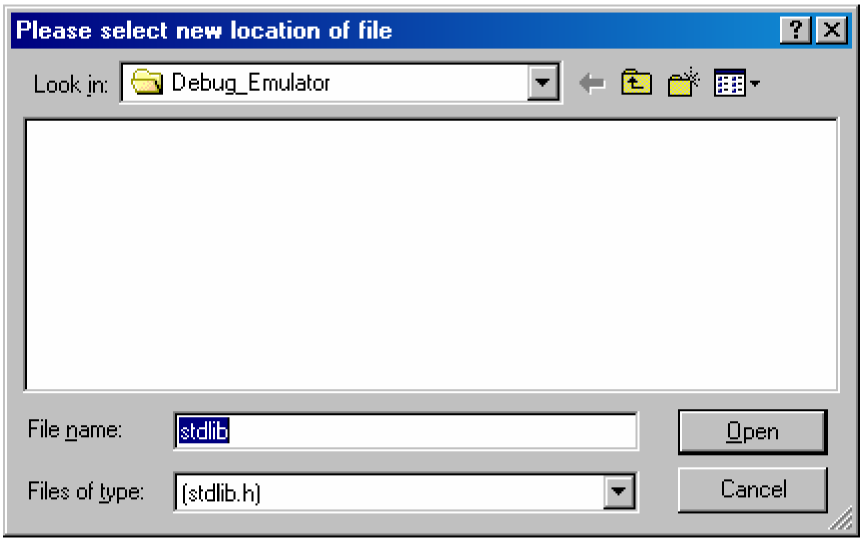
3. Select a directory, and then click the **Open** button.

Any files that could not be found in this directory will invoke the **Locate Files** dialog box, allowing you to search for the unlocated files.

You can also relocate files, via the pop-up menu item shown for files.

To relocate a file to a different directory

1. Right-click on a file in the **Projects** tab of the workspace window.
2. Select the **Relocate file(s)** menu item. Selecting this will bring up a standard windows file open dialog box.



Please select new location of file

Look in: Debug_Emulator

File name: stdlib

Files of type: (stdlib.h)

3. This dialog will only allow you to select a file of the same name.
4. Select a file, and then click the **Open** button.

You can select multiple files to be relocated, this will bring up the **Locate Files** dialog box with the files to be relocated in the files to find section.

(5) Displaying main functions on download

The HEW automatically displays the source file including main functions in source mode within the HEW editor after downloading modules.

To display main functions on download

1. Select [**Setup->Options**] to open the **Options** dialog box.
2. Select **Debug** tab.
3. Select the **Displaying main functions on download** checkbox. By default, this checkbox is not selected.
4. Click **OK**.
5. Download a module. For details on download, see section 17.1.5 (2), Downloading modules.

(6) Unloading of modules

It is possible to manually unload downloaded modules.

When a module is unloaded, its symbols are erased from the HEW debugging system, but the memory contents of the target remains unmodified. After a module has been unloaded, it cannot be debugged unless it is reloaded.

To unload modules

Select one of the following operations:

- Select the module, which you wish to unload, from the [**Debug->Unload Modules**] submenu, **OR**
- Right-click on the module in the Download modules folder on the Projects tab of the workspace window to display a pop-up menu. Select **Unload**. It is possible to select two or more modules.

To unload all modules

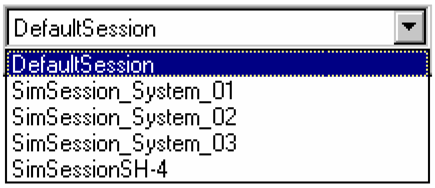
Select [**Debug->Unload Modules->All Downloaded Modules**].

Select the [**Debug->Debug Settings**] menu item. The Debug Settings dialog box will be invoked. This will unload the modules from the target in the order specified in **Download modules** list on the **Target** tab.

17.1.6 Debugger sessions

The HEW allows you to store all of your builder options into a configuration. This means that you can “freeze” all of the options and give them a name. In a similar way, HEW allows the user to store his debugger options in a session. Later on, you can select the session and all of the debugger options will be restored. These sessions allow the user to specify target, download modules and debug options. This means that potentially each session can be targeted at a different end platform.

This facility can allow you to have many different sessions, each with different debugger options defined. For example, it is possible to have each session using the same target but with slight variations in the session options. This can mean it is very easy for the user to switch session and modify such things as register values, or target settings such as clock speed. The figure below shows this principle. The five sessions share the same target but the sessions can be slightly different, with regard to the options defined. This means that both sessions can share the same download module and avoid an unnecessary rebuild. This is because sessions are not directly related to the build configuration data.



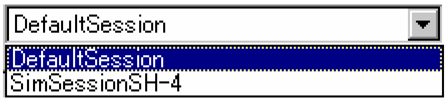
Each session’s data is stored in a separate file to the HEW project. You can then manipulate the data to share or modify as is required in the project.

(1) Selecting a session

To select a session

1. Choose the [**Debug->Debug Sessions**] menu option. The Debug Sessions dialog box will be displayed.

2. Select the session that you want to use from the **Current session** drop-down list box.
3. Click the **OK** button.



DefaultSession

DefaultSession
SimSessionSH-4

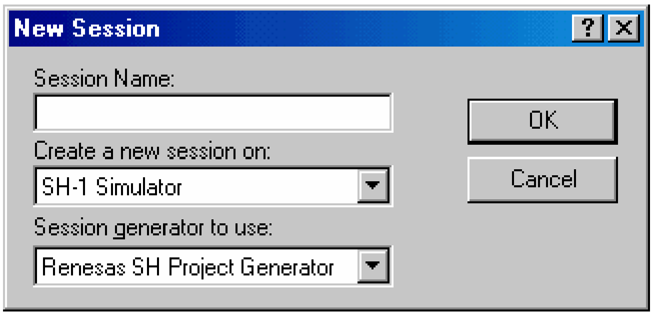
You can also select a different debug session by selecting it from the Current Session drop-down list on the **Standard** toolbar.

(2) Adding a session

You can now create a session with a target attached and setup ready for use. This session can be given a name and the target chosen.

To create a new session, with a target attached and setup

1. Choose the [**File->New Session**] menu option. The New Session dialog box is displayed.



Session Name:

Create a new session on: SH-1 Simulator

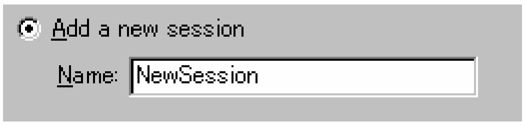
Session generator to use: Renesas SH Project Generator

2. Enter the name of a new session. This can be up to 32 characters in length and contain letters, numbers, and the underscore character. Especially, do not use a minus sign, or a space.
3. Select the target you wish to use in the new session.
4. Choose the **Session generator to use**. This should default to the correct selection. However sometimes there may be multiple generators that support the same target.
5. Click the **OK** button. This should launch the generation process, the process depends on the session generator that was selected. At this point an additional dialog box may be displayed for target setup options.
6. When finished, a new session is added to the current project. It should be available in the sessions drop-down list box on the main toolbar.

You can create a new empty session in the project directory. The session will use the session name as its new file name. If the file name already exists then an error is displayed.

To add a new empty session

1. Choose the [**Debug->Debug Sessions**] menu option. The Debug Sessions dialog box will be displayed.
2. Click the **Add** button. The **Add Session** dialog box will be displayed.
3. Click the **Add New Session** radio button.
4. Enter a name for the session.
5. Click the **OK** button.



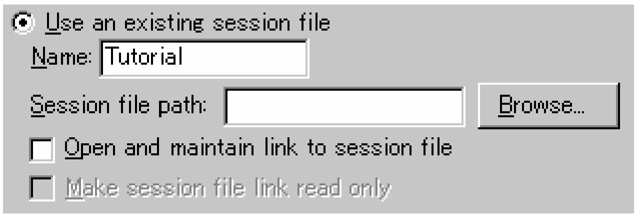
● Add a new session

Name: NewSession

You can import session data from another file and create a new session file in the project directory. All information is an exact copy of the file the data was imported from.

To import an existing session into a new session file

1. Choose the [**Debug->Debug Sessions**] menu option. The Debug Sessions dialog box will be displayed.
2. Click the **Add** button. The **Add Session** dialog box will be displayed.
3. Click the **Add An Existing Session File** radio button.
4. Enter a name for the session.
5. Browse to an existing session file location, which you would like to import into the current project.
6. Click the **OK** button.



● Use an existing session file

Name: Tutorial

Session file path:

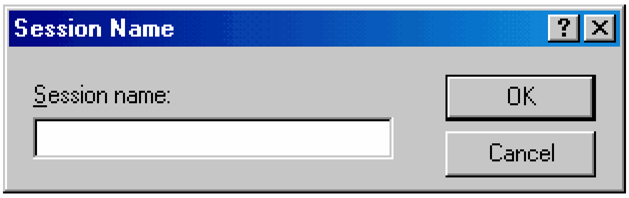
□ Open and maintain link to session file

□ Make session file link read only

This operation can also be achieved by using the [**File->Import session**].

To import an existing session using File->Import Session

1. Choose the [**File->Import Session**] menu option. The **Session Name** dialog box is displayed.



Session Name

Session name:

2. Enter the new session name.
3. Select the session file you wish to import into the new session.
4. Click **OK**. A new session as added with the same settings as the file you browsed to but with the new name.

(3) Importing a link to a session

You can add a new session to the HEW system but link to the session file in its location rather than importing or copying the file to the project directory. This is useful when sharing debugger information with other users in a network environment.

To import a link to an existing session file

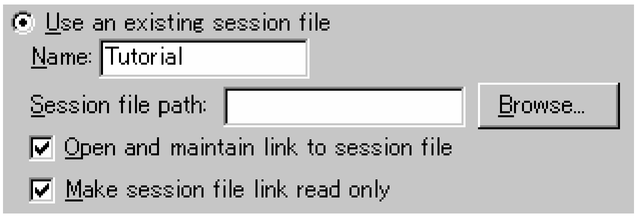
1. Choose the [**Debug->Debug Sessions**] menu option. The Debug Sessions dialog box will be displayed.
2. Click the **Add** button. The **Add new session** dialog box will be displayed.
3. Click the **Use an existing session file** radio button.
4. Enter a name for the session.
5. Browse to an existing session file location, which you would like to import into the current project.

6. Click the **Open and maintain link to session file** checkbox. This means the session will not be imported into the project directory but instead the HEW will link to the session location. This file location was entered in step 5 and it will save all of the session data in this location.
7. Click the OK button to close the Debug Sessions dialog box.

It is possible to make the link to session file read-only. This is useful if you are sharing debugger-setting files and you do not want data to be modified accidentally.

To import a link to an existing session file and make it read-only

1. Choose the [**Debug->Debug Sessions**] menu option. The Debug Sessions dialog box will be displayed.
2. Click the **Add** button. The **Add new session** dialog box will be displayed.
3. Click the **Use an existing session file** radio button.
4. Enter a name for the session.
5. Browse to an existing session file location, which you would like to import into the current project.
6. Click the **Open and maintain link to session file** checkbox.
7. Click the **Make session file link read only** checkbox. This means that the HEW will be unable to save changes to this session and will only be able to read the data when the session is opened.



● Use an existing session file

Name: Tutorial

Session file path:

☑ Open and maintain link to session file

☑ Make session file link read only

(4) Removing a session

To remove a session

1. Choose the [**Debug->Debug Sessions**] menu option. The Debug Sessions dialog box will be displayed.
2. Select the session you would like to remove.
3. Click the **Remove** button.
4. Click the OK button to close the Debug Sessions dialog box.

It is not possible to remove the current session.

(5) Making a session read-only

To make a session read only

□ Read only

1. Choose the [**Debug->Debug Sessions**] menu option. The Debug Sessions dialog box will be displayed.
2. Select the session you would like to view the properties for.
3. Click the **Properties** button. The properties dialog is displayed.

4. Click the **Read only** checkbox. This makes the link read only. This is useful if you are sharing debugger-setting files and you do not want data to be modified accidentally.
5. Click the OK button to close the Debug Sessions dialog box.

(6) Saving session information

To save a session

Select the [**File->Save Session**] menu item.

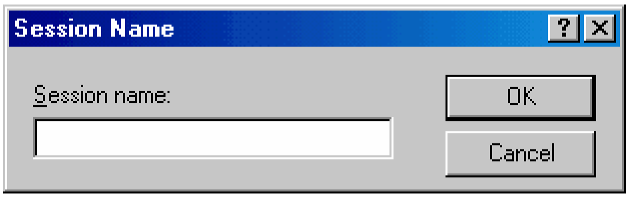
If you have the **Prompt before saving session** checkbox checked, a dialog box is displayed which asks you whether you wish to save the information. Clicking No loses the changes you made in the session. This checkbox is located in the [**Setup->Options**] dialog box on the **Workspace** tab.

To save a session with a different name

1. Choose the [**Debug->Debug Sessions**] menu option. The Debug Sessions dialog box will be displayed.
2. Select the session you would like to save.
3. Click the **Save as** button. The **Save Session** dialog box is displayed.
4. Browse to the new file location.
5. If you only want to export the session file to another location then leave the **Maintain link** checkbox unchecked. If you would like HEW to use this location instead of the current session location then check the **Maintain link** checkbox.
6. Click the OK button.

To save a session with a different name using File->Save Sessions As

1. Choose the [**File->Save Sessions As**] menu option. The **Session Name** dialog box is displayed.



2. Enter the new session name.
3. Click the OK button.

(7) Reloading session information

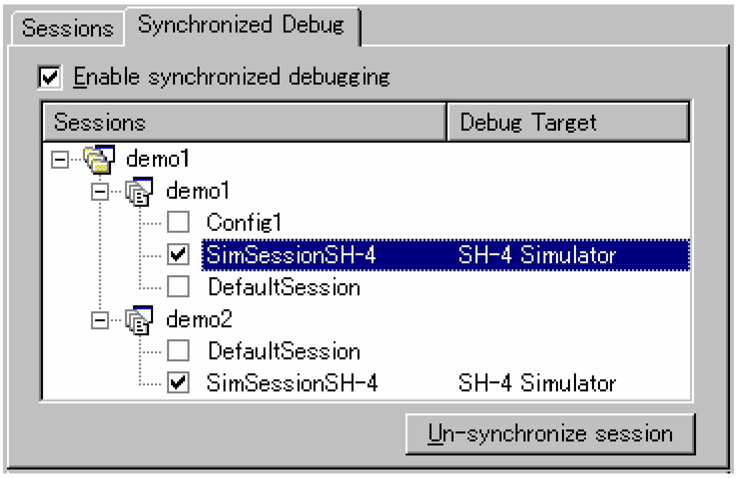
To reload a session

Select [**File->Refresh Session**].

Clicking this will lose any changes to your session currently and the reload the current session into HEW.

(8) Debugging multiple targets

For the method to debug multiple targets in synchronization, see section 17.17, Synchronizing multiple debugging platforms.



17.2 Viewing a Program

This section describes how to look at your program as source code and assembly language mnemonics.

The source code will be shown in the source mode in two types of windows. The main features are:

Editor Window

- A source file opened in the Editor window can be edited.
- When debugging and stepping through source code, each source file (which contains the current PC address) will be opened in a separate window as you step through code from one file to another. When stepping into an area of disassembly that has no associated source-code line, the source-mode will switch to disassembly mode.

Disassembly Window

- Any source file opened in the Disassembly window is read-only and cannot be edited. The Disassembly window pop-up menu will contain a menu-item to open the current file for editing.
- When debugging and stepping through source code, the Disassembly window will automatically switch to the relevant source file (which contains the current PC address) making this simple for you when debugging. Whilst the Disassembly window is open there will be no standard editor windows opened, the source files will be opened within the Disassembly window view as necessary. When stepping into an area of disassembly that has no associated source-code line, the source-mode will switch to disassembly mode.
- You can use disassembly source-mode as primary debug view. If this facility is available, the following actions will be performed:
  - If the Disassembly window is open and in Disassembly mode, then when a break occurs at a known source address, the Disassembly window will switch to source-mode.
  - If the Disassembly window is open, then when a break occurs, no other editor windows are opened.
  - When a break occurs, the window order will not be changed automatically.

  If you wish to continue step in Disassembly mode of the Disassembly window, select the [**Debug->Step mode->Assembly**].

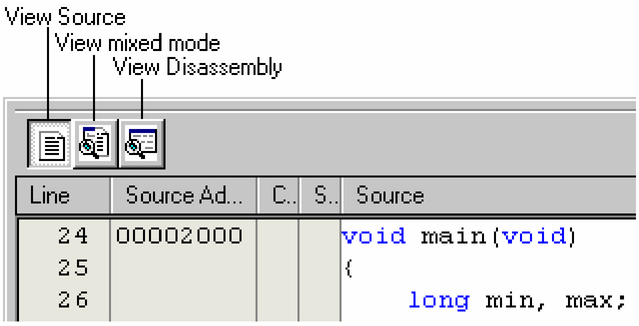
Note:

After a break occurs, the HEW displays the location of the program counter (PC). In some cases, for example if a project has been moved from its original path, the source files may not be automatically found. In this case the HEW will open a source file browser dialog to allow you to manually locate the file – this path will then be used to update any other source files in this debug project.

17.2.1 Opening the Editor window

To view a source file's code, double-click on its icon in the file tree, or right-click on the source file and click the **Open** option on the pop-up menu. The HEW opens the file in the editor.

Toolbar



```
24  00002000    void main(void)
25              {
26                  long min, max;
```

The editor in version 4.00 onwards has been enhanced to include an integrated disassembly view. This integrated view has a toolbar which allows the switching of mode. When each mode is available it is possible to click the button and change to the new view.

Three different modes are possible these are listed below:

Mode	Function
Source mode	This mode is the standard HEW editor. It allows source file editing and keywords are highlighted correctly if you are viewing source files. This view allows line numbers, addresses, breakpoints, bookmarks, and source codes to be viewed.
Mixed mode	The mixed mode facility in a source file is different to the disassembly window mixed mode. Instead of showing the continuous disassembly it shows the disassembly that is related to each line of source code. This view cannot be edited and is only available when the module is downloaded. This view allows line numbers, breakpoints, address, object codes, labels, and mixed codes (for showing both source and disassembly) to be viewed.
Disassembly mode	The disassembly mode shows the true continuous disassembly code in address order. This is the same as clicking the [**View->Disassembly**] menu item. This view is only available when a target is attached to the session. This view allows breakpoints, address, object codes, labels, and disassembled codes to be viewed.

Note:

It is not possible to switch from the source view to the mixed display under the following conditions.

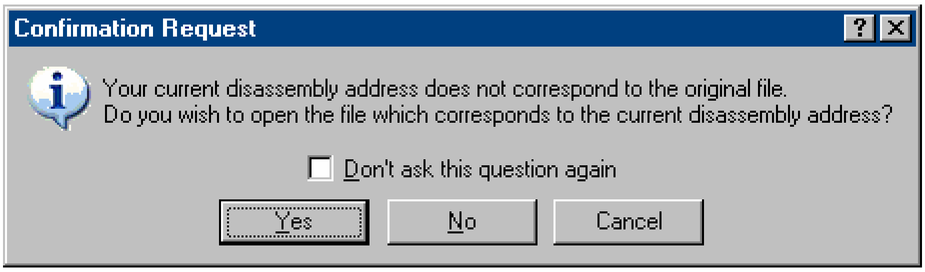
1. The target is not connected to the current session.
2. No download modules have been downloaded.
3. No debug information is available for the current project.
4. The currently displayed file has been edited and the changes not saved.

If an address included in the address range of a source file is then shifted to be in the range of another source file in **View Disassembly mode**, selecting **View Source mode** or **View mixed mode** displays the Confirmation Request dialog box shown below.

To open the new source file for the address, click **Yes**.

To view the source code in the previous mode, click **No**.

If you do not wish to select **View Source mode** or **View mixed mode**, click **Cancel**.



Confirmation Request

Your current disassembly address does not correspond to the original file.
Do you wish to open the file which corresponds to the current disassembly address?

☐ Don't ask this question again

If you do not wish to open this confirmation dialog box, select the **Don't ask this question** again checkbox.

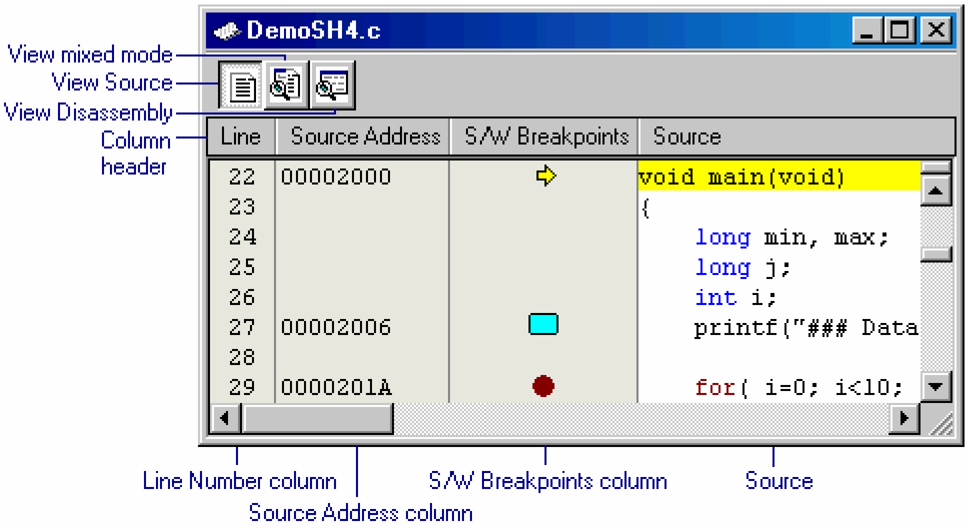
To open the confirmation dialog box again

1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Confirmation** tab.
3. Select the **Switch from disassembly to new source** checkbox. This checkbox is selected by default.
4. Click **OK**.

(1) Viewing the code

To view a source file's code, click the **View Source** button.

Window configuration



```
22  00002000   void main(void)
23             {
24                 long min, max;
25                 long j;
26                 int i;
27  00002006       printf("### Data
28
29  0000201A       for( i=0; i<10;
```

- Clicking another toolbar button switches the display mode. To view disassembly codes in mixed mode, click the "View mixed mode" button. To view disassembly codes in disassembly mode, click the "View Disassembly" button.
- The columns listed in the table below are on the left of the "Source" field.

Column Name	**Description**
Line	Displays the line number for the source file.
Source Address	After your program has been downloaded, the editor window displays the addresses for the current source file.
S/W Breakpoints	Display the PC location (⇨), breakpoints (●) and bookmark (▭). Setting PC breakpoint by double-click.

- The "Source" field includes codes highlighting the syntax.

Option

Right-clicking within the "Source" field opens a pop-up menu containing available options.

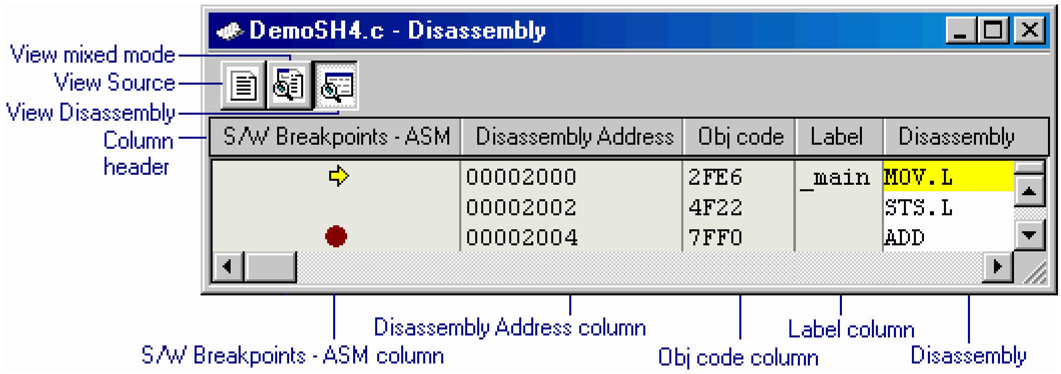
Pop-up menu options		Macro Recording	Function
Build File "<File Name>"		●	Builds the selected files.
Open '<File Name>'		-	Opens a file shown in the editor window.
Cut		-	Removes highlighted text and places it on the Windows® clipboard.
Copy		-	Places a copy of the highlighted text into the Windows® clipboard.
Paste		-	Copies the contents of the Windows® clipboard into the active window at the position of the insertion cursor.
Add File To Project		-	Adds file to a project.
Go to definition of '<Navigation Item Name>'		-	Shows the positions where navigation items of #defines, C functions, or C++ classes are defined.
List Members		-	While the smart editor function is enabled, selecting this option displays an alphabetical list of C #defines, C functions, or C++ classes as a pop-up window. Double-clicking on an item copies this character string and pastes it to the current cursor position.
Find		-	Finds text in the current file.
Replace		-	Replaces text in the current file.
Goto Line		-	Jumps to a line in a file.
Match Braces		-	Finds a matching brace.
Bookmarks	Toggle Bookmark	-	Sets a bookmark at the current line or clears a bookmark at the current line.
	Next Bookmark	-	Jumps to the next bookmark in the current file from the current line.
	Previous Bookmark	-	Jumps to the previous bookmark in the current file from the current line.
	Clear All Bookmarks	-	Clears all bookmarks in the current file.
Templates	Define Templates	-	Defines a template.
	Insert Template	-	Inserts a template.
Toggle Breakpoint		●	Sets or clears a software breakpoint at the line showing the address.
Enable/Disable Breakpoint		●	Enables or disables the current software breakpoint.
Define Column Format		-	Sets the status of editor and disassembly columns.
Columns	Column name	-	Sets the status of editor and disassembly columns.
Turn Header On/Off		-	Shows or hides the column header.
Instant Watch ***1**		-	Launches the Instant Watch dialog box with the name extracted from the view at the current text cursor (not mouse cursor) position.
Go To Cursor		●	Starts executing the user program at the current PC and continues until the PC equals the address indicated by the current text cursor (not mouse cursor) position.
Set PC Here		●	Changes the value of the Program Counter (PC) to the address at the row of the text cursor (not mouse cursor).
Display PC		-	Opens the source file or disassembly at the address of the PC.
View Disassembly		-	Opens a Disassembly window at the address mating the current source line.
Properties		-	Displays file properties.

***1.** Support for this function depends on the debugging platform.

(2) Viewing assembly-language code

The **Disassembly** mode shows at the same address of the cursor line as the **Source** mode. You can view assembly-language codes only when the debugging platform is connected to the session.

Window configuration



DemoSH4.c - Disassembly

S/W Breakpoints - ASM	Disassembly Address	Obj code	Label	Disassembly
⇨	00002000	2FE6	_main	MOV.L
	00002002	4F22		STS.L
●	00002004	7FF0		ADD

- Clicking another toolbar button switches the display mode. To view disassembly codes in mixed mode, click the "View mixed mode" button. To view source codes in source mode, click the "View Source" button.
- The columns listed in the table below are on the left of the "Disassembly" field.

Column Name	**Description**
S/W Breakpoints - ASM	Display the PC location (⇨), breakpoints (●). Setting PC breakpoint by double-click.
Disassembly Address	Display the disassembly address. Open the **Set Address** dialog box by double-click. Enter the address to jump to.
Obj code	Display the object codes.
Label	Display the Labels. This column is not available if no module has been downloaded.

- Double-clicking the "Disassembly" field (in disassembly mode) opens the **Assembler** dialog box. Enter assembly-language codes.

Option

Right-clicking within the "Disassembly" field opens a pop-up menu containing available options.

Pop-up menu options	Macro Recording	Function
Refresh	-	Acquires the latest disassembly information to update the contents of the window.
Lock Refresh	-	It is possible to lock a memory range in the disassembly (in disassembly mode) so that it does not refresh.
View Source	-	Launches editor at location in source.
Set Address	-	Enters a new start address.
Go To Cursor	●	Commences to execute the user program starting from the current PC address. The program will continue to run until the PC reaches the address indicated by the text cursor (not the mouse cursor) or another break condition is satisfied.
Display PC	-	Displays the disassembled codes at the PC location.
Set PC Here	●	Changes the value of the PC to the address indicated by the text cursor (not the mouse cursor).
Edit	-	Modifies the instruction at that address.
Find in Range	-	Searches the range for the specified text string.
Copy	-	Places a copy of the highlighted text into the Windows® clipboard.
Define Column Format	-	Sets the status of editor and disassembly columns.
Turn Header On/Off	-	Shows or hides the column header.
Save Disassembly Text	-	Saves the specific range.
Print	-	Prints the specific range.
Toggle Breakpoint	●	Sets or clears a software breakpoint at the line showing the address.
Enable/Disable Breakpoint	●	Enables or disables the current software breakpoint.

(3) Refreshing the disassembly view

Even if some external operation changes the memory contents shown in the **Disassembly** view, the HEW cannot detect the change. For example, if you use the external flash utility to program the range of memory being displayed, the **Disassembly** view will not be reflected.

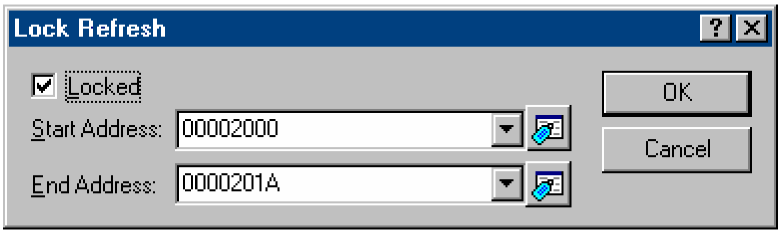
Right-click on the Disassembly filed (in disassembly mode) and select **Refresh**.

This acquires the latest information of the data being displayed and updates the contents of the view.

(4) Disassembly lock refresh

It is possible to lock a memory range in the disassembly view so that it does not refresh. This function is called the disassembly "lock refresh".

Right-click on the Disassembly field (in disassembly mode) and select **Lock Refresh**. The **Lock Refresh** dialog box will be displayed.



☑ Locked

Start Address: 00002000

End Address: 0000201A

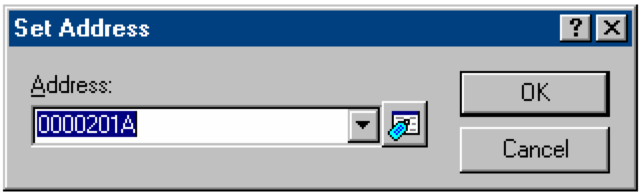
Check the **Locked** check box. The controls should now enable.

Select the start and end address that should be locked and cached so that no updates are displayed. Click **OK**. The view refreshes to only show the locked area.

(5) Looking at a specific address

When you are looking at your program in a Disassembly view, you may want to look at another area of your program's code. Rather than scrolling through a lot of code in the program, you can go directly to a specific address.

Right-click on the Disassembly field (in disassembly mode) and select the **Set Address** option. The **Set Address** dialog box will be displayed.

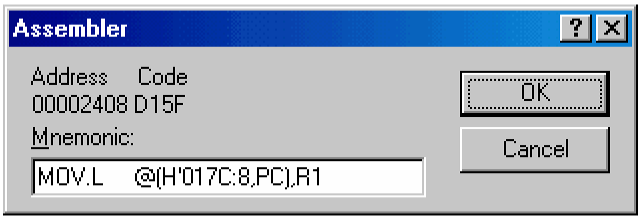


Address: 0000201A

Enter the address or label name in the edit box and either click the **OK** button or press the Enter key. The Disassembly view updates to show the code at the new address. When an overloaded function or a class name is entered, the **Select Function** dialog box opens for you to select a function (Support for this function depends on the debugging platform.).

(6) Modifying assembly-language code

You can modify the assembly-language code in the disassembly view (in disassembly mode) by double-clicking on the instruction that you wish to change. The **Assembler** dialog box will be displayed.



Address Code
00002408 D15F
Mnemonic:
MOV.L @(H'017C:8,PC),R1

The address, machine code and disassembled instructions are shown. Type the new instruction or edit the old instruction in the **Mnemonic** field. Pressing Enter will assemble the instruction into memory and move on to the next instruction. Clicking the **OK** button will assemble the instruction into memory and close the dialog box. Clicking the **Cancel** button or pressing ESC will close the dialog box.

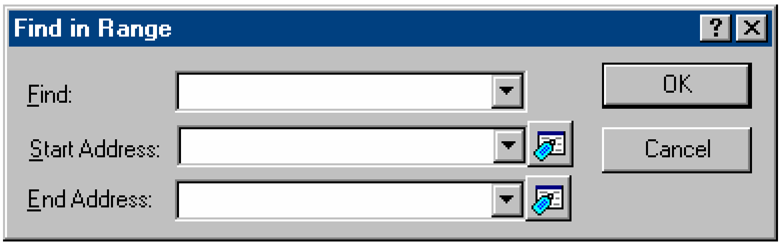
Note:

The assembly-language display is disassembled from the actual machine code in the debugging platform's memory. If the memory contents are changed the dialog box (and Disassembly view) will show the new assembly-language code, but the source view will be unchanged. This is true even if the source file contains assembler.

(7) Disassembly find in range

The disassembly find in range can be used to find a certain text string in the disassembly view between two addresses.

Right-click on the Disassembly field (in disassembly mode) and select **Find in Range**. The **Find in Range** dialog box is displayed.



Enter your search string, the start and end address that should be searched. Click **OK**. The view then selects the first instance of that string in the range.

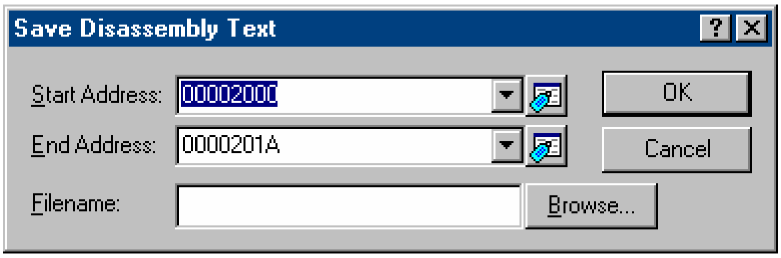
Note:

Subsequent find operations will find strings only in the paged disassembly area not the complete range.

(8) Saving disassembly text

The contents of the disassembly view can be saved by using the **Save Disassembly Text** menu from the **Disassembly** field (in disassembly mode) pop-up menu.

When **Save Disassembly Text** is selected the **Save Disassembly Text** dialog box is displayed that asks you the range of addresses to save.



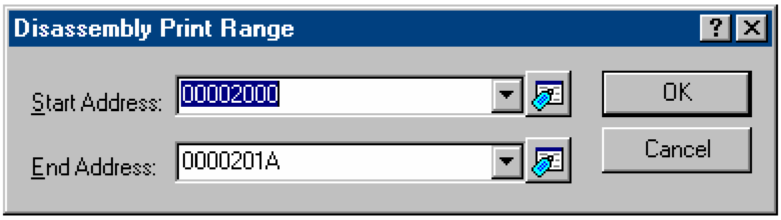
Start Address: 00002000
End Address: 0000201A
Filename:

Both a start and end address should be supplied. You also need to specify the full filename to save the information to. If needed you can browse to the file to use. Click **OK**.

(9) Printing the disassembly view

The disassembly view can be printed by using the [**File->Print**] menu or the **Print** toolbar button () when it is in focus or by using the menu **Print** on the **Disassembly** field (in disassembly mode) pop-up menu.

When **Print** is selected the **Disassembly Print Range** dialog box is displayed that asks you the range of addresses to print.


Disassembly Print Range

Start Address: 00002000

End Address: 0000201A

Both a start and end address should be supplied.

Clicking **OK** on this dialog box then passes the print selection to the standard print formatting and selection dialog box. From here you can choose your printer and page setup options.

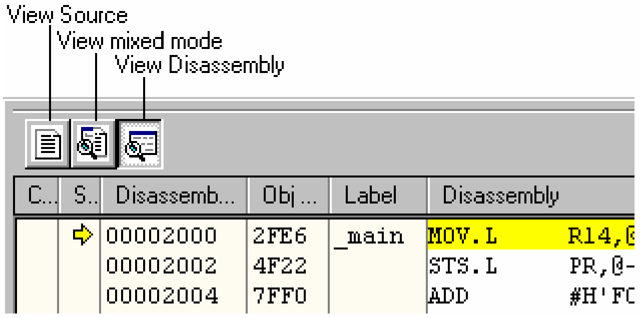
17.2.2 Opening the Disassembly window

If you wish to view code at assembly-language level, then select one of the following operations:

- Click on the **View Disassembly** toolbar button (), **OR**
- Choose the [**View->Disassembly**] menu option, **OR**
- Press CTRL+D.

The **Disassembly** window opens at the current program counter (PC) location.

Toolbar



C...	S..	Disassemb...	Obj ...	Label	Disassembly
	⇨	00002000	2FE6	_main	MOV.L R14,@
		00002002	4F22		STS.L PR,@-
		00002004	7FF0		ADD #H'FC

The disassembly window in version 4.02 onwards has been enhanced to include a source view.

This integrated view has a toolbar which allows the switching of mode. When each mode is available it is possible to click the button and change to the new view.

Three different modes are possible these are listed below:

Mode	Function
Source mode	The source file relating to the current PC address will be opened if available. It allows keywords are highlighted correctly if you are viewing source files. This view is read-only and cannot be edited. This view is only available when the module is downloaded. This view allows line numbers, address, breakpoints, and source codes to be viewed.
Mixed mode	The mixed mode facility in the disassembly window is different to a source file mixed mode. Instead of showing the continuous disassembly it shows the disassembly that is related to each line of source code. This view can be edited the assembly-language code and is only available when the module is downloaded. This view allows breakpoints, address, object codes, labels, and disassembled codes to be viewed.
Disassembly mode	The disassembly mode shows the true continuous disassembly code in address order. This view can be the assembly-language code and is only available when a target is attached to the session. This view allows breakpoints, address, object codes, labels, and disassembled codes to be viewed.

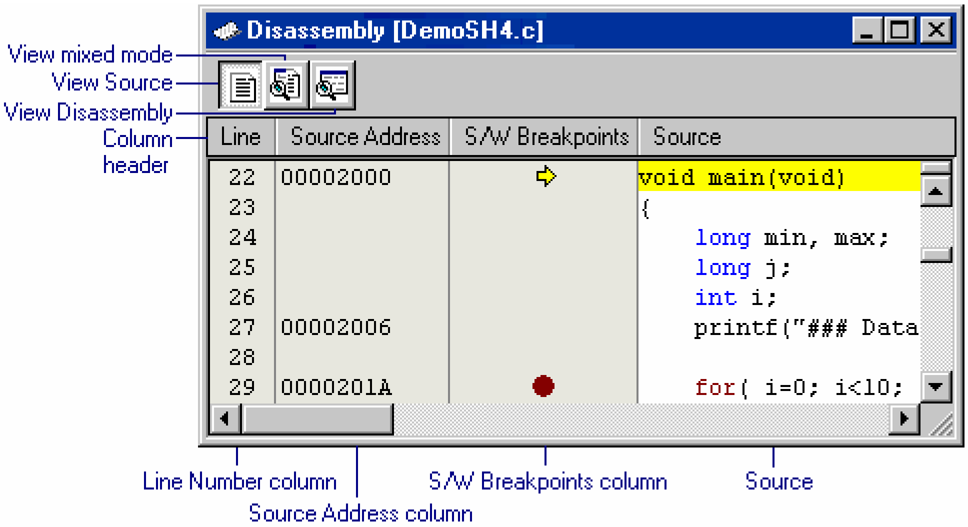
(1) Viewing the code

When switching to the source mode from within the disassembly window, the source file relating to the current PC address will be opened if available. Otherwise either the most recently open disassembly window source file will be re-opened where possible or a prompt will be given to select a source file to be opened.

Any text-based files may be opened in the disassembly window and not just project source files. Any source file opened in the disassembly window is read-only and cannot be edited; it is only intended for the viewing and debugging of source files.

This view is only available when the module is downloaded.

Window configuration



Line	Source Address	S/W Breakpoints	Source
22	00002000	⇨	void main(void)
23			{
24			long min, max;
25			long j;
26			int i;
27	00002006		printf("### Data
28			
29	0000201A	●	for(i=0; i<10;

- Clicking another toolbar button switches the display mode. To view disassembly codes in mixed mode, click the "View mixed mode" button. To view disassembly codes in disassembly mode, click the "View Disassembly" button.
- The columns listed in the table below are on the left of the "Source" field.

Column Name	Description
Line	Displays the line number for the source file.
Source Address	After your program has been downloaded, the source view displays the addresses for the current source file.
S/W Breakpoints	Display the PC location (⇨), and breakpoints (●). Setting PC breakpoint by double-click.

- The "Source" field includes codes highlighting the syntax.

Option

Right-clicking within the "Source" field opens a pop-up menu containing available options.

Pop-up menu options	Macro Recording	Function
Open Source File For Editing	-	Opens a editor window containing the file for editing.
Open File in Source Mode	-	Opens a file shown in the disassembly window.
Copy	-	Places a copy of the highlighted text into the Windows® clipboard.
Find	-	Finds text in the current file.
Goto Line	-	Jumps to a line in a file.
Toggle Breakpoint	●	Sets or clears a software breakpoint at the line showing the address.
Enable/Disable Breakpoint	●	Enables or disables the current software breakpoint.
Turn Header On/Off		Shows or hides the column header.
Define Column Format	-	Sets the status of editor and disassembly columns.
Columns Column name	-	Sets the status of editor and disassembly columns.
Instant Watch ***1**	-	Launches the Instant Watch dialog box with the name extracted from the view at the current text cursor (not mouse cursor) position.
Go To Cursor	●	Starts executing the user program at the current PC and continues until the PC equals the address indicated by the current text cursor (not mouse cursor) position.
Set PC Here	●	Changes the value of the PC to the address at the row of the text cursor (not mouse cursor).
Display PC	-	Opens the source file or disassembly at the address of the PC.
Properties	-	Displays file properties.

***1.** Support for this function depends on the debugging platform.

(2) Opening a source file for editing

The Disassembly source-mode allows only to view and debug source files, and no editing is permitted from within this view.

To edit a source file that is currently open in the Disassembly source-mode view, the pop-up menu item **Open Source File for Editing** must be selected. This will open an editor window containing the file for editing.

Alternatively you may open the file for editing manually via the Workspace Window or by the main menu [**File->Open**].

(3) Opening a source file in the Disassembly window

Opening a source file is possible by selecting **Open File in Source Mode** from the pop-up menu in Disassembly window. This menu item is available on the pop-up menu in all view-modes within the Disassembly window.

Upon selecting this menu item a file-browse dialog will be displayed prompting for the source file to be opened. Any text-based file may be selected and not just source files.

(4) Using disassembly source-mode as primary debug view

It is possible to use disassembly source-mode as primary debug view.

To use disassembly source-mode as primary debug view

1. Select the [**Setup->Options**] menu item. The **Options** dialog box will be displayed.
2. Select the **Debug** tab.
3. Select the **Use disassembly source-mode as primary debug view** checkbox. By default, this checkbox is not selected.
4. Click **OK**.

If this option is set, the following actions will be performed:

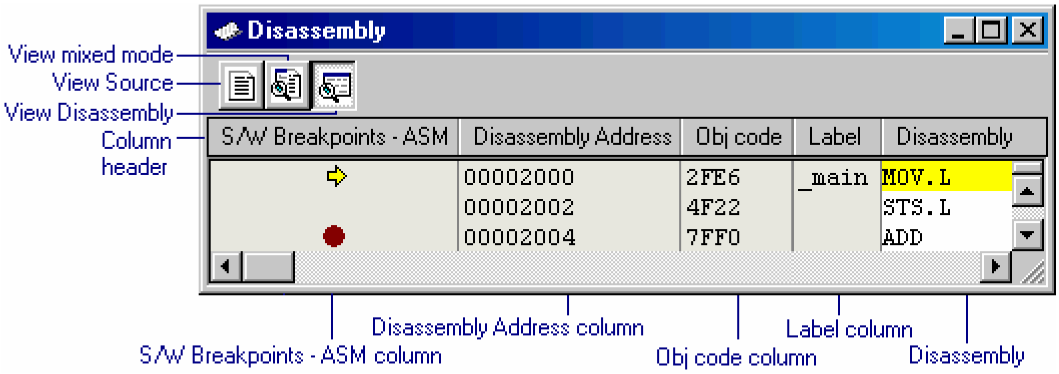
- If the Disassembly window is open and in Disassembly mode, then when a break occurs at a known source address, the Disassembly window will switch to source-mode.
- If the Disassembly window is open, then when a break occurs, no other editor windows are opened.
- When a break occurs, the window order will not be changed automatically.

If you wish to continue step in Disassembly mode of the Disassembly window, select the [**Debug->Step mode->Assembly**].

(5) Viewing assembly-language code

The **Disassembly** mode shows at the current PC location. You can view assembly-language codes only when the debugging platform is connected to the session.

Window configuration



S/W Breakpoints - ASM	Disassembly Address	Obj code	Label	Disassembly
⇨	00002000	2FE6	_main	MOV.L
	00002002	4F22		STS.L
●	00002004	7FF0		ADD

- Clicking another toolbar button switches the display mode. To view disassembly codes in mixed mode, click the "View mixed mode" button. To view source codes in source mode, click the "View Source" button.
- The columns listed in the table below are on the left of the "Disassembly" field.

Column Name	**Description**
S/W Breakpoints - ASM	Display the PC location (⇨), breakpoints (●). Setting PC breakpoint by double-click.
Disassembly Address	Display the disassembly address. Open the **Set Address** dialog box by double-click. Enter the address to jump to.
Obj code	Display the object codes.
Label	Display the Labels. This column is not available if no module has been downloaded.

- Double-clicking the "Disassembly" field opens the **Assembler** dialog box. Enter assembly-language codes.

Option

Right-clicking within the "Disassembly" field opens a pop-up menu containing available options.

Pop-up menu options	Macro Recording	Function
Open File in Source Mode	-	Opens a file shown in the disassembly window.
Refresh	-	Acquires the latest disassembly information to update the contents of the window.
Lock Refresh	-	It is possible to lock a memory range in the disassembly so that it does not refresh.
View Source	-	Launches editor at location in source.
Set Address	-	Enters a new start address.
Go To Cursor	●	Commences to execute the user program starting from the current PC address. The program will continue to run until the PC reaches the address indicated by the text cursor (not the mouse cursor) or another break condition is satisfied.
Display PC	-	Displays the disassembled codes at the PC location.

Set PC Here	●	Changes the value of the PC to the address indicated by the text cursor (not the mouse cursor).
Edit	-	Modifies the instruction at that address.
Find in Range	-	Searches the range for the specified text string.
Copy	-	Places a copy of the highlighted text into the Windows® clipboard.
Define Column Format	-	Sets the status of editor and disassembly columns.
Turn Header On/Off	-	Shows or hides the column header.
Save Disassembly Text	-	Saves the specific range.
Print	-	Prints the specific range.
Toggle Breakpoint	●	Sets or clears a software breakpoint at the line showing the address.
Enable/Disable Breakpoint	●	Enables or disables the current software breakpoint.

(6) Refreshing the disassembly view

Even if some external operation changes the memory contents shown in the **Disassembly** view, the HEW cannot detect the change. For example, if you use the external flash utility to program the range of memory being displayed, the **Disassembly** view will not be reflected.

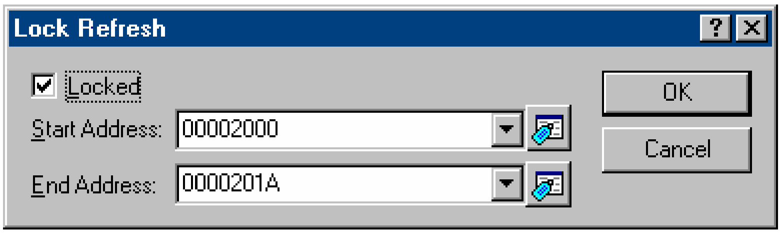
Right-click on the Disassembly filed (in disassembly mode or mixed mode) and select **Refresh**.

This acquires the latest information of the data being displayed and updates the contents of the view.

(7) Disassembly lock refresh

It is possible to lock a memory range in the disassembly view so that it does not refresh. This function is called the disassembly "lock refresh".

Right-click on the Disassembly field (in disassembly mode or mixed mode) and select **Lock Refresh**. The **Lock Refresh** dialog box will be displayed.



☑ Locked

Start Address: 00002000

End Address: 0000201A

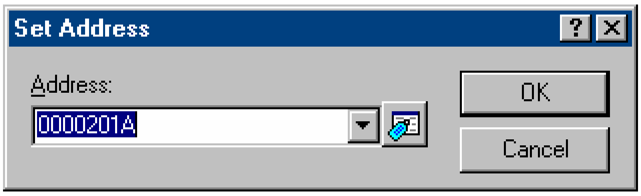
Check the **Locked** check box. The controls should now enable.

Select the start and end address that should be locked and cached so that no updates are displayed. Click **OK**. The view refreshes to only show the locked area.

(8) Looking at a specific address

When you are looking at your program in a Disassembly view, you may want to look at another area of your program's code. Rather than scrolling through a lot of code in the program, you can go directly to a specific address.

Right-click on the Disassembly field (in disassembly mode or mixed mode) and select the **Set Address** option. The **Set Address** dialog box will be displayed.

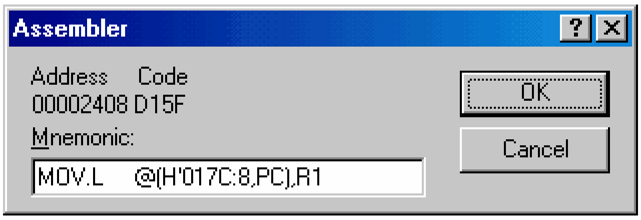


Address: 0000201A

Enter the address or label name in the edit box and either click the **OK** button or press the Enter key. The Disassembly view updates to show the code at the new address. When an overloaded function or a class name is entered, the **Select Function** dialog box opens for you to select a function (Support for this function depends on the debugging platform.).

(9) Modifying assembly-language code

You can modify the assembly-language code in the disassembly view (in disassembly mode or mixed mode) by double-clicking on the instruction that you wish to change. The **Assembler** dialog box will be displayed.



Address Code
00002408 D15F
Mnemonic:
MOV.L @(H'017C:8,PC),R1

The address, machine code and disassembled instructions are shown. Type the new instruction or edit the old instruction in the **Mnemonic** field. Pressing Enter will assemble the instruction into memory and move on to the next instruction. Clicking the **OK** button will assemble the instruction into memory and close the dialog box. Clicking the **Cancel** button or pressing ESC will close the dialog box.

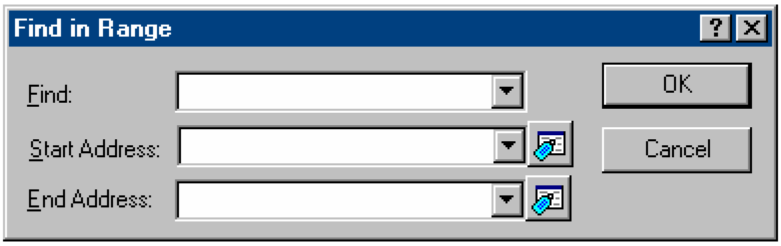
Note:

The assembly-language display is disassembled from the actual machine code in the debugging platform's memory. If the memory contents are changed the dialog box (and Disassembly view) will show the new assembly-language code, but the source view will be unchanged. This is true even if the source file contains assembler.

(10)Disassembly find in range

The disassembly find in range can be used to find a certain text string in the disassembly view between two addresses.

Right-click on the Disassembly field (in disassembly mode or mixed mode) and select **Find in Range**. The **Find in Range** dialog box is displayed.



Enter your search string, the start and end address that should be searched. Click **OK**. The view then selects the first instance of that string in the range.

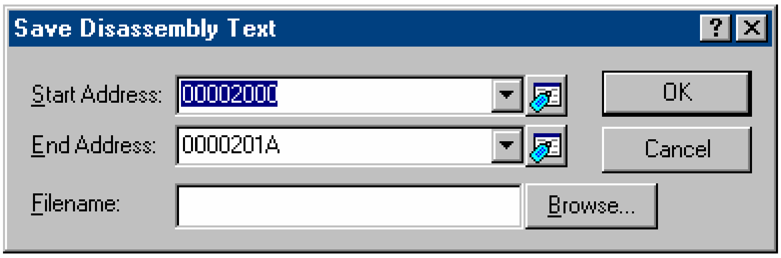
Note:

Subsequent find operations will find strings only in the paged disassembly area not the complete range.

(11)Saving disassembly text

The contents of the disassembly view can be saved by using the **Save Disassembly Text** menu from the **Disassembly** field (in disassembly mode or mixed mode) pop-up menu.

When **Save Disassembly Text** is selected the **Save Disassembly Text** dialog box is displayed that asks you the range of addresses to save.



Start Address: 00002000

End Address: 0000201A

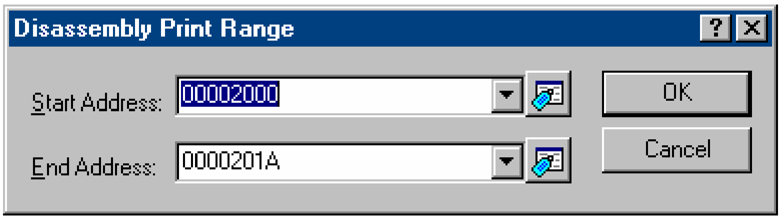
Filename:

Both a start and end address should be supplied. You also need to specify the full filename to save the information to. If needed you can browse to the file to use. Click **OK**.

(12)Printing the disassembly view

The disassembly view can be printed by using the [**File->Print**] menu or the **Print** toolbar button (🖨) when it is in focus or by using the menu **Print** on the **Disassembly** field (in disassembly mode or mixed mode) pop-up menu.

When **Print** is selected the **Disassembly Print Range** dialog box is displayed that asks you the range of addresses to print.



Start Address: 00002000

End Address: 0000201A

Both a start and end address should be supplied.

Clicking **OK** on this dialog box then passes the print selection to the standard print formatting and selection dialog box. From here you can choose your printer and page setup options.

17.2.3 Looking at the current PC position

To automatically display the program counter (PC) click the **Display PC** toolbar button (PC), or select the [**Debug->Display PC**] menu item. This will open the editor or disassembly at the current PC.

17.2.4 Highlighting the line at the PC

In the Editor and Disassembly windows, an icon in the **S/W Breakpoints** column indicates the PC location. The source or assembly-language code at the PC can be highlighted.

To highlight the line at the PC location

1. Select the [**Setup->Options**] menu item. The **Options** dialog box will be displayed.
2. Select the **Debug** tab.
3. Select the **Enable line highlight for PC position** checkbox. By default, this checkbox is selected.
4. Click **OK**.

The color of the line can be customized in the **Format Views** dialog box. The text (foreground) is in black and the background color is yellow by default.

To change the colors of the line at the PC

1. Select the [**Setup->Format Views**] menu option. The **Format Views** dialog box will be displayed.
2. Select an item, for which you wish to change the color, from the left-hand tree in the dialog box and expand it.

- If you are opening the editor window or disassembly window in source mode, select **Source** and expand it.
- If you are opening the editor window or disassembly window in mixed mode or disassembly mode, select **Disassembly** and expand it.

3. Select the **PC Line Highlight** category.
4. Change the selection in the **Foreground** and **Background** lists of the **Color** tab.
5. Click **OK**.

17.3 Operating memory

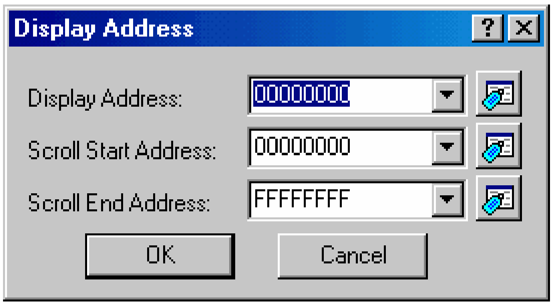
This section describes how to look at memory areas in the CPU's address space. How to look at a memory area in different formats, how to fill and move a memory block, and how to load and verify a memory area with a disk file are described.

17.3.1 Opening the Memory window

The Memory Window displays the contents of contiguous memory.

To open the **Memory** view, click the Memory toolbar button (), or select the [**View->CPU->Memory**] menu option.

You can specify the display start address and the scroll range at opening. The **Display Address** dialog box is opened. Enter the **Display Address**, **Scroll Start Address** and **Scroll End Address**.



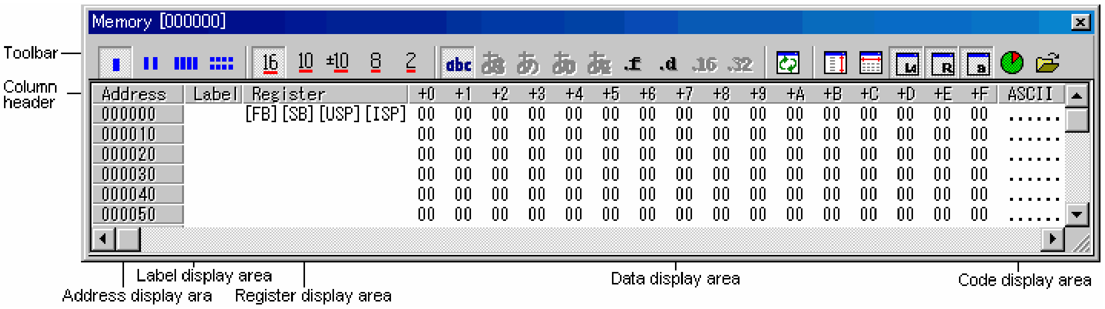
Display Address: 00000000

Scroll Start Address: 00000000

Scroll End Address: FFFFFFFF

Click the OK button or press the Enter key, and the dialog box closes and the **Memory** window opens. The display can be scrolled within the range of the entered display scroll start and end addresses.

Window configuration



```
Address  Label  Register            +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F ASCII
000000          [FB][SB][USP][ISP]  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ......
000010                              00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ......
000020                              00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ......
000030                              00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ......
000040                              00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ......
000050                              00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ......
```

- **+n** shows memory data read from **Address** and ‘n’ means the offset value from the first address of the row.
- **Code** shows the code names.
- **Register** shows the name of the register allocated to the first address of the memory data displayed on this row.
- In-place edit in the address display area/data display area/code display area. (Macro Recording)
- Double-clicking the address display area/label display area open a dialog, which allow you to change the display start address.
- A dialog box which allows you to change the memory data at the clicked address by double-clicking the data display area/code data display area. (Macro Recording)

Option

Right-clicking displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up menu options		Toolbar button	Macro Recording	Function
Set			●	Set data at specified address.
Fill			●	Fill specified memory block with data.
Move			●	Move specified memory block to.
Compare *			●	Comparing the contents of two memory blocks.
Test *			-	Testing an area of memory.
Save Memory contents			-	Saving memory contents in a text file.
Search *			-	Finding a value in memory.
Search Next *			-	Finding a next value in memory.
Address			-	Specify display starting address.
Scroll Area			-	Specify scroll range.
Register *	(xxxxx)	-	-	Starting address to value of the register.
Followed Stack Pointer *			-	Keep tracking of the stack pointer position.
Set Start Up Symbol			-	Changing the program display position Immediately after downloading.
Refresh			-	Update of the window contents.
Lock Refresh		-	-	Disabling update of the window contents.
Data Length	1byte	▮	-	Display in 1-byte units.
	2bytes	▮▮	-	Display in 2-bytes units.
	4bytes	▮▮▮▮	-	Display in 4-bytes units.
	8bytes	⁞⁞⁞⁞	-	Display in 8-bytes units.

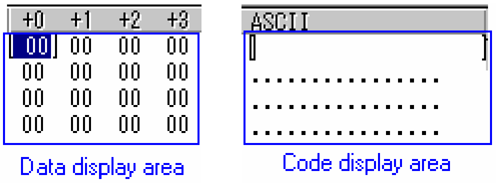
Radix	Hex	16	-	Display in hexadecimal.
	Dec	10	-	Display in decimal.
	Signed Dec	±10	-	Display in signed decimal.
	Oct	8	-	Display in octal.
	Bin	2	-	Display in binary.
Code	ASCII	abc	-	Displaying memory as ASCII characters (default).
	SJIS/JIS/UNICODE/EUC	-	-	Not supported.
	Float	.f	-	Displaying memory as single-precision floating point.
	Double	.d	-	Displaying memory as double-precision floating point.
	16bit Fixed	.16	-	Displaying memory as 16 bit fixed.
	32bit Fixed	.32	-	Displaying memory as 32 bit fixed.
	24bit Accum	-	-	Displaying memory as 24 bit accumulate.
	40bit Accum	-	-	Displaying memory as 40 bit accumulate.
Layout	Label	L	-	Switch display or non-display of Label area.
	Register	R	-	Switch display or non-display of Register area.
	Code	a	-	Switch display or non-display of Code area.
Column		▦	-	Changing the number of digits displayed.
Coverage *	Enable	◔	-	Switching display or non-display of measurement result.
Save		💾	●	Saving memory contents in a file.
Load		📂	●	Loading a memory area from a file.
Split		-	-	Saving an area of memory.
Toolbar display		-	-	Shows or hides the toolbar.
Customize toolbar		-	-	Customizes toolbar buttons.

***.** Support for this function depends on the debugging platform.

17.3.2 Modifying memory contents

To change the contents of memory, follow the procedure below.

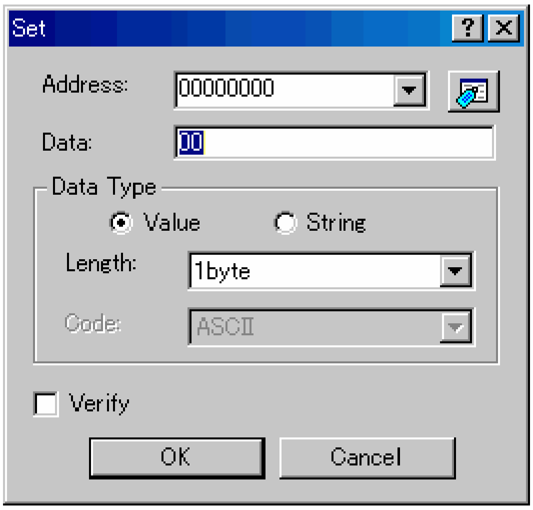
1. In-place edit in the data display area/code display area.



```
+0  +1  +2  +3      ASCII
00  00  00  00
00  00  00  00      ................
00  00  00  00      ................
00  00  00  00      ................
```

2. To change the contents of memory, open the **Set** dialog box by selecting one of the following operations:

- Double-click the data display area/code display area you want to change, **OR**
- Select the data you want to change and choose the **Set** option from the pop-up menu.



Set

Address: 00000000

Data: 00

Data Type

● Value ○ String

Length: 1byte

Code: ASCII

□ Verify

Enter the value (value or character) to be set in the **Data** field. Select the **Verify** check box. Support for verify function depends on the debugging platform.

When setting the value

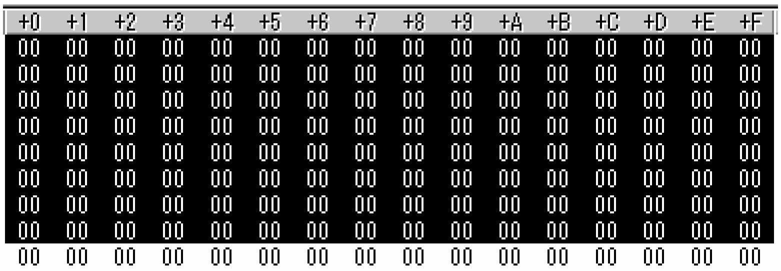
Click the **Value** button in the **Data Type** group. Specify the data length in the **Length** field.

When setting the character

Click the **String** button in the **Data Type** group. Specify the character code in the **Code** field.

17.3.3 Selecting a memory range

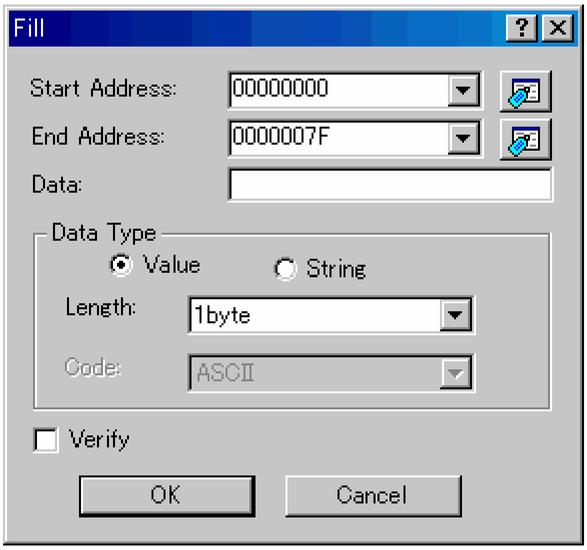
If the memory address range is in the **Memory** view, you can select the range by clicking on the first memory unit (depending on your **Memory** view display choice) and dragging the mouse to the last unit. The selected range is highlighted.



```
+0  +1  +2  +3  +4  +5  +6  +7  +8  +9  +A  +B  +C  +D  +E  +F
00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
```

17.3.4 Filling an area of memory with constant data

You can set the contents of a range of memory addresses to a value using the memory fill feature. Select an address range to fill in the Memory window by dragging the mouse. Choose the **Fill** option from the pop-up menu of the memory window. The **Fill** dialog box is displayed.



Start Address: 00000000

End Address: 0000007F

Data:

Data Type

● Value ○ String

Length: 1byte

Code: ASCII

□ Verify

Enter the data (value or character) to be filled in the **Data** field.

Select the **Verify** check box. Support for verify function depends on the debugging platform.

If you did not drag the address range to be filled, you must enter the start/end address. The end address can also be prefixed by a plus (+); the end address will become the (start address) + (entered value).

When specifying the value

Click the **Value** button in the **Data Type** group. Specify the data length in the **Length** field.

When setting the character

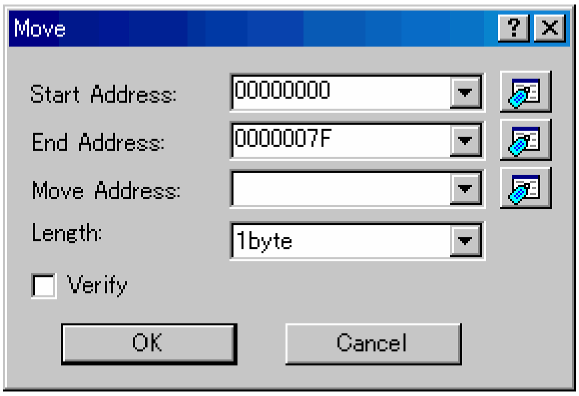
Click the **String** button in the **Data Type** group. Specify the character code in the **Code** field.

When the display data length is two bytes, two bytes' worth of a character can be specified.

Please use the **Set** dialog box to specify the character string. (Select menu **Set**)

17.3.5 Copying an area of memory

You can copy an area of memory using the memory copy feature. Select a copy-source address range in the Memory window by dragging the mouse. Choose the **Move** option from the pop-up menu of the memory window. The **Move** dialog box is displayed.



Start Address: 00000000

End Address: 0000007F

Move Address:

Length: 1byte

□ Verify

Enter the copy destination start address in the **Move Address** field.

Select the **Verify** check box. Support for verify function depends on the debugging platform.

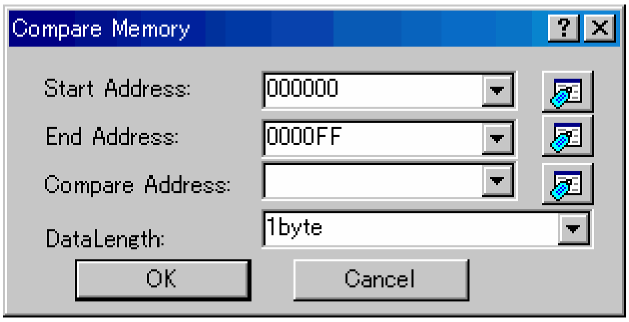
If you did not drag the copy-source address range, you must enter the start/end address. The end address can also be prefixed by a plus (+); the end address will become the (start address) + (entered value).

Drag and Drop

Type of dropped data	Operation
Selected range on the Memory Window's Data area	Copy the contents of a selected range of data to an area starting from the dropped position.

17.3.6 Comparing the memory contents

The contents of two memory blocks can be compared. Select a source address range in the Memory window by dragging the mouse. Choose the **Compare** option from the pop-up menu of the memory window. The **Compare Memory** dialog box is displayed.



Compare Memory

Start Address: 000000

End Address: 0000FF

Compare Address:

DataLength: 1byte

Enter the start address of the destination memory area in the **Compare Address** field and the data length in the **Data Length** field. If you did not drag the comparison-source address range, you must enter the start/end address. The end address can also be prefixed by a plus (+); the end address will become the (start address) + (entered value).

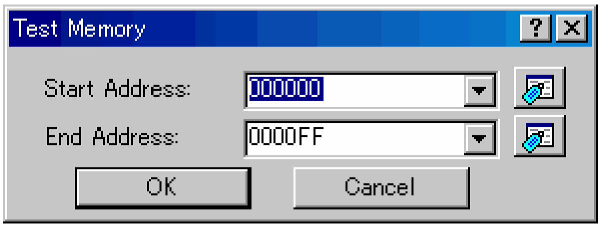
If there is a mismatch, the address where it was found is displayed in a message box.

When the contents of the two memory areas match at comparison, the message "**Comparison successful.**" appears.

Support for this function depends on the debugging platform.

17.3.7 Testing an area of memory

You can test an area of memory in the address space using the **Memory Test** feature. Select an address range to test in the Memory window by dragging the mouse. Choose the **Test** option from the pop-up menu of the memory window. The **Test Memory** dialog box is displayed.



Test Memory

Start Address: 000000

End Address: 0000FF

If you did not drag the address range to be tested, you must enter the start/end address. The end address can also be prefixed by a plus (+); the end address will become the (start address) + (entered value).

Support for this function depends on the debugging platform.

Note:

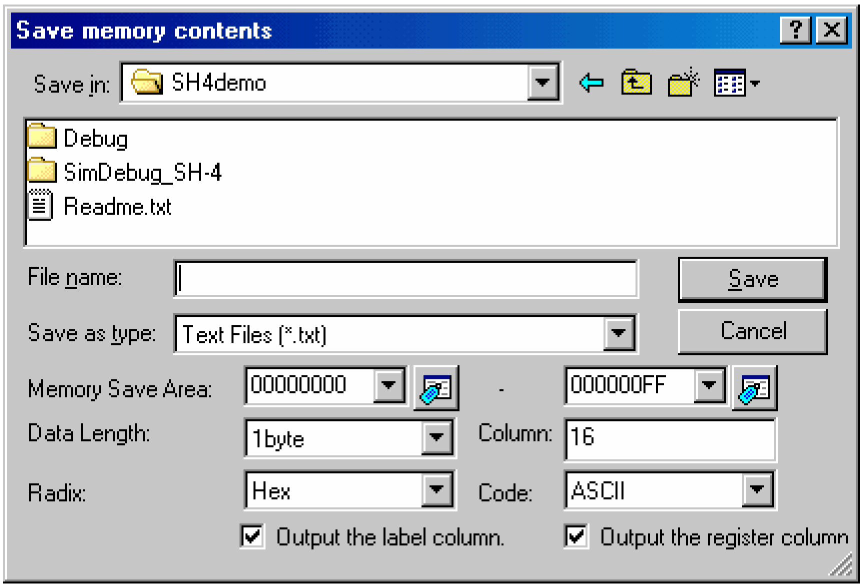
The exact test is target dependent. However, in all cases the current contents of the memory will be overwritten – **YOUR PROGRAM OR DATA WILL BE ERASED**.

17.3.8 Saving memory contents in a text file

You can save an area of memory in the address space to a text file using the **Save Memory Contents** feature. Select an address range to save in the Memory window by dragging the mouse. Choose the **Save Memory contents** option from the pop-up menu of the memory window. The **Save memory contents** dialog box is displayed.

Select the output range in **Memory Save Area**, data format in **Data Length**, number of digits in **Column**, radix in **Radix**, and code in **Code**. It is possible to select showing/hiding of the label display area and register display area by **Output the label column** and **Output the register column**, respectively.

If you did not drag the address range to be saved, you must enter the output range.



Save memory contents

Save in: SH4demo

Debug
SimDebug_SH-4
Readme.txt

File name:

Save as type: Text Files (*.txt)

Memory Save Area: 00000000 - 000000FF

Data Length: 1byte

Column: 16

Radix: Hex

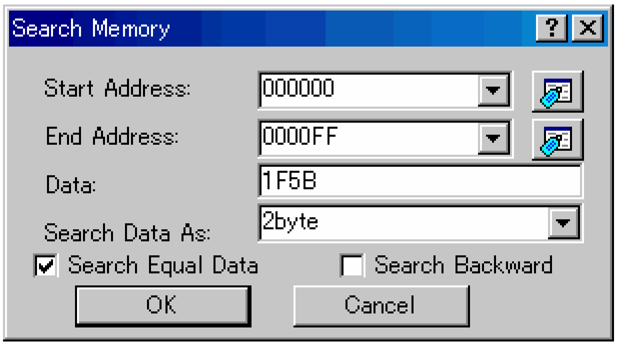
Code: ASCII

☑ Output the label column.

☑ Output the register column

17.3.9 Finding a value in memory

You can find a value in memory using the **Search Memory** feature. Select an address range to search in the Memory window by dragging the mouse. Choose the **Search** option from the pop-up menu of the memory window. The **Search memory** dialog box is displayed.



Search Memory

Start Address: 000000

End Address: 0000FF

Data: 1F5B

Search Data As: 2byte

☑ Search Equal Data

☐ Search Backward

Enter a value you want to find in **Data** and select the data format in **Search Data As**. Check **Search Equal Data** to find the exact value specified. To change the search direction, check **Search Backward**.

If "pattern" is selected as the **Search Data As**, a byte string of up to 256 bytes can be searched for.

The end address can also be prefixed by a plus sign (+), which will use the entered value as a range.

If the data could not be found, the Memory window display remains unchanged and a message box informing that the data could not be found is displayed.

If **Search Next** is selected from the pop-up menu in the state where data has been found, the search will continue from the next address.

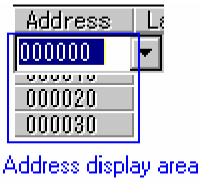
Support for this function depends on the debugging platform.

17.3.10 Changing the display address

Use the scroll bar, Page Up/Page Down key and Up/Down key to change the display position.

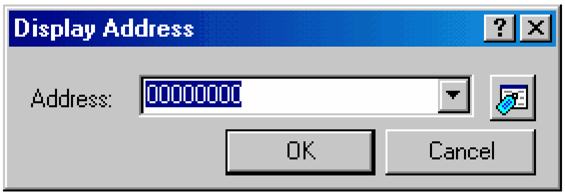
To change the display address directly, follow the procedure below.

1. In-place edit in the address display area.



2. To change the display address, open the **Display Address** dialog box by selecting one of the following operations:

- Double-click the address display area you want to change, **OR**
- Choose the **Address** option from the pop-up menu.


Display Address

Address: 00000000

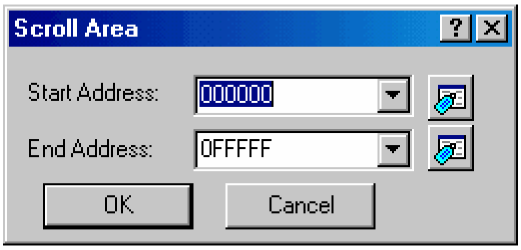
Specify the displaying address in the **Address** field.

Drag and Drop

Manipulation	Operation
Select address on Memory Window's Address area and drop it into another Memory Window's Address area	Changes the window's display start address to that address.
Select address (text string) on Disassembly Window's Address area and drop it into Memory Window's Address area	Changes the window's display start address to that address.
Select variable name (text string) and drop it into Memory Window's Address area	Changes the window's display start address to that address.

17.3.11 Changing the scroll area

Select the **Scroll Area** menu from the pop-up menu of the memory window. The **Scroll Area** dialog box is displayed.



Start Address: 000000

End Address: 0FFFFF

Specify the scroll range to be displayed. By default, the scroll range is set to 0 to the maximum address of MCU.

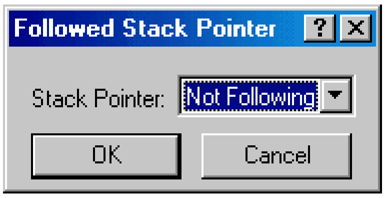
17.3.12 Starting address to value of the register

Select the register from the followings in the pop-up menu **Register** of the memory window.

Support for this function depends on the debugging platform.

17.3.13 Tracking the stack pointer position

The memory window has a function that alters the display address while tracking the stack pointer position (By default, the display does not track the stack pointer position.). To track the stack pointer position, choose **Followed Stack Pointer** option from the pop-up menu of the memory window. The **Followed Stack Pointer** dialog box is displayed.



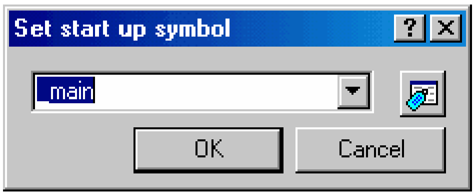
Stack Pointer: Not Following

Select the stack pointer to track from the drop-down list box. The Memory Window has its display addresses automatically altered by tracking the selected stack pointer position. Selecting "Not Following" in the **Followed Stack Pointer** dialog box causes the debugger to stop tracking the stack pointer position.

Support for this function depends on the debugging platform.

17.3.14 Changing the program display position immediately after downloading

To specify the source file position, select **Set Start Up Symbol** option from pop-up menu of the memory window. The **Set start up symbol** dialog box is displayed.



main

Input start up symbol to drop-down list box.

17.3.15 Updating the window contents

The Memory window contents can be forcibly updated. Selecting the **Refresh** from the pop-up menu of the memory window.

17.3.16 Disabling update of the window contents

Automatic update of the Memory window contents, which is performed when user program execution stops and in other cases, can be disabled. The Memory window will be grayed.

Selecting the **Lock Refresh** from the pop-up menu of the memory window.

17.3.17 Changing the data length

Select the data length from the followings in the pop-up menu **Data Length** of the memory window.

Either the following can be specified.

1byte	Display in 1-byte units (default).
2bytes	Display in 2-bytes units.
4bytes	Display in 4-bytes units.
8bytes	Display in 8-bytes units.

17.3.18 Changing the radix

Select the data radix from the followings in the pop-up menu **Radix** of the memory window.

Either the following can be specified.

Hex	Display in hexadecimal (default).
Dec	Display in decimal.
Signed Dec	Display in signed decimal.
Oct	Display in octal.
Bin	Display in binary.

17.3.19 Changing the code

Select the code from the followings in the pop-up menu **Code** of the memory window.

Either the following can be specified.

ASCII	Displaying memory as ASCII characters (default).
SJIS/JIS/UNICODE/EUC	Not support.
Float	Displaying memory as single-precision floating point.
Double	Displaying memory as double-precision floating point.
16bit Fixed	Displaying memory as 16bit fixed.
32bit Fixed	Displaying memory as 32bit fixed.
24bit Accum	Displaying memory as 24bit accumulate.
40bit Accum	Displaying memory as 40bit accumulate.

17.3.20 Setting the layout

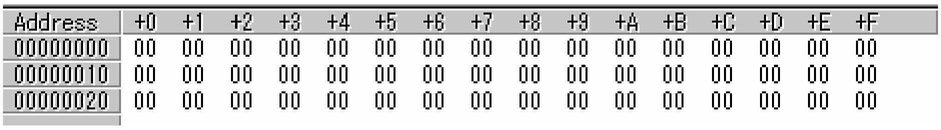
Select the layout from the followings in the pop-up menu **Layout** of the memory window.

The followings can be selected:

Label	Switch display or non-display of label.
Register	Switch display or non-display of Register.
Code	Switch display or non-display of Code.

When the label, register or code is shown, the option is checked.

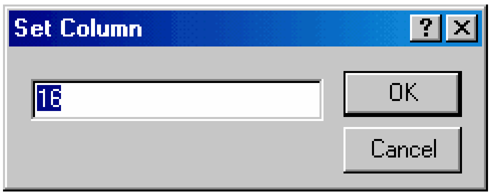
When the label, register and code are hidden:



```
Address   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
00000000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

17.3.21 Changing the number of digits displayed

Choose **Column** menu from the pop-up menu of the memory window. The **Set Column** dialog box is displayed.



16

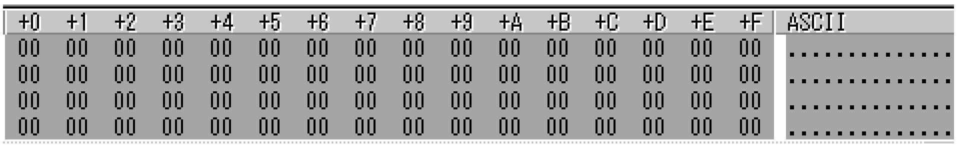
Specify the number of digits in which you want data to be displayed (1-256).

17.3.22 Switching display or non-display of measurement result

In the memory window, a display of coverage measurement is set to "Disable" by default. To enable the display, select **Coverage->Enable** menu from the pop-up menu of the memory window. In the Memory window, the background of the executed lines is displayed in sky blue, and the background of the unexecuted lines is displayed in gray.

During coverage measurement, the default foreground and background colors of the executed codes are black and light blue, respectively, while those of the codes not executed are black and gray. The colors in both cases can be customized in the **Format Views** dialog box.

For detail, see section 17.3.27, Customizing the current colors.



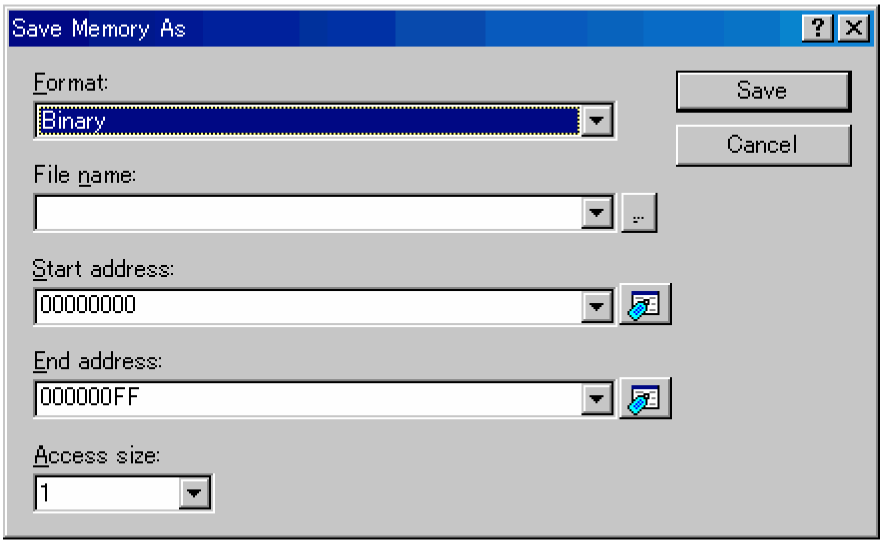
```
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  ASCII
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ..............
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ..............
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ..............
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ..............
```

Support for this function depends on the debugging platform.

17.3.23 Saving an area of memory

You can save an area of memory in the address space to a disk file using the **Save Memory** feature. Select an address range to save in the Memory window by dragging the mouse. Choose the **Save** option from the pop-up menu of the memory window. The **Save Memory As** dialog box is displayed

(This operation can also be achieved by selecting [**Debug->Save Memory**].)



Format: Binary

File name:

Start address: 00000000

End address: 000000FF

Access size: 1

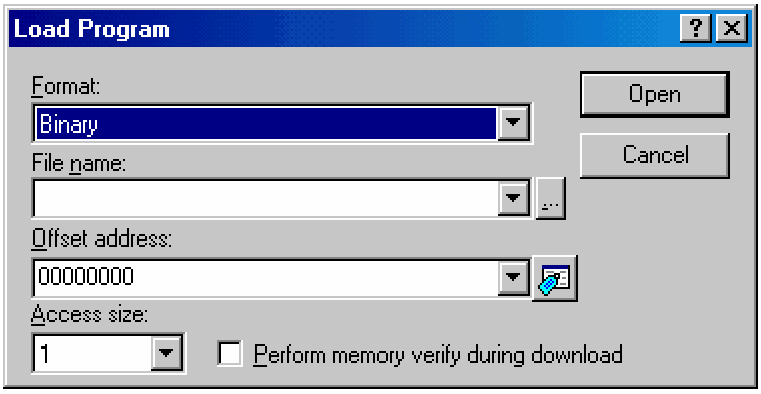
If you did not drag the address range to be saved, you must enter the start/end address. The end address can also be prefixed by a plus sign (+), which will use the entered value as a range.

Enter the file format in **Format**, file name in **File name**, and access size in **Access size**. The **File name** drop-down list contains the last four filenames used for saving memory. Alternatively, click the **Browse** button to launch a standard **Save As** dialog box. The access size for saving data can be selected from the **Access size** drop-down list. When the data is saved in memory with little endian, the order of data depends on the access size.

When the file save is complete a confirmation message box will be displayed.

17.3.24 Loading a memory area from a file

A file can be loaded to the debugging platform's memory. Choose **Load** from the pop-up menu of the memory window. The **Load Program** dialog box is displayed.



Format: Binary

File name:

Offset address: 00000000

Access size: 1

☐ Perform memory verify during download

Enter the file format in **Format**, file name in **File name**, offset address in **Offset address**, and access size in **Access size**. To verify memory, check **Perform memory verify during download**.

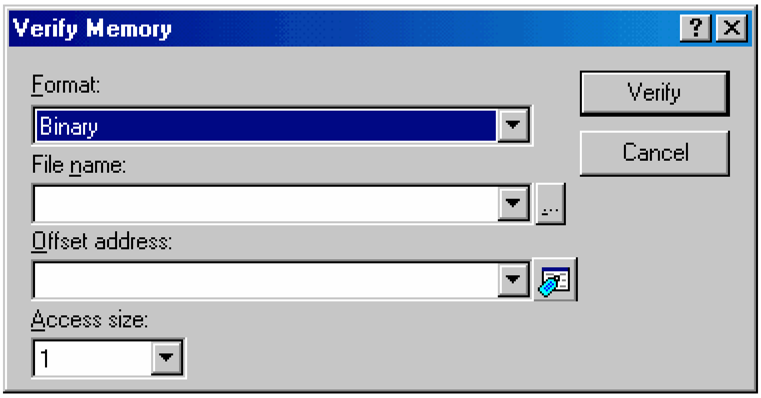
17.3.25 Splitting up the window display

To vertically divide the Memory window display into two, select **Split** from the pop-up menu and move the split-up bar.

To cancel the split-up display, select **Split** from the pop-up menu again.

17.3.26 Verifying a memory area

A memory area in the address space can be verified using the memory verify function. Choose the [**Debug->Verify Memory**] option. The **Verify Memory** dialog box is displayed.



Enter the file format in **Format**, file name in **File name**, offset address in **Offset address**, and access size in **Access size**.

The message “File verified OK” appears when verification is successful. If the verification failed, “Verify failed” appears.

Support for this function depends on the debugging platform.

17.3.27 Customizing the current colors

The color and font of the data or codes shown in the Memory window can be customized in the **Format Views** dialog box (in the same manner as the color and font in other windows). Now it is also possible to customize the color of the changed values or accessed memory, which are shown in the data or code field of the Memory window, in the **Format Views** dialog box.

To customize the current colors

1. Select the [**Setup->Format Views**] menu option. The **Format Views** dialog box will be displayed.
2. Select the **Memory** item from the tree in the left-hand section of the dialog box and expand it.
3. Select a category for which you wish to change the color. The following keyword groups are available:

Category	Foreground color in the Color tab (Default)	Background color in the Color tab (Default)
Normal	Black	White
Accessed ***1** (During coverage measurement: Executed codes)	Black	Sky blue
Not Accessed ***1** (During coverage measurement: Codes not executed)	Black	Gray
Unknown ***1**	Black	White
No Memory ***1**	Gray	Gray
Modified (Changed values)	Red	White

***1**. Support for this function depends on the debugging platform.

4. Change the selection in the **Foreground** and **Background** lists of the **Color** tab.
5. Click OK.

17.4 Displaying memory contents as an Image

The memory contents can be displayed as an image in the **Image** window.

17.4.1 Opening the Image window

Click the **Image** toolbar button (🖼) or choose [**View->Graphic->Image**] to open the **Image Properties** dialog box.

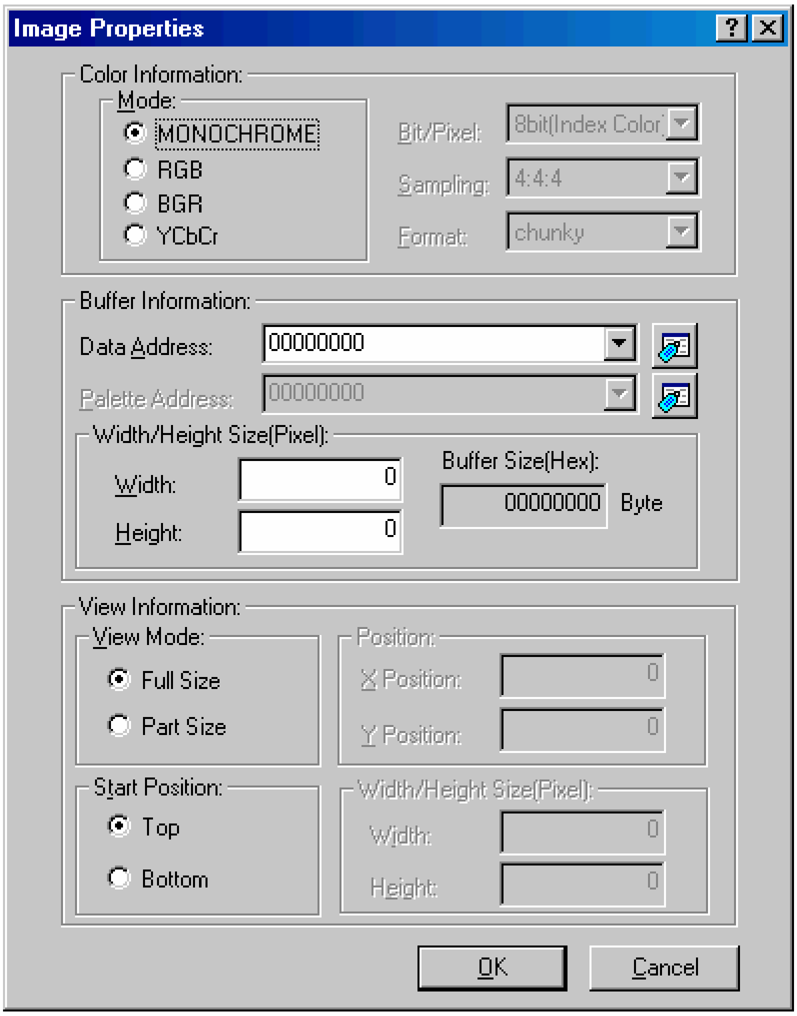


Image Properties

Color Information:

Mode:
● MONOCHROME
○ RGB
○ BGR
○ YCbCr

Bit/Pixel: 8bit(Index Color

Sampling: 4:4:4

Format: chunky

Buffer Information:

Data Address: 00000000

Palette Address: 00000000

Width/Height Size(Pixel):
Width: 0
Height: 0

Buffer Size(Hex): 00000000 Byte

View Information:

View Mode:
● Full Size
○ Part Size

Position:
X Position: 0
Y Position: 0

Start Position:
● Top
○ Bottom

Width/Height Size(Pixel):
Width: 0
Height: 0

The **Image Properties** dialog box is used to specify the display method of the **Image** window.

The following items are to be specified:

Color Information — Specifies the color information of the image to be displayed.

Mode — Specifies the format.

MONOCHROME Displays in black and white.

RGB Displayed in R (red), G (green), and B (blue)

BGR Displayed in B (blue), G (green), and R (red)

YCbCr Displayed by Y (brightness), Cb (color difference in blue), and Cr (color difference in red)

Bit/Pixel Specifies Bit/Pixel according to the selected **Mode**. (Valid when "RGB" or "BGR" is selected)

Sampling Specifies the format of sampling. (Valid when "YCbCr" is selected)
An example is shown below.

Sampling Ratio	**Original Data**	**Sampling Data**	**Size of an Image (4*4 pixels)**
4:4:4	11,12,13,14 21,22,23,24 31,32,33,34 41,42,43,44	11,12,13,14 21,22,23,24 31,32,33,34 41,42,43,44	16-bytes
4:2:2		11,11,13,13 21,21,23,23 31,31,33,33 41,41,43,43	8-bytes
4:1:1		11,11,13,13 11,11,13,13 31,31,33,33 31,31,33,33	4-bytes
4:2:0		11,11,11,11 21,21,21,21 31,31,31,31 41,41,41,41	4-bytes

Format Specifies Chunky/planar. (Valid when "YCbCr" is selected)

Format	**Order of Data**
Chunky (Invalid when 4:2:0 is selected)	Y,Cb,Cr,Y,Cb,Cr,Y,Cb,Cr,...,Y,Cb,Cr
planar	Y,Y,Y,...,Y,Cb,Cb,Cb,...,Cb,Cr.Cr.Cr,...,Cr
planar2	Y,Y,Y,...,Y,Cb,Cr,Cb,Cr,Cb,Cr,...,Cb,Cr

Buffer Information Specifies the area to store data, size, and the address of the palette.

Data Address Specifies the start address of the memory where image data is to be displayed. (Displayed in hexadecimal)

Palette Address Specifies the start address of the memory of color palette data. (Displayed in hexadecimal)
(Valid when "8Bit" is selected for "RGB" or "BGR")

Width/Height Size Specifies the width and height of the image.

Width (Pixel) Specifies the width of the image. (When a prefix is omitted, the values are input and displayed in decimal.)

Height (Pixel) Specifies the height of the image. (When a prefix is omitted, the values are input and displayed in decimal.)

Buffer Size (Hex) Displays the buffer size of the image from the width and height (Displayed in hexadecimal)

View Information Specifies the location, size, and data start location of the part to be displayed among the entire image.

View Mode Specifies the entire/part to be displayed in the image.

Full Size Displays the entire image.

Part Size Displays part of the image.

Start Position Specifies the start position of data to be displayed.

Top Displays data from the upper left.

Bottom Displays data from the lower left.

Position Specifies the start position of the image where part of the image is to be displayed. (Valid when "Part Size" is selected)

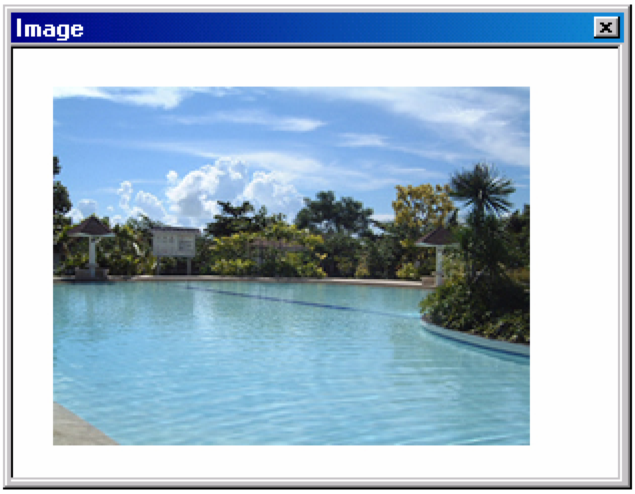
X Position Specifies the X axis of the start location. (When a prefix is omitted, the values are input and displayed

		in decimal.)
	Y Position	Specifies the Y axis of the start location. (When a prefix is omitted, the values are input and displayed in decimal.)
Width/Height Size	Specifies the height and width of the image to be displayed partly.	
	Width (Pixel)	Displays the width of display. (When a prefix is omitted, the values are input and displayed in decimal.)
	Height (Pixel)	Displays the height of display. (When a prefix is omitted, the values are input and displayed in decimal.)

After the settings have been made in the **Image Properties** dialog box, clicking the OK button opens the **Image** window.

Even after the **Image** window is displayed, the display contents can be modified by opening this dialog box by choosing **Properties** from the pop-up menu.

Displays the memory contents as an image.



- Double-clicking within the window displays information on the pixel on which the mouse pointer is located in the **Pixel Information** dialog box.

Option

Right-clicking displays a pop-up menu containing available options.

Pop-up menu options		Macro Recording	Function
Auto Refresh	Nonrefresh	-	Not refresh the window contents.
	Stop	-	Automatically update the window contents when user program execution stops.
	Real time	-	Real-time updated the window contents.
Refresh Now		-	Updates the window contents.
Properties		-	Opens the **Image Properties** dialog box.

17.4.2 Automatically updating the window contents

Checking **Auto Refresh->Nonrefresh** in the pop-up menu will not refresh the window.

Checking **Auto Refresh->Stop** in the pop-up menu will allow the window contents to be automatically updated when user program execution stops.

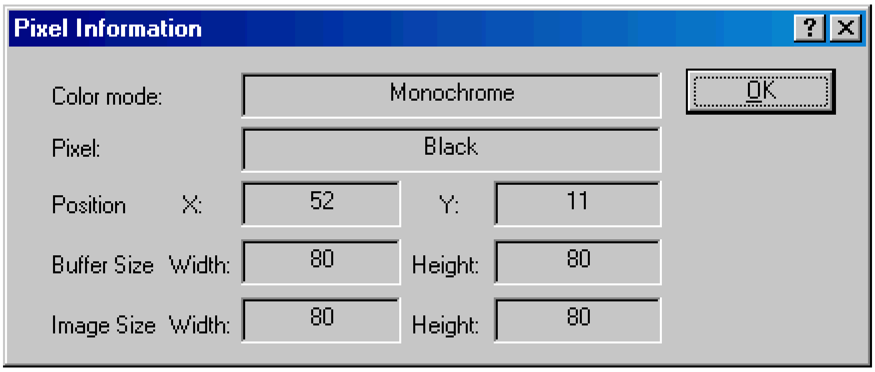
Checking **Auto Refresh->Real time** in the pop-up menu will allow the window contents to be real-time updated.

17.4.3 Updating the window contents

Selecting **Refresh Now** from the pop-up menu immediately updates the window contents.

17.4.4 Displaying the pixel information

Double-clicking within the window displays information on the pixel on which the mouse pointer is located in the **Pixel Information** dialog box.



Color mode: Monochrome

Pixel: Black

Position X: 52 Y: 11

Buffer Size Width: 80 Height: 80

Image Size Width: 80 Height: 80

This dialog box displays pixel information on the cursor location.

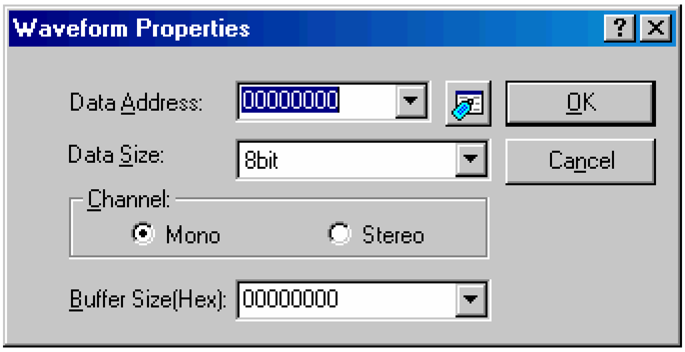
Color Mode Displays the format of the image.
Pixel Displays color information of the cursor location. (Displayed in decimal)
Position Displays the cursor location in X and Y axis. (Displayed in decimal)
X Displays the X axis of the cursor location.
Y Displays the Y axis of the cursor location.
Buffer Size Displays the buffer size. (Displayed in decimal)
Width Displays the buffer width.
Height Displays the buffer height.
Image Size Displays the width and height of the display. (Displayed in decimal)
Width Displays the width.
Height Displays the height.

17.5 Displaying memory contents as Waveforms

Memory contents can be displayed as wave forms in the **Waveform** window.

17.5.1 Opening the Waveform window

Choose [**View->Graphic->Waveform**] or click the **Waveform** toolbar button () to open the **Waveform Properties** dialog box.



Waveform Properties

Data Address: 00000000

Data Size: 8bit

Channel: ● Mono ○ Stereo

Buffer Size(Hex): 00000000

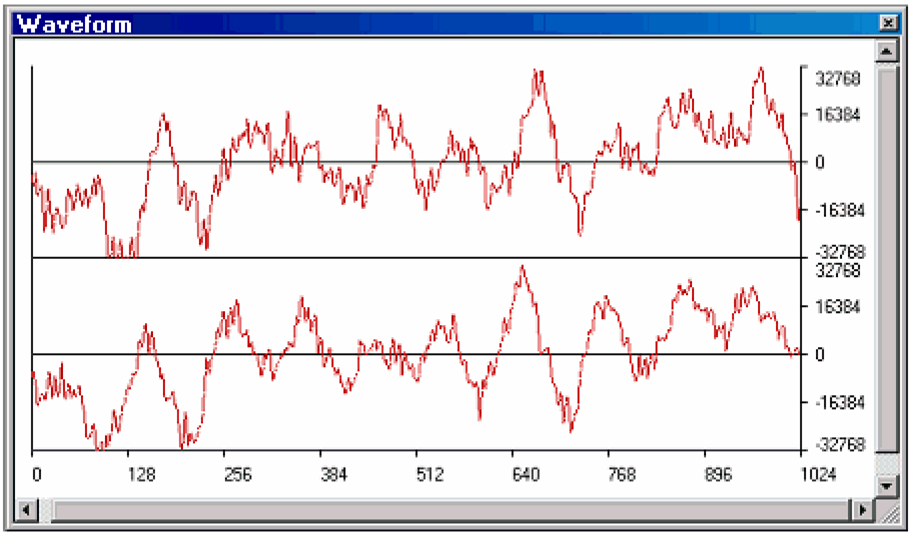
Specifies the waveform format. The following items can be specified.

Data Address	Specifies the start address of data in memory. (Displayed in hexadecimal)
Data Size	Selects 8bit or 16bit.
Channel	Specifies Mono or Stereo.
Buffer Size (Hex)	Specifies the buffer size of data. (Displayed in hexadecimal)

After the settings have been made in the **Waveform Properties** dialog box, clicking the OK button opens the **Waveform** window.

Even after the **Waveform** window is displayed, the display contents can be modified by opening this dialog box by choosing **Properties** from the pop-up menu.

Displays the memory contents as waveforms. The X axis shows the number of sampling data and the Y axis shows the sampling value.



- If you double-click the coordinate where you wish to view the sampling information, the **Sample Information** dialog box appears.

Option

Right-clicking displays a pop-up menu containing available options.

Pop-up menu options		Macro Recording	Function
Auto Refresh	Nonrefresh	-	Not refresh the window contents.
	Stop	-	Automatically update the window contents when user program execution stops.
	Real time	-	Real-time updated the window contents.
Refresh Now		-	Updates the window contents.

Zoom In		-	Zoom-in display.
Zoom Out		-	Zoom-out display.
Reset Zoom		-	Resets the zoom display.
Zoom Magnification	X2	-	The zoom magnification is 2.
	X4	-	The zoom magnification is 4.
	X8	-	The zoom magnification is 8.
Scale	128	-	The size of the X axis is 128 pixels.
	256	-	The size of the X axis is 256 pixels.
	512	-	The size of the X axis is 512 pixels.
Clear Cursor		-	Hides the cursor display.
Sample Information		-	Displays the sampling information of the cursor location.
Properties		-	Opens the **Waveform Properties** dialog box.

17.5.2 Automatically updating the window contents

Checking **Auto Refresh->Nonrefresh** in the pop-up menu will not refresh the window.

Checking **Auto Refresh->Stop** in the pop-up menu will allow the window contents to be automatically updated when user program execution stops.

Checking **Auto Refresh->Real time** in the pop-up menu will allow the window contents to be real-time updated.

17.5.3 Updating the window contents

Selecting **Refresh Now** from the pop-up menu immediately updates the window contents.

17.5.4 Zoom-in display

Selecting **Zoom In** from the pop-up menu displays the waveforms with the horizontal axis enlarged.

17.5.5 Zoom-out display

Selecting **Zoom Out** from the pop-up menu displays the waveforms with the horizontal axis reduced.

17.5.6 Resetting the zoom display

Selecting **Reset Zoom** from the pop-up menu displays the waveforms in its original size.

17.5.7 Setting the zoom magnification

In the **Zoom Magnification** submenu of the pop-up menu, the zoom magnification can be selected from 2, 4, or 8.

17.5.8 Setting the horizontal scale

In the **Scale** submenu of the pop-up menu, the size of the X axis can be selected from 128, 256, or 512 pixels.

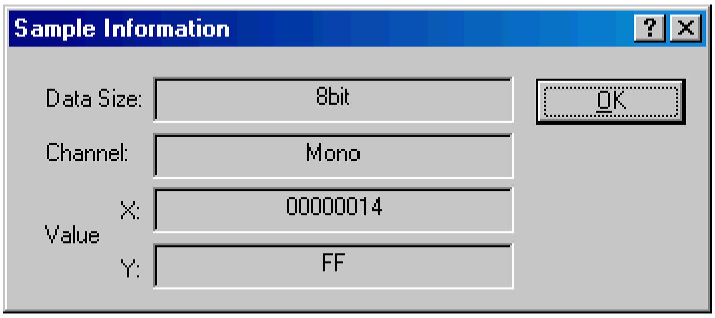
17.5.9 Non-display of cursor

Selecting **Clear Cursor** from the pop-up menu hides the cursor display.

17.5.10 Displaying the sampling information

When anywhere within the graph is clicked, a cursor (green vertical line) appears. The cursor can be moved by pressing the left or right arrow key. Right-click in the window to display a pop-up menu. Selecting **Sample Information** from the pop-up menu displays the **Sample Information** dialog box.

If you double-click the coordinate where you wish to view the sampling information, the **Sample Information** dialog box appears.



Sample Information

Data Size: 8bit

Channel: Mono

Value X: 00000014

Value Y: FF

Displays the sampling information of the cursor location in the **Waveform** window. The following information is displayed.

Data Size   Displays 8bit or 16bit.
Channel   Displays the data channel.
Value   X   Displays the X axis of cursor location.
    Y   Displays the Y axis of cursor location (displays Y axes for both the upper and lower plots when Stereo is selected).

17.6 Looking at I/O memory

As well as a CPU and ROM/RAM, a micro-controller also contains on-chip peripheral modules. The exact number and type of peripheral modules differ between devices but typical modules are DMA controllers, serial communications interfaces, A/D converters, integrated timer units, a bus state controller and a watchdog timer. Accessing registers, which are mapped to the micro-controller's address space, programs the on-chip peripherals.

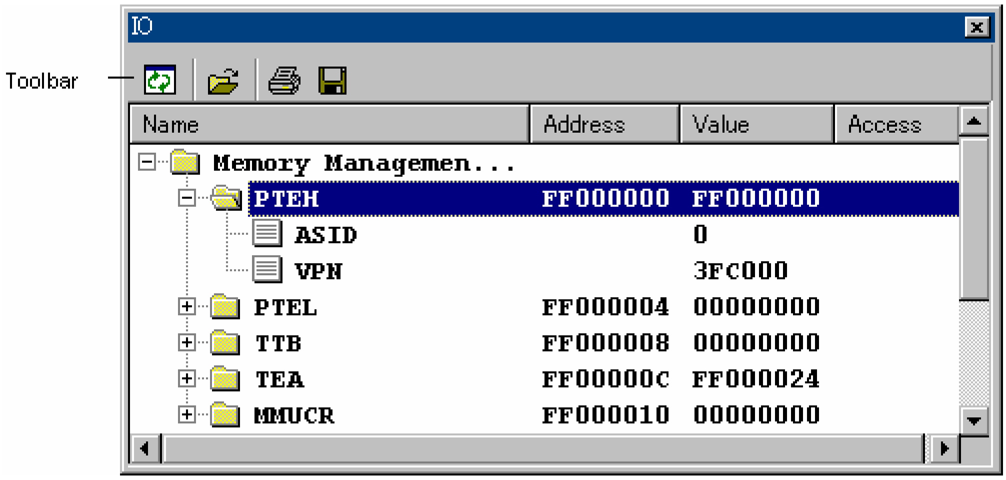
Since the setting up and use of these on-chip peripheral registers is usually very important in an embedded micro-controller application, it is useful to be able to look clearly at the contents of these registers. The **Memory** view only allows you to look at data in memory as byte, word, long word, single-precision floating-point, double-precision floating-point, or ASCII values, so the HEW also provides an **I/O Registers** view to ease the inspection and setting up of these registers.

To open an **I/O Registers** view, select the [**View->CPU->IO**] menu option, or click the View I/O toolbar button (I/O). Modules that match the on-chip peripherals organize the I/O register information. When an **I/O Registers** view is first opened, only a list of module names is displayed.

17.6.1 Opening the IO window

To open the **IO** window, select [**View->CPU->IO**] or click the **View IO** toolbar button (I/O). Modules that match the on-chip peripheral modules organize the I/O register information. When the **IO** window is first opened, only a list of module names is displayed.

Window configuration



Name	Address	Value	Access
⊟ Memory Managemen...			
⊟ PTEH	FF000000	FF000000	
ASID		0	
VPN		3FC000	
⊞ PTEL	FF000004	00000000	
⊞ TTB	FF000008	00000000	
⊞ TEA	FF00000C	FF000024	
⊞ MMUCR	FF000010	00000000	

- Expanding the I/O register/bit display.
- If the I/O register/bit value is changed, the value is displayed in red.
- Double-clicking the I/O register/bit display line opens a dialog, which allows you to change a register value. (Macro Recording)
- The I/O register/bit contents line can be changed by in-place editing. (Macro Recording)

Option

Right-clicking displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up menu options	Toolbar button	Macro Recording	Function
Refresh	↻	-	Updates the window contents.
Load IO File	📂	-	Manually load an IO file.
Print	🖨	-	Prints the contents currently displayed in the window.
Save To File	💾	-	Saves the contents currently displayed in the window in a text file.
Toolbar display	-	-	Shows or hides the toolbar.
Customize toolbar	-	-	Customizes toolbar buttons.

17.6.2 Expanding an I/O register display

To display the names, addresses and values of the I/O registers, double-click on the module name or select the module name, by clicking on it or using the cursor keys, and press the cursor right key. The module display will expand to show the individual registers of that peripheral module and their names, addresses and values. Double-clicking (or pressing the cursor left key) again on the module name will close the I/O register display.

Note:

If you are using an emulator-based debugging platform, reading data from an I/O register can sometimes affect the operation of your program. For example, reading a data register can cancel a pending interrupt. Data is only read from I/O modules that have been expanded in the **IO** window (so that the register values are displayed). Therefore, as long as I/O modules are collapsed when they no longer need to be displayed, this will not cause a problem. Also, note that having a Memory window (or Disassembly window) open on the I/O area can have the same effect.

17.6.3 Updating the window contents

Selecting **Refresh** from the pop-up menu immediately updates the window contents.

17.6.4 Loading an I/O file

I/O files can be manually loaded to the IO window.

To load an I/O file

1. Right-click in the IO window to display a pop-up menu.
2. Select **Load IO File**. The **Set I/O File** dialog box appears.
3. Specify an I/O file you wish to load in the **I/O File** field. To insert a placeholder in the **I/O File** field, click the Placeholder button (▶) and select a placeholder from the pop-up menu. To browse a file, click the **Browse** button.
4. If the **Save path name of the I/O file** checkbox is not selected, the path name in the **I/O File** field will not be saved when you save the session. Next time this session is selected, the contents to be loaded to the IO window depend on the debugging platform. When the I/O file has been automatically downloaded, ensure that this checkbox is not selected. The I/O file dependent on the selected CPU is chosen by the debugging platform and automatically loaded. If the **Save path name of the I/O file** checkbox is selected, the path name in the **I/O File** field will be saved when you save the session. Next time this session is selected, the most recently loaded I/O file will be loaded to the IO window.
5. Click OK.

See Reference 6, I/O File Format, for more information about a I/O file format.

17.6.5 Printing the currently displayed contents

The contents currently displayed in the window can be printed in a text file. Select **Print** from the pop-up menu.

17.6.6 Saving the currently displayed contents

The contents currently displayed in the window can be saved in a text file. Select **Save to File** from the pop-up menu.

17.6.7 Modifying I/O register contents

To edit the value in an I/O register you can type hex values directly into the view. To enter more complex expressions, double-click or press Enter on the register to open a dialog box to modify the register contents. When you have entered the new number or expression, click the OK button or press Enter. The dialog box closes and the new value is written into the register.

17.7 Looking at registers

If you are debugging at assembly-language level using the **Disassembly** view, you will probably find it useful to see the contents of the CPU's general registers. You can do this by using the **Register** window.

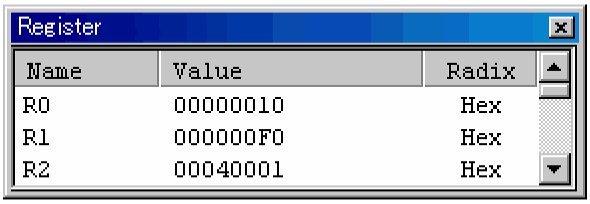
17.7.1 Opening the Register window

The Register window displays the register data and flag data. You can change a register/flag value from the window.

To open the **Register** window, click the **Registers** toolbar button (R1), or select the [**View->CPU->Registers**] menu option. The **Register** window opens showing all of the CPU's general registers and values (displayed in hexadecimal).

Window configuration

Example of SH-4 debugger:



Register

Name	Value	Radix
R0	00000010	Hex
R1	000000F0	Hex
R2	00040001	Hex

- Double-clicking the register display line opens a dialog box, which allows you to change a register value. (Macro Recording)
- You can change a flag value by clicking the button corresponding to the flag. (Macro Recording)
- The right-click menu allows you to change the display radix point and the register bank (Change of the register bank can be selected only when the target platform supports this function.).

Option

Right-clicking displays a pop-up menu containing available options.

Pop-up menu options		Macro Recording	Function
Radix	Hex	-	Display in hexadecimal.
	Dec	-	Display in decimal.
	Oct	-	Display in octal.
	Bin	-	Display in binary.
Bank0 *		-	Display registers of bank 0.
Bank1 *		-	Display registers of bank 1.
Layout	Radix	-	Switches display or non-display of radix.
	FLAGs	-	Switches display or non-display of flags display area.
	Settings	-	Chooses a register to be displayed.
Edit		●	Changes a register's contents.
Split		-	Splits up the window Display.
Save To File		-	Saves register contents in a text file.

***.** Support for this function depends on the debugging platform.

17.7.2 Changing the register display radix

You can change the display radix by register.

To do this, click the mouse right button on the register to be changed and select the display radix from the pop-up menu which is opened.

The followings can be selected

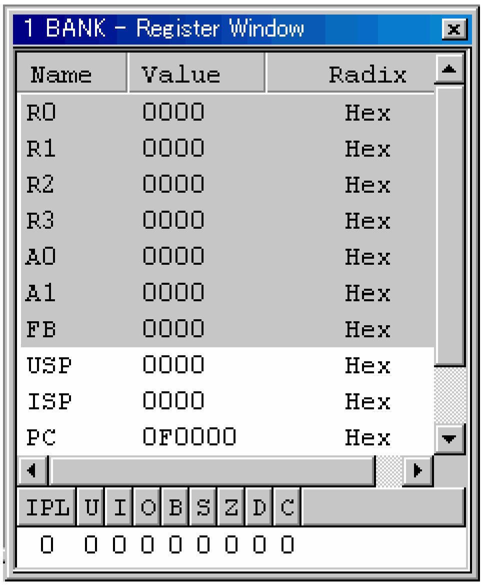
Hex	Display in hexadecimal.
Dec	Display in decimal.
Oct	Display in octal.
Bin	Display in binary.

17.7.3 Switching Register Bank

Immediately after opening the Register window, the register data for the bank corresponding to the value of flag is displayed.

To reference the register data of Bank1, select **Bank1** menu option with the Register window active.

The register specific to Bank1 is displayed in the gray background. (Example of M16C family debugger)



Name	Value	Radix
R0	0000	Hex
R1	0000	Hex
R2	0000	Hex
R3	0000	Hex
A0	0000	Hex
A1	0000	Hex
FB	0000	Hex
USP	0000	Hex
ISP	0000	Hex
PC	0F0000	Hex

IPL	U	I	O	B	S	Z	D	C
0	0	0	0	0	0	0	0	0

To reference the register data of Bank0, select **Bank0** menu option with the Register window active. (Through the operation of option **Bank0** and **Bank1**, the value of flag does not change.)

To switch the bank, you can also use the pop up menu which is displayed by clicking the mouse right button on the register display area in the Register window, or change the value of flag.

(If you change the value of flag, the register bank also changes in response to the value.)

Support for this function depends on the debugging platform.

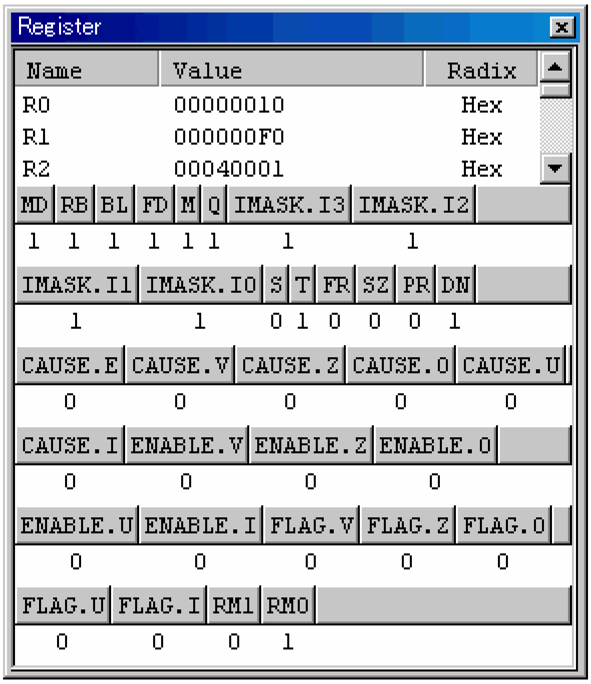
17.7.4 Setting the layout

To set the layout of the Register Window, choose **Layout** menu from the register Window pop-up menu. The followings can be selected:

Radix Switch display or non-display of radix.
FLAGs Switch display or non-display of flags display area.

When the radix or flag is shown, the option is checked.

When the radix and flags are displayed



Name	Value	Radix
R0	00000010	Hex
R1	000000F0	Hex
R2	00040001	Hex

MD	RB	BL	FD	M	Q	IMASK.I3	IMASK.I2
1	1	1	1	1	1	1	1

IMASK.I1	IMASK.I0	S	T	FR	SZ	PR	DN
1	1	0	1	0	0	0	1

CAUSE.E	CAUSE.V	CAUSE.Z	CAUSE.O	CAUSE.U
0	0	0	0	0

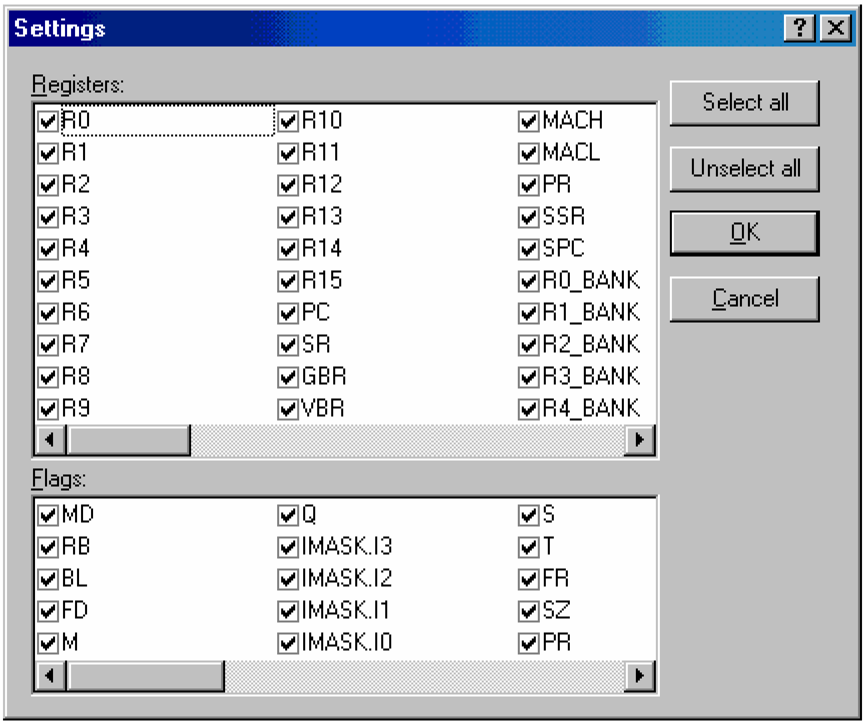
CAUSE.I	ENABLE.V	ENABLE.Z	ENABLE.O
0	0	0	0

ENABLE.U	ENABLE.I	FLAG.V	FLAG.Z	FLAG.O
0	0	0	0	0

FLAG.U	FLAG.I	RM1	RM0
0	0	0	1

17.7.5 Choosing a register to be displayed

To choose a register to be displayed in the Register window, choose **Settings** menu from the register pop-up menu. This dialog box is shown in following figure.



Registers:

☑ R0
☑ R1
☑ R2
☑ R3
☑ R4
☑ R5
☑ R6
☑ R7
☑ R8
☑ R9
☑ R10
☑ R11
☑ R12
☑ R13
☑ R14
☑ R15
☑ PC
☑ SR
☑ GBR
☑ VBR
☑ MACH
☑ MACL
☑ PR
☑ SSR
☑ SPC
☑ R0_BANK
☑ R1_BANK
☑ R2_BANK
☑ R3_BANK
☑ R4_BANK

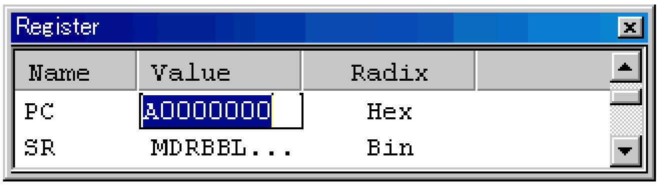
Flags:

☑ MD
☑ RB
☑ BL
☑ FD
☑ M
☑ Q
☑ IMASK.I3
☑ IMASK.I2
☑ IMASK.I1
☑ IMASK.I0
☑ S
☑ T
☑ FR
☑ SZ
☑ PR

17.7.6 Modifying register contents

To change register contents, follow the procedure below.

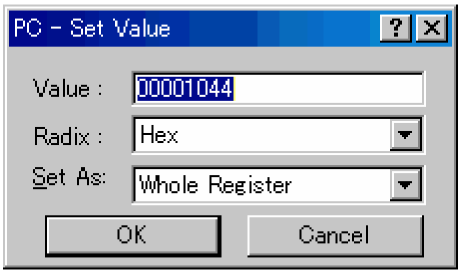
Enter a value in the **Value** field of the register you want to change.



Name	Value	Radix
PC	A0000000	Hex
SR	MDRBBL...	Bin

To change a register's contents, open the Set Value dialog by selecting one of the following operations

- Double-click the register you want to change, **OR**
- Select the register you want to change and choose the **Edit** option from the pop-up menu.



Value : 00001044

Radix : Hex

Set As: Whole Register

You can enter a number or C/C++ expression in the **Value** field.

You can choose the radix from the **Radix** drop-down list box.

You can choose whether to modify the entire contents of the register, a masked area, floating bits or flag bits, by selecting an option from the **Set As** drop-down list box (the contents of this list depends on the CPU model and selected register).

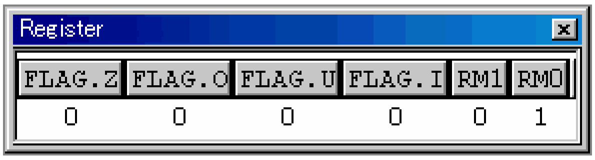
When you have entered the new number or expression, click the **OK** button or press Enter. The dialog box closes and the new value is written into the register.

17.7.7 Setting the flag value

When the flag itself is displayed

Click the button of the flag to be changed. Every time you click the button the flag status (1/0) is switched. If a flag is composed of multiple bits, a dialog is opened, where you can enter a value to be changed.

Example of SH-4 debugger

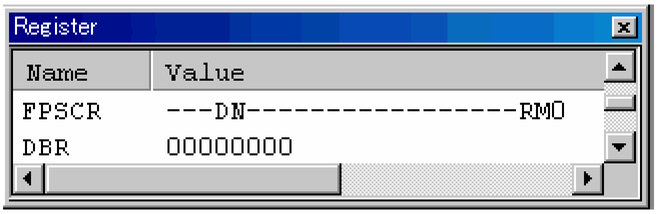


FLAG.Z	FLAG.O	FLAG.U	FLAG.I	RM1	RM0
0	0	0	0	0	1

When the flag is displayed in the register

Double-click the FLG line. A dialog is opened. Enter the value to be changed.

Example of SH-4 debugger (FLG line (FPSCR))



Name	Value
FPSCR	---DN----------------RM0
DBR	00000000

17.7.8 Splitting up the window display

To vertically divide the Register window display into two, select **Split** from the pop-up menu and move the split-up bar.

To cancel the split-up display, select **Split** from the pop-up menu again.

17.7.9 Saving register contents

To save register contents in a text file, choose **Save To File** menu from the register pop-up menu. The **Save As** dialog box shown below will be displayed. Specify the file name.

17.7.10 Using register contents

It can be useful to be able to use the value contained in a CPU register when you are entering a value elsewhere in the HEW, for example when displaying a specified address in the **Disassembly** or **Memory** views. You can do this by specifying the register name prefixed by the '#' (when the SH, H8, or R8C (E7/E8) family is in use) or '%' (when the M32C, M32R, M16C, or R8C (excluding E7/E8) family is in use) character, e.g.: '#PC' or '%PC'.

Support for this function depends on the debugging platform.

17.7.11 Customizing the color of the changed values

The text color and font of the register or flag values shown in the **Register** window can be customized in the **Format Views** dialog box (in the same manner as the color and font in other windows).

Now it is also possible to customize the color of the changed register or flag values in the **Format Views** dialog box. The default foreground and background colors of the changed values are red and white, respectively. The color of the background will be the same as in cases where the text category is selected.

To customize the color of the changed values

1. Select the [**Setup->Format Views**] menu option. The **Format Views** dialog box will be invoked.
2. Select the **Register** item from the tree in the left-hand section of the dialog box and expand it.
3. Select the **Modified** category.
4. Change the selection in the **Foreground** list of the **Color** tab.
5. Click OK.

17.8 Resetting the target MCU

To reset the target MCU, click the **Reset CPU** toolbar button (), or select the [**Debug->Reset CPU**] menu option.

Resetting the target MCU initializes the on-chip I/O registers and makes the program counter jump to the address set in the reset vector.

17.9 Setting PC to the address at cursor

To changes the value of the PC to the address at the row of the text cursor, click the **Set PC to cursor** toolbar button ( ), or select the [**Debug->Set PC to Cursor**] menu option.

17.10 Initializing the debugging platform

Select the [**Debug->Initialize**] menu option.

It will close down any open child windows and shut down the link to the debugging platform. If this is successful, an attempt to re-establish the link to the debugging platform will be made.

17.11 Connecting/disconnecting the debugging platform

To connect the debugging platform

Select one of the following operations:

- Click the **Connect** toolbar button ( ), **OR**
- Select the [**Debug->Connect**] menu option.

To disconnect the debugging platform

Select one of the following operations:

- Click the **Disconnect** toolbar button ( ), **OR**
- Select the [**Debug->Disconnect**] menu option.

Support for this function depends on the debugging platform.

17.12 Executing your program

This section describes how you can execute your program's code. You will learn how to do this by either running your program continuously or stepping single or multiple instructions at a time.

17.12.1 Continuing run

When your program is stopped and the debugger is in **Break Mode**, the HEW will display a yellow arrow (⇨) in the gutter of the line in the source and disassembly views that correspond to the CPU's current Program Counter (PC) address value. This will be the next instruction to be executed if you perform a step or continue running.

To continue running from the current PC address

- Click the **Go** toolbar button ( ), **OR**
- Choose the [**Debug->Go**] menu option.

17.12.2 Running from reset

To reset your user system and run your program from the Reset Vector address, choose the [**Debug->Reset Go**] menu option or click the **Reset Go** toolbar button (). The program will run until it hits a breakpoint or a break condition is met. You can stop the program manually at any time by choosing the [**Debug->Halt Program**] menu option or by clicking the **Halt** toolbar button (STOP).

Note:

The program will start running from whatever address is stored in the Reset Vector location. Therefore it is important to make sure that this location contains the address of your startup code.

17.12.3 Running to cursor

Sometimes as you are going through your application you may only want to run a small section of code, that would require many single steps to execute. In this case it would be useful to be able to run to a particular point. You can do this by using the **Go To Cursor** feature.

How to use the Go To Cursor feature

1. Make sure that an editor or disassembly view is open, showing the address at which you wish to stop.
2. Position the cursor on the line containing the address at which you wish to stop.
3. Choose the **Go To Cursor** option from the pop-up menu.

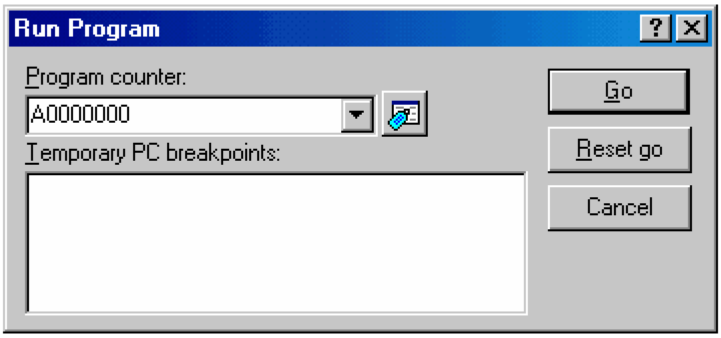
The debugging platform will run your code from the current PC value until it reaches the address indicated by the cursor's position.

Notes:

- If your program never executes the code at this address, the program will not stop. If this happens, you can stop code execution by pressing ESC, choosing the [**Debug->Halt Program**] menu option, or clicking the **Halt** toolbar button (STOP).
- The Go To Cursor feature requires a temporary breakpoint – if you have already used all those available then the feature will not work, and the menu option will be disabled.

17.12.4 Running from a specified address

The **Run Program** dialog box allows the user to run the program from any address. Choose [**Debug->Run**] to open the **Run Program** dialog box.



Run Program

Program counter: A0000000

Temporary PC breakpoints:

The following execution conditions can be specified in this dialog box:

Program counter — Instruction address to start execution. The initial value is the current PC value.

Temporary PC Breakpoints — A temporary PC breakpoint. When execution started by this dialog box stops, this breakpoint is cleared.

Clicking the **Go** button starts execution according to the settings. Clicking the **Reset Go** button starts execution from the reset vector. Clicking the Cancel button closes this dialog box without executing instructions.

Note:

The Temporary PC Breakpoints feature requires a PC breakpoint - if you have already used all those available then the feature will not work.

17.12.5 Continuing execution to a main function at a reset

The HEW has a function to continue execution of the program at a reset until it reaches the top of a main function. Then the HEW sets a temporary software breakpoint at the main function. When execution of the program is stopped, this breakpoint is deleted.

To continue execution of the program to the top of a main function at a reset

1. Select [**Setup->Options**] to open the **Options** dialog box.
2. Select the **Debug** tab.
3. Select the **Run to main function on reset** checkbox. By default, this box is not selected.
4. Reset the target microcomputer. Select one of the following operations:

- Click **Reset CPU** toolbar button ( ), **OR**
- Select [**Debug->Reset CPU**].

Note:

The Temporary PC Breakpoints feature requires a PC breakpoint - if you have already used all those available then the feature will not work.

17.12.6 Single step

When you are debugging your code it is very useful to be able to step a single line or instruction at a time and examine the effect of that instruction on the system. In the source view, a step operation will step a single source line. In the disassembly view, a step operation will step a single assembly-language instruction. If the instruction calls another function or subroutine, you have the option to either step into or step over the function. If the instruction does not perform a call, then either option will cause the debugger to execute the instruction and stop at the next instruction.

If you choose to step into the function, the debugger will execute the call and stop at the first line or instruction of the function.

To step into the function

- Click the **Step In** toolbar button ( ), **OR**
- Choose the [**Debug->Step In**] menu option.

If you choose to step over the function the debugger will execute the call and all of the code in the function (and any function calls that function may make) and stop at the next line or instruction of the calling function.

To step over the function

- Click the **Step Over** toolbar button ({}), **OR**
- Choose the [**Debug->Step Over**] menu option.

During debugging, there are occasions when you may have entered a function, finished stepping through the instructions that you want to examine and would like to return to the calling function without tediously stepping through all the remaining code in the function. Alternatively you may have stepped into a function by accident, when you meant to step over it and so want to return to the calling function without stepping all the way through the current function. You can do this with the **Step Out** feature.

To step out of the current function

- Click the **Step Out** toolbar button ({}), **OR**
- Choose the [**Debug->Step Out**] menu option.

To choose the step mode to use while stepping

Select the [**Debug->Step Mode**] menu option.

Sub-menu	Function
Auto (default)	Automatically chooses the step mode
Assembly	Steps through assembly instructions
Source	Steps through source code

While performing **Step In** in source mode within the HEW editor, you may wish to keep the source mode display of the library functions (e.g. printf) containing no debugging information. By default, the execution jumps to the **Disassembly** window and steps every assembly-language instruction.

You can select not to actually step in addresses where no debugging information exists even when the **Step In** execution reaches such addresses.

To select not to step into addresses where no debugging information exists

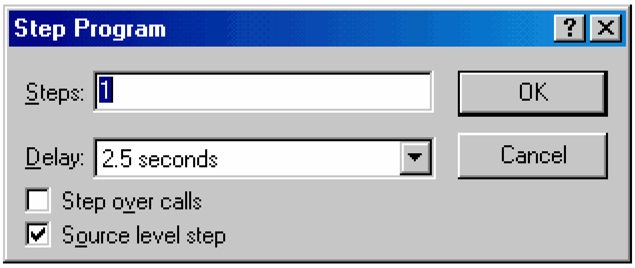
1. Select the [**Setup->Options**] menu option. The **Options** dialog box will be displayed.
2. Select the **Debug** tab.
3. Select the **Only step in when debug information is available** checkbox. By default, this checkbox is not selected.

Note:

Support for this function depends on the debugging platform.

17.12.7 Multiple steps

Sometimes you may find it useful to step through several instructions at a time. You can do this by using the **Step Program** dialog box. The dialog box also provides an automated step with a selectable delay between steps. Open it by choosing the [**Debug->Step**] menu option. The **Step Program** dialog box is displayed.



Steps: 1

Delay: 2.5 seconds

☐ Step over calls

☑ Source level step

Steps Number of steps to be executed. By default, this value is 1.

Delay Delay between steps when the program is automatically stepped. "No Refresh" (Prevents the update of the windows) or 0 to 3 seconds can be selected in 0.5 second units. By default, this value is 2.5 seconds.

Step over calls Selecting this box steps over function calls. By default, this checkbox is not selected.

Source level step Selecting this box steps the program at the source level. By default, this checkbox is selected.

Click the OK button or press Enter to start stepping.

17.13 Stopping your program

This section describes how you can halt execution of your application's code. This section describes how to do this directly by using the **Halt** toolbar button (STOP) and by setting breakpoints at specific locations in your code.

17.13.1 Halting execution

When your program is running, the Halt Program toolbar button is enabled (STOP), and when the program has stopped it is disabled (STOP).

To stop the program

- Click on the **Halt** toolbar button (STOP), **OR**
- Choose the [**Debug->Halt Program**] menu option.

Your program's execution is halted, with the message ‘Break = User Break’ displayed on the **Status Bar**. The HEW will then update any open views. The cause of the last break can also be viewed in the **Platform** tab of the **Output** view.

17.13.2 Standard PC breakpoints

When you are trying to debug your program, you will want to be able to stop the program running when it reaches specific points in your code. You can do this by setting a program counter (PC) breakpoint on the line or instruction at which to want the execution to stop. The following instructions will show you how to quickly set and clear simple PC breakpoints. If more complex breakpoint operation is supported by the target, it may provide a **Breakpoints** view.

To set a program counter (PC) breakpoint

1. Make sure that the source or disassembly view is open at the place at which you want to set a PC breakpoint.
2. Select the **Toggle Breakpoint** pop-up menu item, or press F9, at the line showing the address at which you want the program to stop.
3. You will see a red circle (●) appear in the gutter to indicate that a PC breakpoint has been set.
4. It is also possible to add, remove and edit the current breakpoint setup by selecting the [**Edit->Source Breakpoints**] menu option.
5. Now when you run your program and it reaches the address at which you set the PC breakpoint, execution halts with the message 'PC Breakpoint' displayed in the **Debug** tab, and the source or disassembly view is updated with the PC breakpoint line highlighted.

Note:

The line or instruction at which you set a PC breakpoint is not actually executed – the program stops just before it is about to execute it. If you choose to **Go** or **Step** after stopping at the PC breakpoint, then the highlighted line will be the next instruction to be executed. When multiple targets are debugged, it is possible to specify either or both of them is stopped. For details, see section 17.17, Synchronizing multiple debugging platforms.

To change the PC breakpoint setting by using the Breakpoints dialog box

The breakpoint dialog can be displayed by selecting the [**Edit->Source Breakpoints**] menu option. It allows you to view the current breakpoints set in the workspace and view the code associated with each one. From this dialog it is also possible to remove one or all breakpoints.

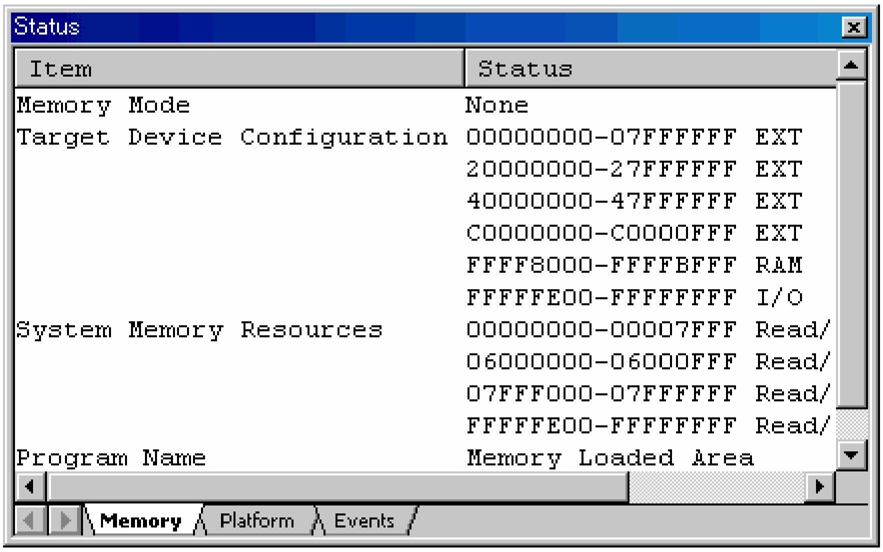
To toggle PC breakpoints

It is possible to toggle PC Breakpoints either by double-clicking in the breakpoint (BP) column of the line at which the PC breakpoint is set, or by placing the caret on the line and using the F9 key. The display will cycle through the available standard breakpoint types – a red circle (●) will be shown in the gutter.

17.14 Viewing the current status

To check the configuration and status of the debugging platform in the **Status** window, click the **View Status** button (), or choose the [**View->Status**] menu item.

Window configuration



Item	Status
Memory Mode	None
Target Device Configuration	00000000-07FFFFFF EXT
	20000000-27FFFFFF EXT
	40000000-47FFFFFF EXT
	C0000000-C0000FFF EXT
	FFFF8000-FFFFBFFF RAM
	FFFFFE00-FFFFFFFF I/O
System Memory Resources	00000000-00007FFF Read/
	06000000-06000FFF Read/
	07FFF000-07FFFFFF Read/
	FFFFFE00-FFFFFFFF Read/
Program Name	Memory Loaded Area

The **Status** panel has three sheets:

Platform

Contains information about the current status of the debugging platform, typically including CPU type and mode, run status and timing information.

Memory

Contains information about the current memory status including the memory mapping resources and the areas used by the currently loaded object file.

Events

Contains information about the current event (breakpoint) status, including resource information.

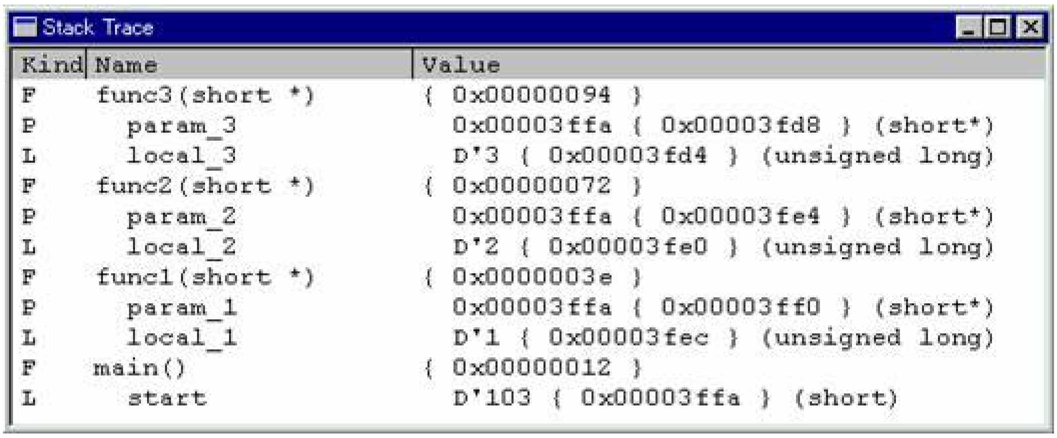
17.15 Viewing the function call history

The **Stack Trace** window shows the function call history.

17.15.1 Opening the Stack Trace window

To open the **Stack Trace** window, choose [**View->Code->Stack Trace**] or click the **Stack Trace** toolbar button ().

Window configuration



```
Kind Name              Value
F    func3(short *)    { 0x00000094 }
P      param_3           0x00003ffa { 0x00003fd8 } (short*)
L      local_3           D'3 { 0x00003fd4 } (unsigned long)
F    func2(short *)    { 0x00000072 }
P      param_2           0x00003ffa { 0x00003fe4 } (short*)
L      local_2           D'2 { 0x00003fe0 } (unsigned long)
F    func1(short *)    { 0x0000003e }
P      param_1           0x00003ffa { 0x00003ff0 } (short*)
L      local_1           D'1 { 0x00003fec } (unsigned long)
F    main()            { 0x00000012 }
L      start             D'103 { 0x00003ffa } (short)
```

The following items are displayed.

Kind	Indicates the type of the symbol. F: Function P: Function parameter * L: Local variable *
Name	Indicates the symbol name.
Value	Indicates the value, address, and type of the symbol.

***.** Support for this function depends on the debugging platform.

Option

Right-clicking displays a pop-up menu containing available options.

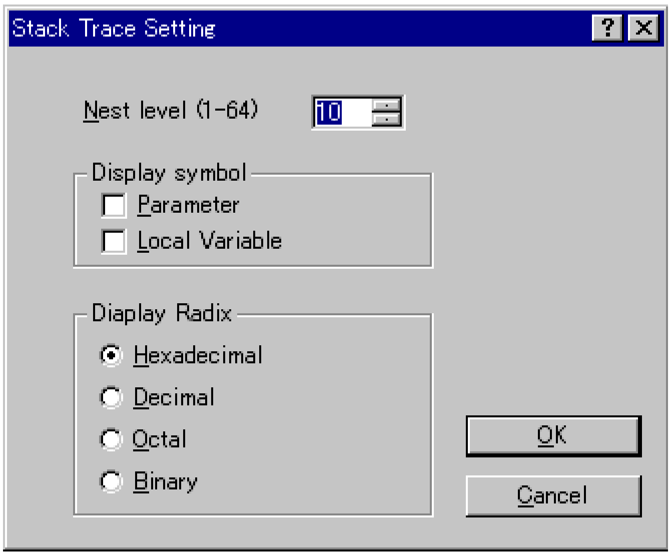
Pop-up menu options	Macro Recording	Function
Go to Source	-	Go to the associated source line.
View Setting	-	Specifying the **Stack Trace** window settings.
Copy	-	Places a copy of the highlighted text into the Windows® clipboard.

17.15.2 Viewing the source program

Select a function and choose **Go to Source** from the pop-up menu to display, then the source program corresponding to the function, which has been selected by opening the editor window, is displayed.

17.15.3 Specifying the view

Choose **View Setting** from the pop-up menu to open the **Stack Trace Setting** dialog box, which allows the user to specify the Stack Trace window settings.



Stack Trace Setting

Nest level (1-64) 10

Display symbol
□ Parameter
□ Local Variable

Diaplay Radix
● Hexadecimal
○ Decimal
○ Octal
○ Binary

Nest level	Specifies the level of function call nesting to be displayed in the Stack Trace window.
Display symbol	Specifies the symbol types to be displayed in addition to functions.
Display Radix	Specifies the radix for displays in the Stack Trace window.

17.16 Using an external debugger

The High-performance Embedded Workshop can launch an external debugger tool. If you want to use another debugger then you must add it to the Tools menu.

The **Debugger** tab of the **Setup Customize** dialog box is where the external debugger related information is configured. You may wish to use an older version of the debugger if certain targets are not currently supported in the new environment. Invoke it by selecting the [**Setup->Customize**] menu option and then selecting the **Debugger** tab.

The first choice to make is which debug tool you would like to use.

Once this has been selected the external debugger must be configured.

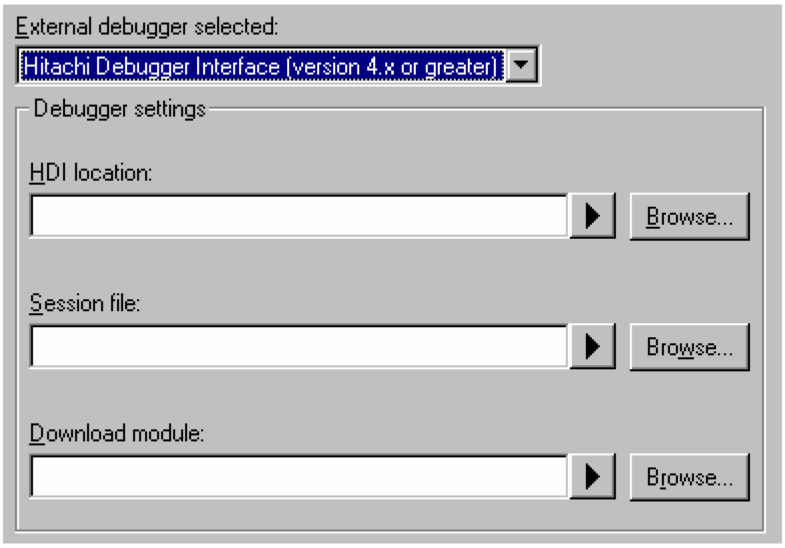
Hitachi Debugger Interface (version 4.x or greater)	Configuring the Hitachi Debugging Interface to integrate with HEW
Renesas PD debugger	Configuring the PD debugger to integrate with HEW
Other external debugger	Configuring an external debugger to integrate with HEW
Non selected	Not use the external debugger

Click the **Launch External Debugger** toolbar button () to invoke the debugger with the specified session file.

After a build, if the download module has been updated, the HEW will switch back to the debugger to enable immediate debugging. Whilst using an external debugger, double-clicking in any editor window will switch back to the HEW with the source file open at the line that was double-clicked.

17.16.1 Configuring the Hitachi Debugging Interface to integrate with HEW

The following details the information required to setup the Hitachi Debugging Interface to integrate with HEW and launch from the external debugger option in HEW.



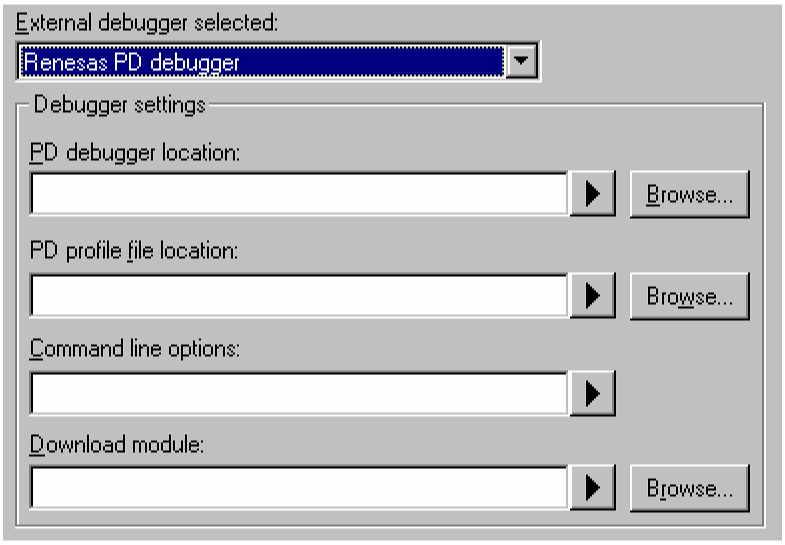
External debugger selected: Hitachi Debugger Interface (version 4.x or greater)

To configuring the Hitachi Debugging Interface to integrate with HEW

1. Firstly, the location of the debugger executable must be specified. This must be version Hitachi Debugging Interface 4.0 or greater, otherwise its behavior is not guaranteed. This may have been configured by the installation program or a project generation utility.
2. The second item of data is the session file. This tells the debugger which session to load when it is launched.
3. Finally, the location of the download module is required. This allows the HEW to automatically switch to the debugger when the download module changes after a build.

17.16.2 Configuring the PD debugger to integrate with HEW

The following details the information required to setup the PD debugger to integrate with HEW and launch from the external debugger option in HEW.



External debugger selected: Renesas PD debugger

Debugger settings

PD debugger location:

PD profile file location:

Command line options:

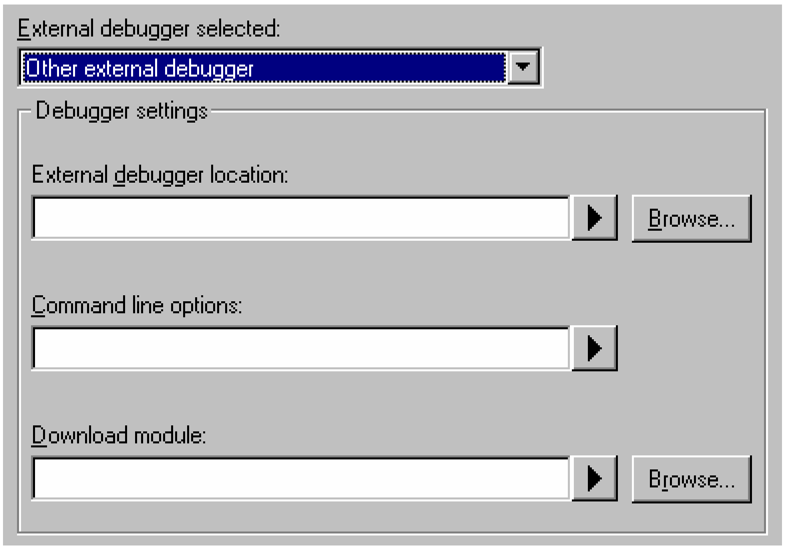
Download module:

To configure the PD debugger to integrate with HEW

1. Firstly, the location of the debugger executable must be specified. This may have been configured by the installation program or a project generation utility.
2. The second item of data is the profile file. This tells the debugger which profile file to load when it is launched. This file stores the debug setup information.
3. The third item of data are the command line options. This field allows additional options to be specified which can modify the behavior of the external debugger.
4. Finally, the location of the download module is required. This allows the HEW to automatically switch to the debugger when the download module changes after a build.

17.16.3 Configuring an external debugger to integrate with HEW

The following details the information required to setup an external debugger which is not Hitachi Debugging Interface or the PD debugger to integrate with HEW and launch from the external debugger option in HEW.



External debugger selected: Other external debugger

Debugger settings

External debugger location:

Command line options:

Download module:

To configure an external debugger to integrate with HEW

1. Firstly, the location of the debugger executable must be specified. This may have been configured by the installation program or a project generation utility.
2. The second item of data are the command line options. This field allows additional options to be specified which can modify the behavior of the external debugger.
3. Finally, the location of the download module is required. This allows the HEW to automatically switch to the debugger when the download module changes after a build.

17.17 Synchronizing multiple debugging platforms

Multiple debugging platforms can be operated at the same time in the HEW. There are two methods available to achieve this. These are outlined below; the external method was available in HEW 2.x. The Internal synchronization of debugger targets is the new preferred method for multiple target debugging in HEW 3.1.

17.17.1 External HEW synchronization

Initiating a HEW from another HEW synchronizes multiple debugging platforms. The HEW that initiates another HEW is called the master, and the initiated HEW is called the slave. Choose [**Tools->Launch Slave HEW**] or click the **Launch Slave HEW** toolbar button () to initiate a slave HEW.

The slave HEW has the same functionality as the master HEW.

The slave HEW is notified of the following actions done in the master HEW to ensure synchronization of the slave HEW and the master HEW.

- Reset go
- Go
- Stop debugging

Note:

The master HEW can initiate multiple slave HEW applications, but slave HEW applications cannot be nested (no slave HEW can initiate another slave HEW).

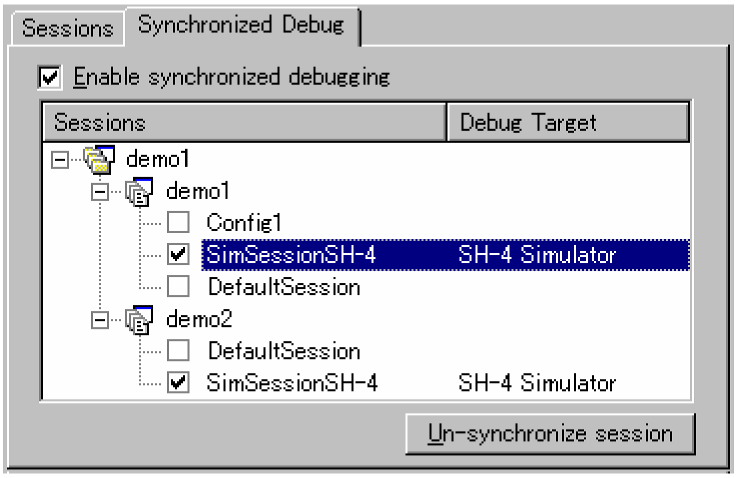
17.17.2 Internal HEW synchronization

The HEW also supports internal multiple target debugging. This will allow you to connect to multiple target components in the same HEW application. These targets can then be debugged simultaneously. The system allows the user to setup a number of sessions with different targets. Then when debugging the sessions can be synchronized so that certain events in one session can trigger the same events in the others. This is very similar to that seen in the External HEW synchronization section above. This facility though has the added advantage that it is easy to swap sessions and see what is happening in the same application.

To setup internal HEW synchronization

1. Select the [**Debug->Debug Sessions**] menu option. The **Debug Sessions** dialog box is displayed.
2. Select the **Synchronized Debug** tab.
3. Select the sessions you wish to synchronize. All currently available sessions in the workspace are displayed.
4. Click the **Synchronize session** button. The icons for these sessions should changed to checked rather than unchecked.

5. Click the **Enable synchronized debugging** check box to switch this facility on.
6. Click OK to verify the changes.

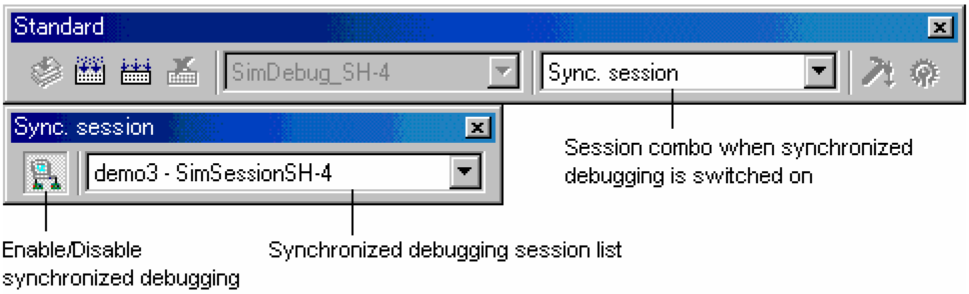


☑ Enable synchronized debugging

Sessions	Debug Target
demo1	
demo1	
□ Config1	
☑ SimSessionSH-4	SH-4 Simulator
□ DefaultSession	
demo2	
□ DefaultSession	
☑ SimSessionSH-4	SH-4 Simulator

To use internal HEW synchronization

1. Follow the options as specified in to setup internal HEW synchronization.
2. Click on the session combo-box located on the standard toolbar. Select the **Sync. session**. This option is only available when synchronized debugging has been added to the system.
3. Once selected the HEW debugger enables the Synchronized debugging facilities. This means the addition of another toolbar this is named **Sync. session**.
4. The enable disable toolbar button on the **Synch. session** toolbar allows you to temporarily switch off the synchronization without losing your settings for this feature.
5. When enabled changing the session in the **Sync. session** combo box changes the session you are currently viewing. In the normal HEW debugging state this would mean the session is closed. In Synch. session you can have multiple sessions open and the currently selected one on this toolbar is the session you are viewing.
6. This system allows you to debug multiple targets or CPU cores simultaneously. Changing the session changes the views you can see on the screen and the data displayed on these views.


Standard
SimDebug_SH-4
Sync. session
Sync. session
demo3 - SimSessionSH-4
Session combo when synchronized debugging is switched on
Enable/Disable synchronized debugging
Synchronized debugging session list

Note:

There are a number of capabilities that are synchronized when this facility is enabled.

The following tables display the capabilities when synchronized debugging is switched on. The example shows what happens in the two synchronized sessions.

Debugging function	Target Debugger Session 1	Target Debugger Session 2
User click "GO" in any Session	"GO"	"GO"
User click "STEP Into/out/over" in any Session	"Step"	"Step"
User click "ESC" in any Session	"BREAK"	"BREAK"
-	"BREAK" by breakpoints, illegal access or due to illegal User Program	Stop running (Same Effect of pressing ESC)
-	Stop running (Same Effect of pressing ESC)	"BREAK" by breakpoints, illegal access or due to illegal User Program
RESET CPU in any Session	RESET CPU	RESET CPU

The following tables display the capabilities when synchronized debugging is switched off. The example shows what happens in the two synchronized sessions.

Debugging function	Target Debugger Session 1	Target Debugger Session 2
User click "GO" in Session 1	"GO"	No activities, "GO" can only be executed manually by User
User click "GO" in Session 2	No activities, "GO" manually executed by User	"GO"
User click "STEP" in session 1	"STEP"	No activities, must "STEP" manually by User
User click "STEP" in session 2	No activities, must "STEP" manually by User	"STEP"
User click "ESC" in session 1	"BREAK"	Still executing, if previously executing User Target Program
User click "ESC" in session 2	Still executing, if previously executing User Target Program	"BREAK"
-	"BREAK" by breakpoints, illegal access or due to illegal User Program	Still executing, if previously executing User Target Program
-	Still executing, if previously executing User Target Program	"BREAK" by breakpoints, illegal access or due to illegal User Program
RESET CPU in session 1	RESET CPU	No activities
RESET CPU in session 2	No activities	RESET CPU

Note:

Another difference to the standard debugging system is that it is possible to view the download modules in the workspace window for all synchronized sessions. This allows modules to be downloaded for any of the available sessions easily.

17.18 Debugging functions dependent on the debugging platform

The debugging functions listed in this section are not common to all HEW products included in tool packages. (User's manual and help information just include descriptions of these debugging functions as the previous version did.) This section covers only some of the functions available with SuperH RISC engine family or H8, H8S, and H8SX family debugging platforms.

For other functions dependent on the debugging platform, see the user's manual or help information packaged with the emulator or simulator.

17.18.1 Looking at labels

In addition to the debugging information that the HEW uses to link your program's source code to the actual code in memory, the **debug object file** also contains symbolic information. This is a table of text names that represent an address in the program and are referred to as **labels** in HEW. In the **Disassembly** view, you will see the first eight characters of the label in place of the corresponding address, and as part of an instruction's operand.

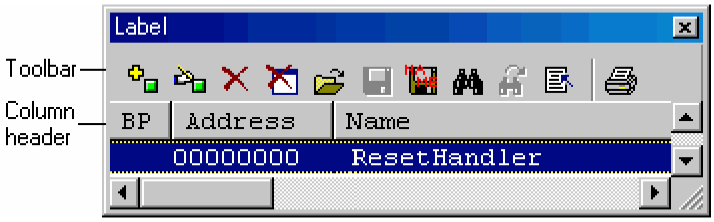
Note:

An instruction's operand is replaced with a label name if the operand and label value match. If two or more labels have the same value, then the label that comes first alphabetically will be displayed.

(1) Listing labels

To see a list of all the labels defined in the current debugger session, click the **View Labels** toolbar button ( ), or select the [**View->Symbol->Labels**] menu option.

Configuration of Label window



BP	Address	Name
	00000000	ResetHandler

- You can view symbols, sorted either alphabetically (by ASCII code) or by address value, by clicking on the respective column heading.
- You can quickly toggle a software break at the entry point of a function by double clicking in the BP (breakpoint) column. Alternatively, right-click to show the pop-up menu and select Break.

Option

Right-clicking displays a pop-up menu containing available options.

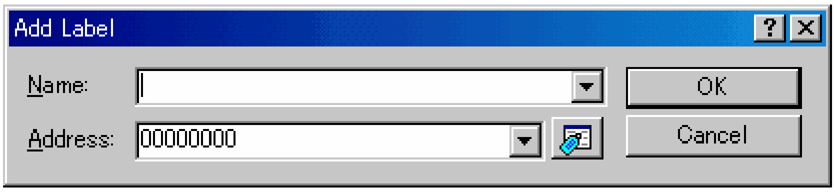
A basic operation is allocated to the toolbar.

The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up menu options	Toolbar button	Macro Recording	Function
Add	✚	●	Adding a label.
Edit	✎	-	Editing a label.
Delete	✕	●	Deleting a label.
Delete All	☒	●	Deleting all labels.
Load	📂	●	Loading labels from a file.
Save	💾	-	Saving labels into a file.
Save As	💾	-	Saving labels into a file.
Find	🔍	-	Searching for a label.
Fine Next	🔍	-	Searching for the next label.
View Source	📄	-	Viewing the source corresponding to a label.
Print	🖨	-	Prints the currently displayed contents.
Toolbar display	-	-	Shows or hides the toolbar.
Customize toolbar	-	-	Customizes toolbar buttons.

(2) Adding a label

Choose **Add** from the pop-up menu and open the **Add Label** dialog box to add a label:



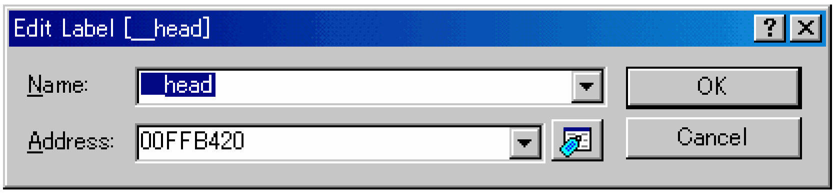
Name:

Address: 00000000

Enter the new label name into the **Name** field and the corresponding value into the **Address** field and press OK. The **Add Label** dialog box closes and the label list is updated to show the new label. When an overloaded function or a class name is entered in the **Address** field, the **Select Function** dialog box opens for you to select a function. For details, see section 17.18.2 (3), Supporting duplicate labels.

(3) Editing a label

Choose **Edit** from the pop-up menu and open the **Edit Label** dialog box to edit a label:


Edit Label [_head]

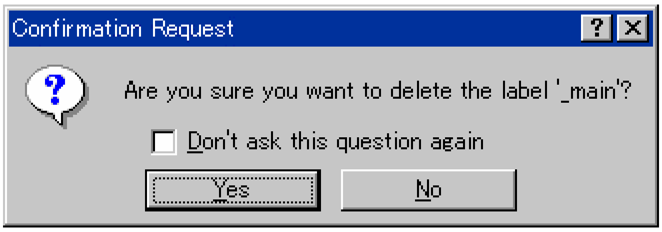
Name: _head

Address: 00FFB420

Edit the label name and value as required and then press OK to save the modified version in the label list. The list display is updated to show the new label details. When an overloaded function or a class name is entered in the **Address** field, the **Select Function** dialog box opens for you to select a function. For details, see section 17.18.2 (3), Supporting duplicate labels.

(4) Deleting a label

To delete a label, select the label and choose **Delete** from the pop-up menu. A confirmation message box appears:


Confirmation Request

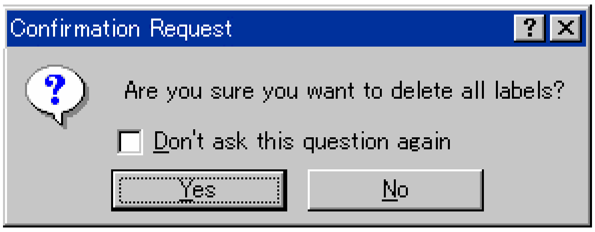
Are you sure you want to delete the label '_main'?

□ Don't ask this question again

If you click OK, the label is removed from the list and the window display is updated.

(5) Deleting all labels

To delete all the labels from the list, choose [**Delete All**] from the pop-up menu. A confirmation message box appears:


Confirmation Request

Are you sure you want to delete all labels?

□ Don't ask this question again

If you click OK, all the labels are removed from the HEW system's symbol table and the list display will be cleared.

(6) Loading labels from a file

A symbol file can be loaded and merged into the HEW's current symbol table. Choose **Load** from the pop-up menu to open the load symbols dialog box:

The dialog box operates like a standard Windows® Open file dialog box; select the file and click **Open** to start loading. The standard file extension for symbol files is ".sym".

(7) Saving labels into a file

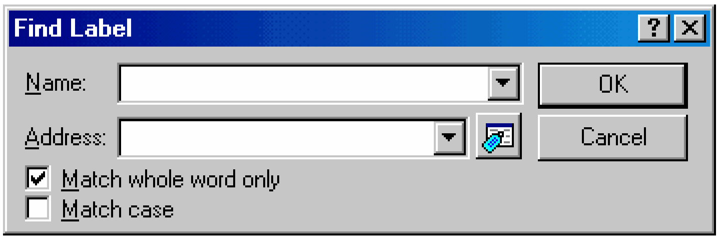
Choose **Save As** from the pop-up menu to open the save symbols dialog box. The save symbols dialog box operates like a standard Windows® Save As dialog box. Enter the name for the file in the **File name** field and click **Save** to save the HEW's current label list to a symbol file. The standard file extension for symbol files is ".sym".

See Reference 7, Symbol File Format, for symbol file format.

Once a file is specified by the **Save As** menu, the current symbol table can be saved in the same symbol file just by choosing **Save** from the pop-up menu.

(8) Searching for a label

Choose **Find** from the pop-up menu to open the **Find Label** dialog box



Find Label

Name:

Address:

☑ Match whole word only

☐ Match case

1. Enter the label name that you wish to find into the **Name** field or select a label name that you have searched before from the drop-down list box.
2. Enter the address of the label that you wish to find into the **Address** field or select an address that you have searched before from the drop-down list box.
3. To find the label having the name entered in the Name field, select the **Match whole word only** check box. When this option is not selected, the HEW searches for any labels that partly match the name entered in the Name field (the Address field cannot be used).
4. When the label name must be case sensitive, select the **Match case** check box.
5. Click the OK button to start the search.

Note:

Only the label is stored by 1024 characters of the start, therefore the label name must not overlap mutually in 1024 characters or less.

(9) Searching for the next label

Choose **Find Next** from the pop-up menu to find the next occurrence of the label containing the text that you entered.

(10)Viewing the source code corresponding to a label

Select a label and choose **View Source** from the pop-up menu to open the source file containing the address corresponding to the label.

17.18.2 Elf/Dwarf2 support

The HEW supports the Elf/Dwarf2 object file format for debugging applications written in C/C++ and assembly language for Renesas microcomputers. It provides a powerful way of accessing, observing and modifying the symbolic level debugging information about the user application that is running.

Key Features:

- Source level debugging
- C/C++ operators
- C/C++ expression (casting, pointers, references, etc.)
- Ambiguous function names
- Overlay memory loading
- Watch – locals and user defined
- Stack trace

(1) C/C++ operators

The C/C++ language operators are available:

```
+, -, *, /, &, |, ^, ~, !, >>, <<, %, (, ), <, >, <=, >=, ==, !=, &&, ||
Buffer_start + 0x1000
#R1 | B'10001101
((pointer + (2 * increment_size)) & H'FFFF0000) >> D'15
!(flag ^ #ER4)
```

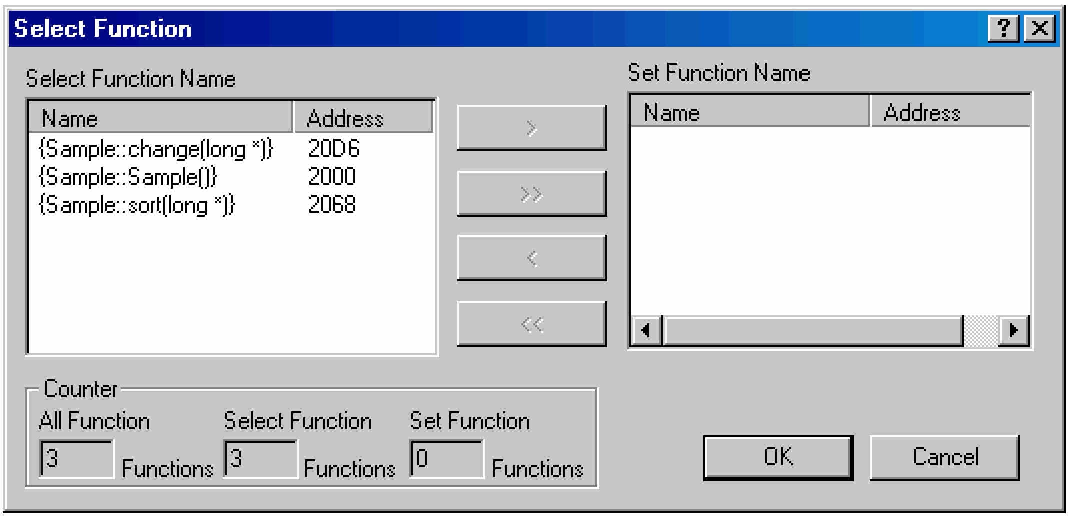
(2) C/C++ expressions

Expression examples:

Object.value	Specifies direct reference of a member (C/C++)
p_Object->value	Specifies indirect reference of a member (C/C++)
Class::value	Specifies reference of a member with class (C++)
*value	Specifies a pointer (C/C++)
&value	Specifies a reference (C/C++)
array[0]	Specifies an array (C/C++)
Object.*value	Specifies reference of a member with pointer (C++)
::g_value	Specifies reference of a global variable (C/C++)
Class::function(short)	Specifies a member function (C++)
(struct STR) *value	Specifies a cast operation (C/C++)

(3) Supporting duplicate labels

In some languages, for example in C++ overloaded functions, a label may represent more than one address. Just entering the label name is ambiguous, so the HEW will display the **Select Function** dialog box to display overloaded functions and member functions.



Select Function

Select Function Name

Name	Address
{Sample::change(long *)}	20D6
{Sample::Sample()}	2000
{Sample::sort(long *)}	2068

Set Function Name

Name	Address

Counter

All Function: 3 Functions

Select Function: 3 Functions

Set Function: 0 Functions

Select overloaded functions or member functions in the **Select Function** dialog box. Generally, only one function can be selected at one time (except for setting breakpoints, as multiple functions can be selected). This dialog box has three areas:

Select Function Name	Displays the same-name functions or member functions and their detailed information.
Name	Displays the names of the functions
Address	Displays the addresses of the functions

Set Function Name	Displays the function to be set and their detailed information.
Name	Displays the names of the functions
Address	Displays the addresses of the functions

Counter	Displays the number of the functions having the same name.
All Function	Displays the number of same-name functions or member functions.
Select Function	Displays the number of functions displayed in the **Select Function Name** list box.
Set Function	Displays the number of functions displayed in the **Set Function Name** list box.

(a) Selecting a function

Click the function you wish to select in the **Select Function Name** list box, and click the > button. You will see the selected function in the **Set Function Name** list box. To select all functions in the **Select Function Name** list box, click the >> button.

(b) Deselecting a function

Click the function you wish to deselect from the **Set Function Name** list box, and click the < button. To deselect all functions, click the <<button. The deselected function(s) will be moved from the **Set Function Name** list box back to the **Select Function Name** list box.

(c) Setting a function

Click the **OK** button to set the functions displayed in the **Set Function Name** list box. The functions are set and the **Select Function** dialog box closes.

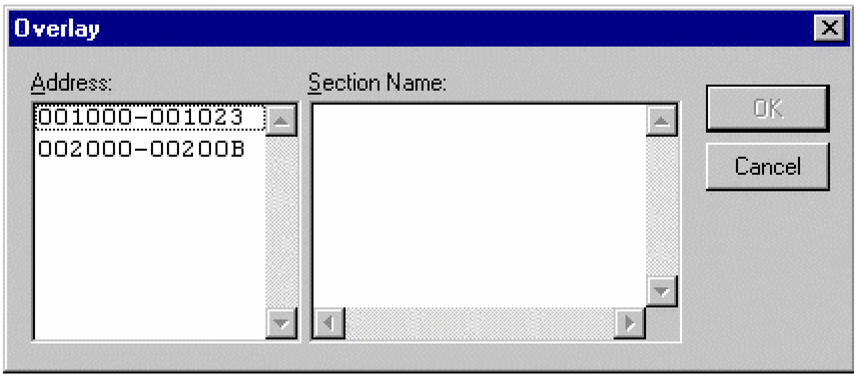
Click the **Cancel** button to close the dialog box without setting the functions.

(4) Debugging an overlay program

Programs making use of the **Overlay** function can be debugged. This section explains the settings for using the **Overlay** function.

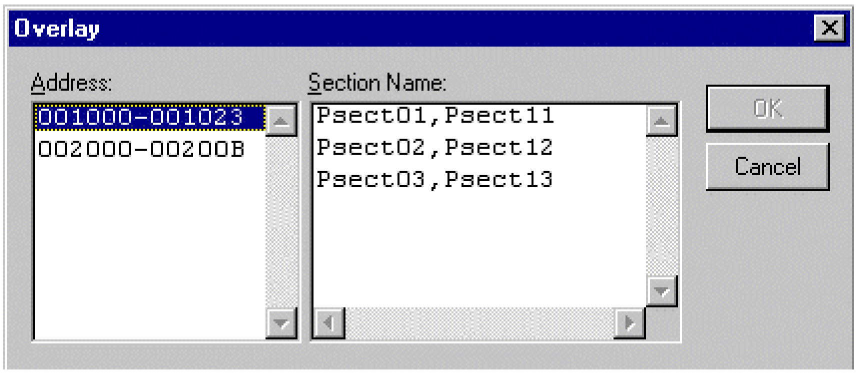
(a) Displaying section group

When the **Overlay** function is used (i.e. when several section groups are assigned to the same address range), the address ranges and section groups are displayed in the **Overlay** dialog box.



001000-001023
002000-00200B

Open the **Overlay** dialog box by choosing the [**Debug->Overlay**] menu option. This dialog box has two areas: the **Address** list box and the **Section Name** list box. The **Address** list box displays the address ranges used by the **Overlay** function. Click to select one of the address ranges in the Address list box.



001000-001023
002000-00200B

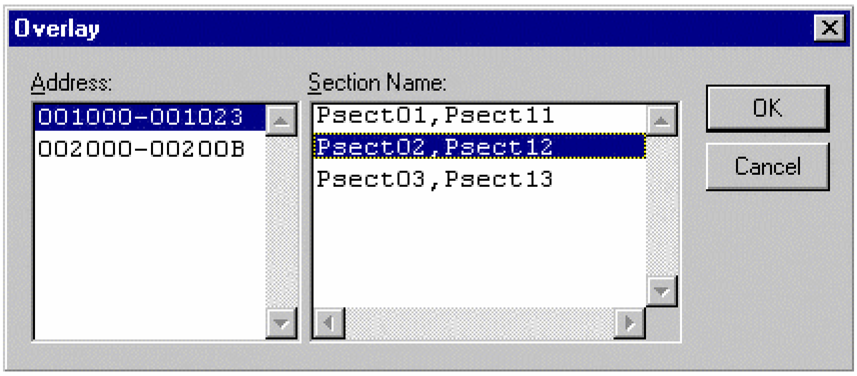
Psect01,Psect11
Psect02,Psect12
Psect03,Psect13

The **Section Name** list box displays the section groups assigned to the selected address range.

(b) Setting section group

When using the **Overlay** function, the highest-priority section group must be selected in the **Overlay** dialog box, otherwise the HEW will operate incorrectly.

Firstly, click one of the address ranges displayed in the **Address** list box. The section groups assigned to the selected address range will then be displayed in the **Section Name** list box. Click to select the section group with the highest-priority among the displayed section groups.



```
001000-001023
002000-00200B
```

```
Psect01,Psect11
Psect02,Psect12
Psect03,Psect13
```

After selecting a section group, clicking the OK button stores the priority setting and closes the dialog box. Clicking the Cancel button closes the dialog box without storing the priority setting.

Note:

Within the address range used by the **Overlay** function, the debugging information for the section specified in the **Overlay** dialog box is referred to. Therefore, the same section of the currently loaded program must be selected in the **Overlay** dialog box.

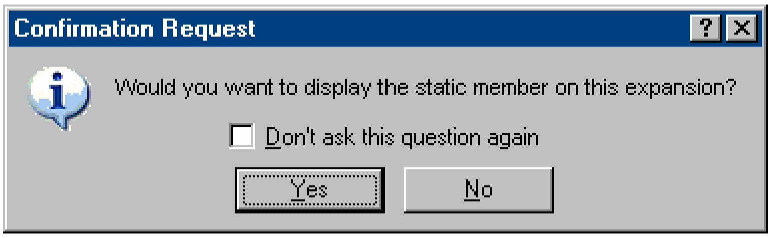
17.18.3 Looking at variables

This section describes how you can look at variables in the source program.

If you are debugging ELF/DWARF2 load modules created by using Renesas' SHC or H8S compiler, static member variables can be displayed together with other member variables when structures are expanded in the **Watch** or **Locals** window. However, it will take time to display the variables if the window contains a large amount of debugging information.

To select not to display static member variables when structures are expanded in the Watch or Locals window

1. Select [**Setup->Options**] to open the **Options** dialog box.
2. Select the **Confirmation** tab.
3. Select the **Display static member on the variable expansion in the Watch/Locals** checkbox. By default, this checkbox is not selected.
4. A confirmation dialog box appears. Select **No**.



Would you want to display the static member on this expansion?

□ Don't ask this question again

The static member variables are hidden even when structures are expanded and this will save time taken for display. (Even after you have selected **No**, it is still possible to view the static member variables by adding them to the **Watch** window.)

(1) Tooltip watch

Use this function to know the value of a variable defined in the source program. Open the editor window or disassembly window (in source mode) to view the source program and rest the mouse cursor over the variable name that you want to examine. A tooltip (pop-up window) will appear showing the watch information.

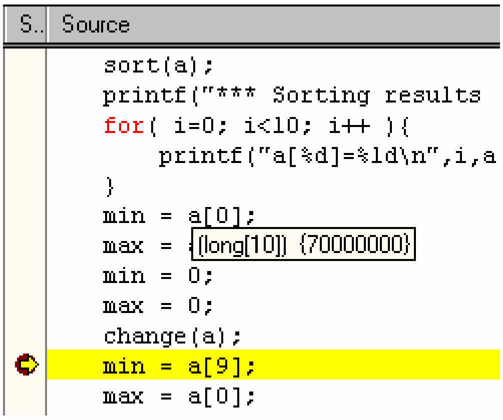
To use Tooltip watch

1. Select [**Setup->Options**] menu option. The **Options** dialog box is displayed.
2. Select the **Editor** tab.
3. Check the **Enable tooltip watch** check box.
4. Click OK.

☑ Enable tooltip watch

To view a tooltip watch on the editor window or disassembly window (in source mode)

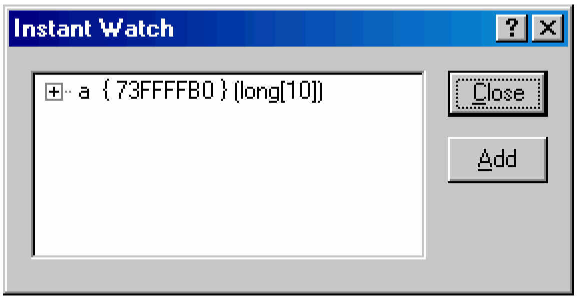
1. Open the editor window or disassembly window (in source mode) showing the variable that you want to examine.
2. Rest the mouse cursor over the variable name that you want to examine. A tooltip will appear near the variable containing basic watch information for that variable.



```
sort(a);
printf("*** Sorting results
for( i=0; i<10; i++ ){
    printf("a[%d]=%ld\n",i,a
}
min = a[0];
max = (long[10]) {70000000}
min = 0;
max = 0;
change(a);
min = a[9];
max = a[0];
```

(2) Instant watch

Display the source file containing the variable that you want to examine on the editor window or disassemble window (in source mode).
Rest the mouse cursor over the variable name that you want to examine and choose **Instant Watch** from the pop-up menu; the **Instant Watch** dialog box will appear and display the variable at the cursor location.



⊞ a { 73FFFFB0 } (long[10])

"+" shown to the left of the variable name indicates that the information may be expanded by clicking on the variable name, and "-" indicates that the information may be collapsed. Clicking **Add** registers the variable in the Watch window. Clicking **Close** closes the window without registering the variable in the Watch window.

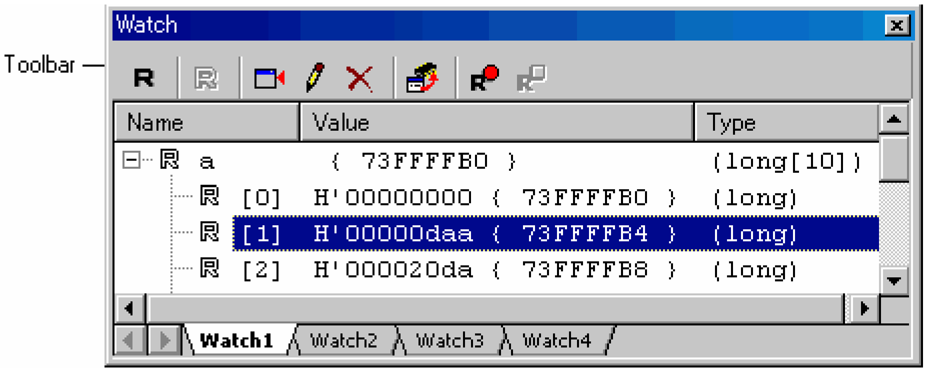
(3) Watch window

The HEW allows you to open Watch windows, which contain a list of variables and their values.

(a) Opening the Watch window

To open a **Watch** window, click on the **Watch** toolbar button ( ) or choose [**View->Symbol->Watch**] if it is visible. A **Watch** window opens. Initially the contents of the window will be blank.

Window configuration



Name	Value	Type
a	{ 73FFFFB0 }	(long[10])
[0]	H'00000000 { 73FFFFB0 }	(long)
[1]	H'00000daa { 73FFFFB4 }	(long)
[2]	H'000020da { 73FFFFB8 }	(long)

- This window allows the user to view and modify C/C++-source level variables.
- The contents of this window are displayed only when the debugging information available in the absolute file (*.abs) includes the information on the C/C++ source program. The variable information is not displayed if the source program information is excluded from the debugging information during optimization by the compiler. In addition, the variables that are declared as macro cannot be displayed.
- A variable can be dragged from the editor window or disassembly window (in source mode) and dropped into the Watch window.

The following items are displayed.

Name	Name of the variable.
Value	Value and assigned location. The assigned location is enclosed by { }.
Type	Type of the variable

The **R** mark shows that the value of the variable can be updated during user program execution. When the color of the **R** mark is black, a value is real-time updated.

Option

Right-clicking displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

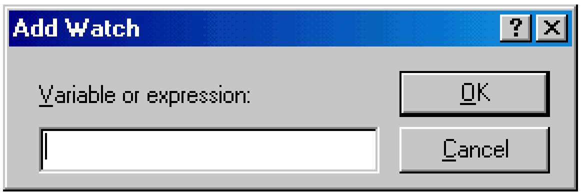
Pop-up menu options		Toolbar button	Macro Recording	Function
Auto Update		R	●	Marks the selected variable with a bold R and updates the variable in real time.
Auto Update All		R	●	Marks all variables with bold Rs and updates all variables in real time.
Delete Auto Update		R	●	Marks the selected variable with an outlined R and cancels realtime update.
Delete Auto Update All		R	●	Marks all variables with outlined Rs and cancels realtime update.
Record Update Value	Start Recording	R	●	Opens the **Recording Settings** dialog box and starts recording the history of updating of values
	Stop Recording	R	●	Stops recording the history of updating of values.
Add Watch			●	Launches the Add Watch dialog box, allowing the user to enter a variable or expression to be watched.
Edit Value		✎	●	Launches the Edit Watch dialog box, allowing the user to change the variable's value.
Delete		✕	●	Removes the variable indicated by the text cursor from the Watch window.
Delete All		☒	●	Removes all the variables from the Watch window.
Radix	Hexadecimal	16	●	Display in hexadecimal.
	Decimal	10	●	Display in decimal.
	Octal	8	●	Display in octal.
	Binary	2	●	Display in binary.
Copy			-	Places a copy of the highlighted text into the Windows® clipboard.
Save As		💾	-	Saves the currently displayed contents.
Go To Memory			-	Opens a Memory window for the address.
Toolbar display		-	-	Shows or hides the toolbar.
Customize toolbar		-	-	Customizes toolbar buttons.

(b) Adding a watch item

Use the **Add Watch** dialog box in the Watch window to add Watch items to the Watch window.

To use Add Watch from a Watch window

1. Open the Watch window.
2. Choose **Add Watch** from the pop-up menu.
3. The Add Watch dialog box opens:



Add Watch

4. Enter the name of the variable that you wish to watch and click OK. The variable is added to the Watch window.

A variable can be dragged from the editor window or disassembly window (in source mode) and dropped into the Watch window.

Note:

If the variable that you have added is a local variable that is not currently in scope, the HEW will add it to the Watch window but its value will be ‘Not available now’.

(c) Expanding a watch item

If a watch item is a pointer, array, or structure, then you will see a plus sign (+) expansion indicator to left of its name, this means that you can expand the watch item. To expand a watch item, click on it. The item expands to show the elements (in the case of structures and arrays) or data value (in the case of pointers) indented by one tab stop, and the plus sign changes to a minus sign (-). If the elements of the watch item also contain pointers, structures, or arrays then they will also have expansion indicators next to them.

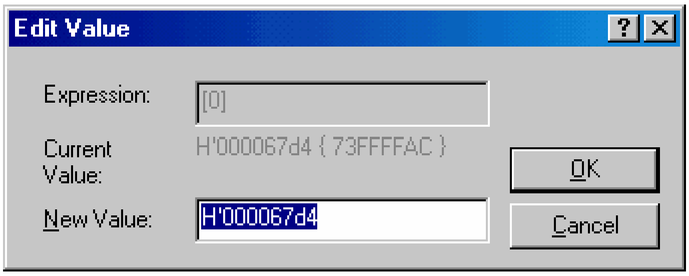
To collapse an expanded watch item, click on the item again. The item's elements will collapse back to the single item and the minus sign changes back to a plus sign.

(d) Editing a watch item's value

You may wish to change the value of a watch variable, e.g. for testing purposes or if the value is incorrect due to a bug in your program. To change a watch item's value use the **Edit Value** function.

To edit a watch item's value

1. Enter a value directly in the window. In another way, select the item to edit by clicking on it, you will see a flashing cursor on the item.
   Choose **Edit Value** from the pop-up menu.
2. The **Edit Value** dialog box opens:



3. Enter the new value or expression in the **New Value** field and click OK. The Watch window is updated to show the new value.

(e) Deleting a watch item

To delete a watch item, select it from the Watch view and choose the **Delete** option from the pop-up menu. The item is deleted and the Watch view is updated.

To delete all watch item, select it from the Watch view and choose the **Delete All** option from the pop-up menu. The all items are deleted and the Watch view is updated.

Watch items that you have set in the Watch window are saved in the session file.

(f) Modifying the radix

The radix for the selected variable display can be modified by choosing **Radix** from the pop-up menu.

(g) Saving the Watch window contents in a file

To save the contents of the Watch window, choose **Save As** from the pop-up menu; the Save As dialog box opens. It allows the user to specify the name of a file and to save the contents of the Watch window in the file. If the **Append** check box is selected, the window contents are appended to the existing file, and if it is not selected, the existing file is overwritten.

(h) Opening a Memory window

The contents of the memory area to which the selected variable is assigned can be displayed in the **Memory** window. Choosing **Go To Memory** from the pop-up menu opens the **Memory** window.

(i) Specifying realtime update

The R mark shown to the left of each variable indicates whether the variable is updated in real time.

A pop-up menu containing the following options is available in the Watch window:

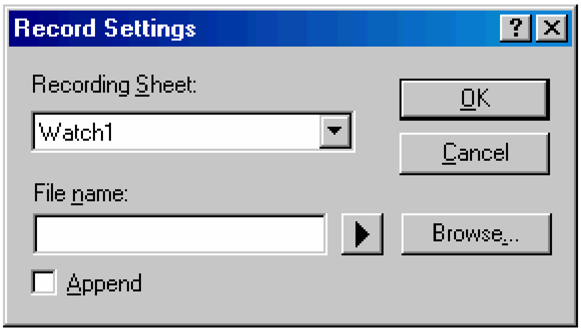
Auto Update	Marks the selected variable with a bold R and updates the variable in real time.
Auto Update All	Marks all variables with bold Rs and updates all variables in real time.
Delete Auto Update	Marks the selected variable with an outlined R and cancels realtime update.
Delete Auto Update All	Marks all variables with outlined Rs and cancels real-time update.

(j) Recording the history of updating of values

The history of updating of the values of watched items, to which automatic update has been applied, can be recorded into a file.

To start recording the history of updating of values

1. Right-click within the window to open a pop-up menu.
2. Select **Record Update Value** and then **Start Recording** from submenus.
3. The **Recording Settings** dialog box opens.



Record Settings

Recording Sheet: Watch1

File name:

□ Append

4. In the **Recording Settings** dialog box, specify how to record the history.
5. Select a sheet that you wish to record the history in **Recording Sheet**. Only one sheet can be used for recording at one time.
6. Enter the file name in **File name**. To insert a placeholder in **File name**, click the **Placeholder** button and select **Placeholder** from the pop-up menu. To browse a file, click the **Browse** button.
7. To append data to the specified file, select the **Append** checkbox.
8. Clicking the OK button starts recording except in the following cases:

- The sheet selected in **Recording Sheet** does not contain watched items to which automatic update has been applied.
- The **Append** checkbox is selected but the watched items recorded in the file selected in **File name** and those to be recorded contained in the sheet selected in **Recording Sheet** do not match or the radixes are different.

Once recording is started, the values of variables are recorded into the file when:

- Values are automatically updated (in realtime) during execution
- Values are updated with the stop of execution
- Values are changed
- Display is refreshed by an operation in the window

To stop recording the history of automatically updated values

1. Right-click within the window to open a pop-up menu.
2. Select **Record Update Value** and then **Stop Recording** from submenus.

Recording is automatically stopped when:

- A program is downloaded or unloaded
- Expansion of watched items to be recorded is collapsed
- The radix for watched items to be recorded is changed
- A new item to be watched is added to the sheet selected for recording
- The order of watched items to be recorded is changed
- Automatic update of watched items to be recorded is canceled
- The Watch window is closed

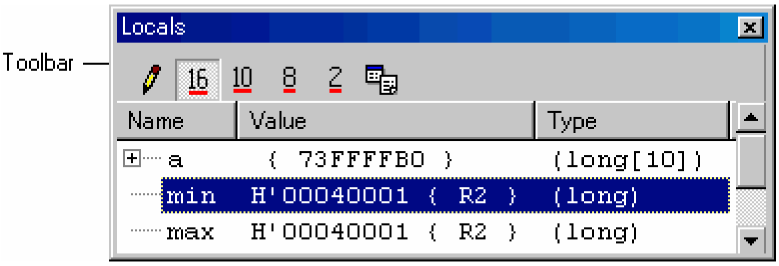
(4) Locals window

The local variables and their values can be displayed in the **Locals** window.

(a) Opening the Locals window

To open the **Locals** window, click the **Locals** toolbar button () or choose [**View->Symbol->Locals**].

Window configuration



Name	Value	Type
⊞ a	{ 73FFFFB0 }	(long[10])
min	H'00040001 { R2 }	(long)
max	H'00040001 { R2 }	(long)

- As you debug your program, the **Locals** window will be updated. If a local variable is not initialized when defined, then the value in the **Locals** window will be undefined until a value is assigned to the local variable.
- The local variable values and the radix for local variable display can be modified in the same manner as in the Watch window.

Option

Right-clicking displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of **Toolbar display** and **Customize toolbar** are also included in the pop-up menu displayed by right-clicking the toolbar area.

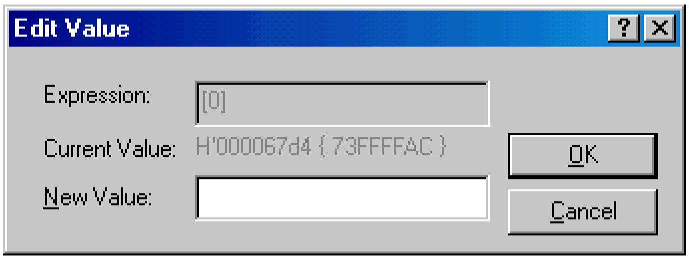
Pop-up menu options		Toolbar button	Macro Recording	Function
Edit Value		✎	●	Launches a dialog box to modify the selected variable's value.
Radix	Hexadecimal	16	-	Display in hexadecimal.
	Decimal	10	-	Display in decimal.
	Octal	8	-	Display in octal.
	Binary	2	-	Display in binary.
Copy		⧉	-	Places a copy of the highlighted text into the Windows® clipboard.
Toolbar display		-	-	Shows or hides the toolbar.
Customize toolbar		-	-	Customizes toolbar buttons.

(b) Editing a local item's value

You may wish to change the value of a local variable, e.g. for testing purposes or if the value is incorrect due to a bug in your program. To change a watch item's value use the Edit Value function.

Editing a local item's value

1. Enter a value directly in the window. In another way, select the item to edit by clicking on it, you will see a flashing cursor on the item. Choose **Edit Value** from the pop-up menu.
2. The **Edit Value** dialog box opens:



Expression: [0]

Current Value: H'000067d4 { 73FFFFAC }

New Value:

3. Enter the new value or expression in the **New Value** field and click OK.
4. The **Locals** window is updated to show the new value.

(c) Modifying the radix

The radix for the selected variable display can be modified by choosing **Radix** from the pop-up menu.

18. Technical Support

18.1 Check for updates

To check for HEW product updates or service packs

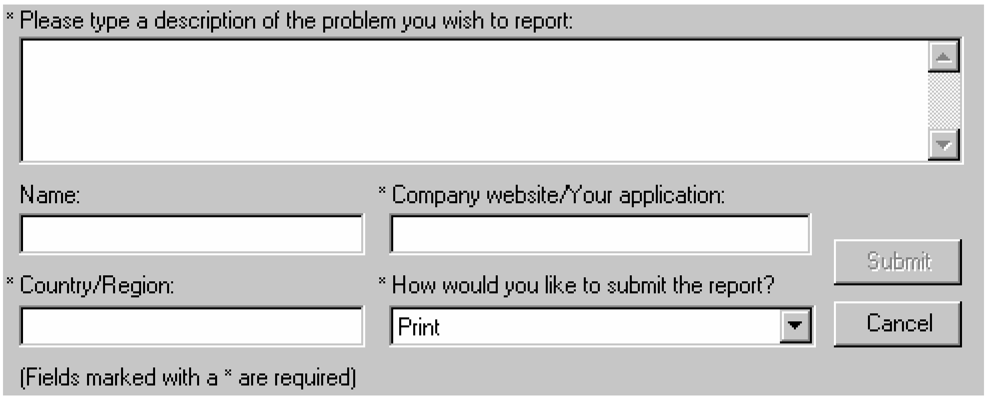
1. Select [**Help->Technical Support->Check Website For Updates**].
2. Your default web browser is invoked and defaults to the HEW download page for your region.
3. Browse this area for HEW updates to fix bugs or add new features.

18.2 Creating a bug report

Occasionally you may experience some unforeseen problems with the HEW application. If a problem does occur that results in an application crash the HEW bug tracking program will be invoked automatically. This allows you to compile a bug report and this can then be sent to your technical support contact in a variety of ways. It is also possible to invoke this tracker program manually. This is described below:

To create and send a HEW bug report

1. Select [**Help->Technical Support->Create Bug Report**] menu option.
2. Detailed information is generated from your HEW system. This may take some time. The **Submit a Bug Report** dialog box is then displayed.
3. Write the description of your problem in the **Please type a description of the problem you wish to report** edit box. (This item must be completed.)
4. Type your name in the **Name** edit box.
5. Type your company URL or purpose of the product being developed in the **Company website/Your application** edit box. (This item must be completed.)
6. Type your country or region in the **Country/Region** edit box. (This item must be completed.)
7. Once you are happy with your report, choose the method of sending the report in the **How would you like to submit the report?** drop-down list box. You can print it, e-mail it, or save it to a disk. (This selection is required.)
8. Then click **Submit**. This will send the report.



* Please type a description of the problem you wish to report:

Name:

* Company website/Your application:

* Country/Region:

* How would you like to submit the report? Print

(Fields marked with a * are required)

Note:

Fields marked with an asterisk (*) are required. You can click the **Submit** button after these fields are completed.

Reference

1. List of Menus

1.1 List of File Menu

Menu	Menu Option	Keyboard Shortcuts	Toolbar Button	Macro Recording *1	Function
File	New	CTRL+N	🗋	-	Creates a new document.
	Open	CTRL+O	📂	-	Opens an existing document.
	Close	CTRL+F4	-	-	Closes the active document.
	New Workspace	-	-	-	Creates a new workspace.
	Open Workspace	-	-	●	Opens an existing workspace.
	Save Workspace	-	-	●	Saves the current workspace.
	Close Workspace	-	-	●	Closes the current workspace.
	New Session	-	-	●	Creates a new session.
	Import Session	-	-	●	Imports an existing session.
	Save Session	-	-	●	Saves the current session.
	Save Session As	-	-	-	Saves the current session with a new session name.
	Refresh Session	-	-	●	Reloads the session file for the current session.
	Download A New Module	-	-	●	Downloads a new module.
	Save	CTRL+S	💾	-	Saves the active document.
	Save As	-	-	-	Saves the active document with a new file name.
	Save All	CTRL+SHIFT+S	💾	-	Saves all modified documents in the workspace.
	Page Setup	-	-	-	Changes the printing options.
	Print	CTRL+P	🖨	-	Prints the active document.
	Recent Files	-	-	-	Opens this document.
	Recent Workspaces	-	-	●	Opens this workspace.
	Recent Test Suites	-	-	-	Opens this test suite.
	Recent Downloaded Modules	-	-	●	Downloads this module.
	Exit	-	-	-	Exits HEW.

*1. Operations with some menu items can be recorded as HEW command-line commands by the macro-recording support facility. A macro record icon (●) in the "Macro Recording" column of a menu item indicates that this function can be recorded into a HEW macro file. For details, see section 15.5.1, Recordable functions (common to all HEW products).

1.2 List of Edit Menu

Menu	Menu Option	Keyboard Shortcuts	Toolbar Button	Macro Recording *1	Function
Edit	Undo	CTRL+Z	-	-	Reverses the last editing operation.
	Redo	CTRL+Y	-	-	Repeats the last undone editing operation.
	Cut	CTRL+X	✂	-	Removes highlighted text and places it on the Windows® clipboard.
	Copy	CTRL+C	📄	-	Places a copy of the highlighted text into the Windows® clipboard.
	Paste	CTRL+V	📋	-	Copies the contents of the Windows® clipboard into the active window at the position of the insertion cursor.
	Clear	Delete	-	-	Removes highlighted text (it is not copied to the Windows® clipboard)
	Select All	CTRL+A	-	-	Selects (i.e. highlights) the entire contents of the active window
	Find	CTRL+F	🔍	-	Finds text in the current file.
	Find In Files	F4	🔍	-	Finds text in multiple files.
	Replace	CTRL+H	-	-	Replaces text in the current file.
	Goto Line	CTRL+G	-	-	Jumps to a line in a file.
	Match Braces	CTRL+SHIFT+M	{•}	-	Finds a matching brace.

Bookmarks	Toggle Bookmark	CTRL+F2	▣	-	Sets a bookmark at the current line or clears a bookmark at the current line.
	Next Bookmark	F2	▣	-	Jumps to the next bookmark in the current file from the current line.
	Previous Bookmark	SHIFT+F2	▣	-	Jumps to the previous bookmark in the current file from the current line.
	Clear All Bookmarks	-	▣	-	Clears all bookmarks in the current file.
Templates	Define Templates	-	T	-	Defines a template.
	Insert Template	CTRL+SHIFT+T	T	-	Inserts a template.
Toggle Breakpoint		F9	✋	●	Sets or clears a software breakpoint at the line showing the address.
Enable/Disable Breakpoint		CTRL+F9	✋	●	Enables or disables the current software breakpoint.
Define Column Format		-	-	-	Sets the status of editor columns.
Source Breakpoints		CTRL+B	-	-	Opens the **Breakpoins** dialog box.
Evaluate		-	-	-	Evaluates simple and complex expressions.

*1. Operations with some menu items can be recorded as HEW command-line commands by the macro-recording support facility. A macro record icon (●) in the "Macro Recording" column of a menu item indicates that this function can be recorded into a HEW macro file. For details, see section 15.5.1, Recordable functions (common to all HEW products).

1.3 List of View Menu

Menu	Menu Option		Keyboard Shortcuts	Toolbar Button	Macro Recording *1	Saving into Test-Image File *2	Function
View	Differences		-	-	-	-	Opens the **Difference** window.
	Map *3		-	MAP	-	-	Opens the map window.
	Command Line		CTRL+L	>_	-	-	Opens the **Command Line** window.
	TCL Toolkit		CTRL+SHIFT+K	▣	-	-	See the "Tcl/Tk Additional document".
	Workspace		ALT+K	▣	●	-	Opens the **Workspace** window.
	Output		ALT+O	▣	-	💾*4	Opens the **Output** window.
	Disassembly		CTRL+D	▣	●	-	Opens the **Disassembly** window.
	CPU	Registers	CTRL+R	R1	●	💾	Opens the **Registers** window.
		Memory	CTRL+M	▣	●	💾	Opens the **Memory** window.
		IO	CTRL+I	I/O	●	💾	Opens the **IO** window.
		Status	CTRL+U	▣	-	💾	Opens the **Status** window.
	Graphic	Image	CTRL+SHIFT+G	▣	-	-	Opens the **Image** window.
		Waveform	CTRL+SHIFT+V	▣	-	-	Opens the **Waveform** window.
	Code	Stack Trace	CTRL+K	▣	-	💾	Opens the **Stack Trace** window.

*1. Operations in some windows can be recorded as HEW command-line commands by the macro-recording support facility. A macro record icon (●) in the **Macro Recording** column of a menu item indicates that this function can be recorded into a HEW macro file. For details, see section 15.5.1, Recordable functions (common to all HEW products).

Note that a function of another window dependent on the debugging platform can also be recorded as a command even though this window is not included in the list of menus. For details, see section 15.5.2, Recordable functions (dependent on the debugging platform).

*2. Data in some windows can be saved as test-image data into test-image files by the test-support facility. A save file icon (●) in the **Saving into Test-Image File** column of a menu item indicates that this data can be saved into a test-image file. For details, see section 16.6.1, Functions that can be saved into test-image files (common to all HEW products).

Note that some functions of other windows dependent on the debugging platform can also be saved into test-image files even though these windows are not included in the list of menus. For details, see section 16.6.2, Functions that can be saved into test-image files (dependent on the debugging platform).

***3**. Support for this function depends on the debugging platform.

***4**. The data of the Build tab or the Debug can be saved.

1.4 List of Project Menu

Menu	Menu Option	Keyboard Shortcuts	Toolbar Button	Macro Recording *1	Function
Project	Set Current Project	-	-	●	Set this project as the current project.
	Insert Project	-	-	●	Add project to workspace.
	Dependent Projects	-	-	-	Show dependent projects.
	Edit Project Configuration ***2**	-	-	●	Edit the project configuration.
	Create Project Type	-	-	-	Create a new project type.
	Add Files	-	-	-	Add files to project.
	Remove Files	-	-	-	Remove files from project.
	File Extensions	-	-	-	Display current project file extensions.
	Components	-	-	-	Load/unload components.

*1. Operations with some menu items can be recorded as HEW command-line commands by the macro-recording support facility. A macro record icon (●) in the "Macro Recording" column of a menu item indicates that this function can be recorded into a HEW macro file. For details, see section 15.5.1, Recordable functions (common to all HEW products).

***2**. Support for this function depends on the debugging platform.

1.5 List of Build Menu

Menu	Menu Option	Keyboard Shortcuts	Toolbar Button	Macro Recording *1	Function
Build ***2**	Toolchain	-	-	-	Sets options for a build phase.
	Build File	CTRL+F7		●	Builds the selected files.
	Build	F7		●	Builds out of date project files.
	Build All	-		●	Builds project files, regardless of whether the project files are out of date.
	Build Multiple	-	-	●	Builds multiple projects.
	Clean Current Project	-	-	●	Cleans the current configuration in this project
	Clean All Project	-	-	●	Cleans all configurations in all projects in this workspace.
	Update All Dependencies	-	-	-	Updates a project's dependencies.
	Stop Tool Execution	CTRL+Break		-	Stops tool execution.
	Include/Exclude Build	-	-	-	Excludes a file from build or cancels the exclusion.
	Build Phase	-	-	-	Adds, removes and modifies phase.
	Build Configurations	-	-	●	Selects the current configuration.
	Linkage Order	-	-	-	Customizes the HEW linkage order.
	Generate Makefile	-	-	-	Generates a makefile.

*1. Operations with some menu items can be recorded as HEW command-line commands by the macro-recording support facility. A macro record icon (●) in the "Macro Recording" column of a menu item indicates that this function can be recorded into a HEW macro file. For details, see section 15.5.1, Recordable functions (common to all HEW products).

*2. This menu is not displayed while a debug-only project "Debugger only - xxxxxx" created by HEW V.4.01 or later is in use.

1.6 List of Debug Menu

Menu	Menu Option		Keyboard Shortcuts	Toolbar Button	Macro Recording *1	Function
Debug	Debug Sessions		-	-	●	Opens the **Debug Sessions** dialog box to list, add, or remove the debug session.
	Debug Settings		-	-	-	Opens the **Debug Settings** dialog box to set the debugging conditions or download modules.
	Reset CPU		-	▣	●	Resets the microprocessor.
	Go		F5	▣	●	Starts executing the user program at the current PC.
	Reset Go		SHIFT+F5	▣	●	Executes the user program from the reset vector address.
	Go to Cursor		-	▣	●	Starts executing the user program at the current PC and continues until the PC equals the address indicated by the current text cursor position.
	Set PC to Cursor		-	PC	●	Changes the value of the Program Counter (PC) to the address at the row of the text cursor.
	Run		-	-	●	Launches the **Run Program** dialog box allowing the user to enter temporary breakpoints before executing the user program.
	Display PC		CTRL+ SHIFT+Y	PC	-	Opens the editor or **Disassembly** window at the address of the PC.
	Step In		F11	{}	●	Executes a block of user program before breaking.
	Step Over		F10	{}	●	Executes a block of user program before breaking. If a subroutine call is reached, then the subroutine will not be entered.
	Step Out		SHIFT+ F11	{}	●	Executes sufficient user program to reach the end of the current function.
	Step		-	-	●	Launches the **Step Program** dialog box allowing the user to modify the settings for stepping.
	Step Mode	Auto	-	-	●	Steps only one source line when the editor window is active. When the **Disassembly** window is active, stepping is executed in a unit of assembly instructions.
		Assembly	-	-	●	Executes stepping in a unit of assembly instructions.
		Source	-	-	●	Steps only one source line.
	Halt Program		-	STOP	●	Stops the execution of the user program.
	Initialize		-	-	●	Disconnects the debugging platform and connects it again.
	Connect ***2**		-	▣	●	Connects the debugging platform.
	Disconnect ***2**		-	▣	●	Disconnects the debugging platform.
	Save Memory		-	-	●	Saves the specified memory area data to a file.
	Verify Memory ***2**		-	-	●	Verifies file contents against memory contents.
	Download Modules		-	-	●	Downloads the object program.
	Unload Modules		-	-	●	Unloads the object program.

*1. Operations with some menu items can be recorded as HEW command-line commands by the macro-recording support facility. A macro record icon (●) in the "Macro Recording" column of a menu item indicates that this function can be recorded into a HEW macro file. For details, see section 15.5.1, Recordable functions (common to all HEW products).

*2. Support for this function depends on the debugging platform.

1.7 List of Setup Menu

Menu	Menu Option		Keyboard Shortcuts	Toolbar Button	Macro Recording *1	Function
Setup	Customize		-	-	-	Customizes the HEW application.
	Options		-	-	-	Sets option of the HEW application.
	Format Views		-	-	-	Configures fonts, colors, keywords and so on, for the window.
	Radix	Hex	-	16	●	Sets radix to base Hex.
		Decimal	-	10	●	Sets radix to base Decimal.
		Oct	-	8	●	Sets radix to base Octal.
		Bin	-	2	●	Sets radix to base Binary.

*1. Operations with some menu items can be recorded as HEW command-line commands by the macro-recording support facility. A macro record icon (●) in the "Macro Recording" column of a menu item indicates that this function can be recorded into a HEW macro file. For details, see section 15.5.1, Recordable functions (common to all HEW products).

1.8 List of Tools Menu

Menu	Menu Option		Keyboard Shortcuts	Toolbar Button	Function
Tools	Administration		-	-	Controls the components.
	Change Toolchain Version		-	-	Changes toolchain version.
	Version Control	Select	-	-	Selects a version control system.
		Configure	-	-	Setups the version control system.
		Add to VCS	-	🔒	Adds files to Visual SourceSafe.
		Remove from VCS	-	🔒	Gets a read-only copy of a file from Visual SourceSafe.
		Get from VCS	-	🔒	Gets a read-only copy of a file from Visual SourceSafe.
		Check out from VCS	-	🔓	Checks out a writable copy of a file from Visual SourceSafe.
		Check in VCS	-	🔒	Checks in your edits to a file into Visual SourceSafe.
		Get VCS status	-	🔍	Views the status of a file in Visual SourceSafe.
	Launch External Debugger		-	↓	Launches an external debugger tool.
	Launch Slave HEW		-	⚙	Launches a slave HEW.
	Macros		-	-	Displays **Macro** dialog box.
	Record Macro		-	●	Starts macro recording.
	Play Macro		-	▷	Plays macro.
	Stop Macro		-	□	Stops macro.

Under the **Stop Macro** menu option, you can add custom menu options to which recorded macros have been assigned.

1.9 List of Test Menu

Menu	Menu Option	Keyboard Shortcuts	Toolbar Button	Function
Test	Create New Test Suite	-	-	Creates a new blank test suite so you can begin adding and running tests.
	Open Test Suite	-	-	Opens an existing test suite.
	Edit Test Suite	-	-	Edits the current test suite. Allows you to add and remove tests to the test suite.
	Close Test Suite	-	-	Closes the current test suite.
	Create New Test Image File	-	-	Setups and customizes the test image data saved to the test image file (*.HIF).
	Compare Test Image File	-	-	Compares test image file allows you to compare a test image file with the current HEW system or with another test image file already on your disk. The results are then displayed in the test browser.
	Run Tests	-	-	Runs tests allows you to run multiple tests that you have defined in the test suite and see the results of the comparisons in the test browser. There are various options to configure the test run execution.
	Test Result Browser	-	-	Displays the test results for one or more test executions. It shows the pass and fail results and the detailed reason why the test failed.

1.10 List of Window Menu

Menu	Menu Option		Keyboard Shortcuts	Toolbar Button	Function
Window	Cascade		-	-	Arrange all open windows so that they overlap.
	Tile Horizontally		-	-	Arrange all open windows horizontally.
	Tile Vertically		-	-	Arrange all open windows vertically.
	Arrange Icons		-	-	Line up all minimized windows.
	Close All		-	-	Close all open windows.
	Virtual desktop	Desktop Manager	-	-	Rename your configuration to a more meaningful name.
		Default 1-4	-	-	Switch desktop configurations.

1.11 List of Help Menu

Menu	Menu Option		Keyboard Shortcuts	Toolbar Button	Function
Help	Help Topic		-	-	Opens the main High-performance Embedded Workshop help window.
	Technical Support	Create Bug Report	-	-	Create a High-performance Embedded Workshop bug report.
		Check Website For Updates	-	-	Check for High-performance Embedded Workshop product updates or service packs.
	About High-performance Embedded Workshop		-	-	Launches the About High-performance Embedded Workshop dialog box allowing the user to view the version of High-performance Embedded Workshop.
	Debugger Help		-	-	Shows the help window of the emulator or simulator when the debugging platform is connected.

2. List of Windows

Window Name	Menu
Differences	[View->Difference]
Map Section Information	[View->Map]
Map Symbol Information	[View->Map]
Command Line	[View->Command Line]
Console ***1**	[View->TCL Toolkit]
Workspace	[View->Workspace]
Output	[View->Output]
Disassembly	[View->Disassembly]
Registers	[View->Registers]
Memory	[View->Memory]
IO	[View->CPU->IO]
Status	[View->CPU->Status]
Image	[View->Graphic->Image]
Waveform	[View->Graphic->Waveform]
Stack Trace	[View->Code->Stack Trace]

***1**. See the "Tcl/Tk Additional document".

3. List of Commands

3.1 Command List (Alphabetically Order)

Command Name	Abbreviation	Description
!	-	Comment.
ADD_FILE	AF	Adds a file to the current project.
ASSERT	-	Checks if an expression is true or false.
AUTO_COMPLETE	AC	Switches the auto-completion.
BUILD ***1**	BU	Performs a build on the current project.
BUILD_ALL ***1**	BL	Performs a build all on the current project.
BUILD_FILE ***1**	BF	Performs a build on the file.
BUILD_MULTIPLE ***1**	BM	Performs a build on the multiple projects and configurations.
CACHE ***2**	-	Sets caching on or off.
CHANGE_CONFIGURATION	CC	Sets the configuration to the specified configuration name.
CHANGE_PROJECT	CP	Sets the specified project file as the current project.
CHANGE_SESSION	CS	Sets the specified session as the current session.
CHANGE_SUB_SESSION ***2**	CB	Changes the current sub-session when using multiple session debugging.
CLEAN	CL	Deletes intermediate and output files produced in building.
CLEAR_OUTPUT_WINDOW	COW	Clears the contents of the specified tab in the output window.
CLOSE_TEST_SUITE	CTS	Closes the current test suite.
CLOSE_WORKSPACE	CW	Closes a workspace.
COMPARE_TEST_DATA	CTD	Compares test data and create results.
CONNECT ***2**	CN	Connects the debugging platform.
DEFAULT_OBJECT_FORMAT	DO	Sets the object format to be used by default.
DISCONNECT ***2**	DN	Disconnects the debugging platform.
ERASE	ER	Clears the Command Line window.
EVALUATE	EV	Evaluates an expression.
FILE_LOAD	FL	Loads an object (program) file.
FILE_LOAD_ALL	LA	Loads all object (program) files.
FILE_SAVE	FS	Saves memory to a file.
FILE_UNLOAD	FU	Unloads an object (program) file from memory.
FILE_UNLOAD_ALL	UA	Unloads all object (program) files from memory.
FILE_VERIFY ***2**	FV	Verifies file contents against memory.
GENERATE_MAKE_FILE ***1**	GM	Generates a build makefile for the current workspace.
GO	GO	Runs program.
GO_RESET	GR	Runs program from reset.
GO_TILL	GT	Runs program until specified addresses.
HALT	HA	Halts program.
HELP	HE	Displays help for Command Line or help on a command.
INITIALIZE	IN	Initializes the debugging platform system.
LOG	LO	Controls command output logging.
MEMORY_COMPARE ***2**	MC	Compares memory contents.
MEMORY_DISPLAY	MD	Displays memory contents.
MEMORY_EDIT	ME	Modifies memory contents.
MEMORY_FILL	MF	Fills a block of memory.
MEMORY_FIND ***2**	MI	Finds a string in an area of memory.
MEMORY_MOVE	MV	Moves a block of memory.
MEMORY_TEST ***2**	MT	Tests a block of memory.
OPEN_TEST_SUITE	OTS	Opens a test suite.
OPEN_WORKSPACE	OW	Opens the specified workspace file.
QUIT	QU	Exits HEW.
RADIX	RA	Sets default input radix.
REFRESH_SESSION	RSE	Reloads the session file.
REMOVE_FILE	REM	Removes a file from the current project.
RESET	RE	Resets the microprocessor.
RUN_TEST	RT	Runs a test.
SAVE_SESSION	SE	Saves the current session.
SAVE_WORKSPACE	SW	Saves the current workspace.
SET_DISASSEMBLY_SOFT_BREAK	SDB	Sets or deletes a software breakpoint at the disassembly level.
SET_SOURCE_SOFT_BREAK	SSB	Sets or deletes a software breakpoint at the source level.
SLEEP	-	Delays command execution.
STATE_DISASSEMBLY_SOFT_BREAK	TDB	Enables or disables a software breakpoint at the disassembly level.
STATE_SOURCE_SOFT_BREAK	TSB	Enables or disables a software breakpoint at the source level.
STEP	ST	Steps through program (by instructions or source lines).
STEP_MODE	SM	Sets the step mode.
STEP_OUT	SP	Steps out of the current function.

STEP_OVER	SO	Steps through program without stepping into functions.
STEP_RATE	SR	Sets rate of stepping.
SUBMIT	SU	Executes a file of commands.
TCL	-	Turns TCL commands on or off.
TOOL_INFROMATION	TO	Outputs the tool information.
UPDATE_ALL_DEPENDENCIES ***1**	UD	Updates the current projects build dependencies.

***1.** Available only when there is a toolchain installed.

***2.** Support for this command depends on the debugging platform.

For the syntax of each command, refer to the online help.

3.2 Command List (Listed as the Functions)

HEW Application Control Commands

Command Name	Abbreviation	Description
ADD_FILE	AF	Adds a file to the current project.
CHANGE_CONFIGURATION	CC	Sets the configuration to the specified configuration name.
CHANGE_PROJECT	CP	Sets the specified project file as the current project.
CHANGE_SESSION	CS	Sets the specified session as the current session.
CHANGE_SUB_SESSION ***1**	CB	Changes the current sub-session when using multiple session debugging.
CLEAR_OUTPUT_WINDOW	COW	Clears the contents of the specified tab in the output window.
CLOSE_WORKSPACE	CW	Closes the specified workspace file.
EVALUATE	EV	Evaluates an expression.
OPEN_WORKSPACE	OW	Opens the specified workspace file.
QUIT	QU	Exits HEW.
RADIX	RA	Sets default input radix.
REFRESH_SESSION	RSE	Reloads the session file.
REMOVE_FILE	REM	Removes a file from the current project.
SAVE_SESSION	SE	Saves the current session.
SAVE_WORKSPACE	SW	Saves the current workspace.
TOOL_INFORMATION	TO	Outputs the tool information.

***1**. Support for this command depends on the debugging platform.

Build Commands (Available only when there is a toolchain installed)

Command Name	Abbreviation	Description
BUILD	BU	Performs a build on the current project.
BUILD_ALL	BL	Performs a build all on the current project.
BUILD_FILE	BF	Performs a build on the file.
BUILD_MULTIPLE	BM	Performs a build on the multiple projects and configurations.
CLEAN	CL	Deletes intermediate and output files produced in building.
GENERATE_MAKE_FILE	GM	Generates a build makefile for the current workspace.
UPDATE_ALL_DEPENDENCIES	UD	Updates the current projects build dependencies.

Command Line Operating Commands

Command Name	Abbreviation	Description
!	-	Comment.
ASSERT	-	Checks if an expression is true or false.
AUTO_COMPLETE	AC	Switches the auto-completion.
ERASE	ER	Clears the Command Line window.
HELP	HE	Displays help for Command Line or help on a command.
LOG	LO	Controls command output logging.
SLEEP	-	Delays command execution.
SUBMIT	SU	Executes a file of commands.
TCL	-	Turns TCL commands on or off.

Test Supporting Facilities Commands

Command Name	Abbreviation	Description
CLOSE_TEST_SUITE	CTS	Closes the current test suite.
COMPARE_TEST_DATA	CTD	Compares test data and create results.
OPEN_TEST_SUITE	OTS	Opens a test suite.
RUN_TEST	RT	Runs a test.

Execution Commands (Available when the debugger is connected)

Command Name	Abbreviation	Description
GO	GO	Runs program.
GO_RESET	GR	Runs program from reset.
GO_TILL	GT	Runs program until specified addresses.
HALT	HA	Halts program.
INITIALIZE	IN	Initializes the debugging platform system.
RESET	RE	Resets the microprocessor.
STEP	ST	Steps through program (by instructions or source lines).
STEP_MODE	SM	Sets the step mode
STEP_OUT	SP	Steps out of the current function.
STEP_OVER	SO	Steps through program without stepping into functions.
STEP_RATE	SR	Sets rate of stepping.

Memory Operation Commands (Available when the debugger is connected)

Command Name	Abbreviation	Description
CACHE ***1**	-	Sets caching on or off.
FILE_LOAD	FL	Loads an object (program) file.
FILE_LOAD_ALL	LA	Loads all object (program) files.
FILE_SAVE	FS	Saves memory to a file.
FILE_UNLOAD	FU	Unloads an object file from memory.
FILE_UNLOAD_ALL	UA	Unloads all object (program) files from memory.
FILE_VERIFY ***1**	FV	Verifies file contents against memory.
MEMORY_COMPARE ***1**	MC	Compares memory contents.
MEMORY_DISPLAY	MD	Displays memory contents.
MEMORY_EDIT	ME	Modifies memory contents.
MEMORY_FILL	MF	Fills a block of memory.
MEMORY_FIND ***1**	MI	Finds a string in an area of memory.
MEMORY_MOVE	MV	Moves a block of memory.
MEMORY_TEST ***1**	MT	Tests a block of memory.

***1**. Support for this command depends on the debugging platform.

Software Break Setting Commands (Available when the debugger is connected)

Command Name	Abbreviation	Description
SET_DISASSEMBLY_SOFT_BREAK	SDB	Sets or deletes a software breakpoint at the disassembly level.
SET_SOURCE_SOFT_BREAK	SSB	Sets or deletes a software breakpoint at the source level.
STATE_DISASSEMBLY_SOFT_BREAK	TDB	Enables or disables a software breakpoint at the disassembly level.
STATE_SOURCE_SOFT_BREAK	TSB	Enables or disables a software breakpoint at the source level.

Other Debugging Commands (Available when the debugger is connected)

Command Name	Abbreviation	Description
CONNECT ***1**	CN	Connects the debugging platform.
DISCONNECT ***1**	DN	Disconnects the debugging platform.
INITIALIZE	IN	Initializes the debugging platform system.

***1**. Support for this command depends on the debugging platform.

For the syntax of each command, refer to the online help.

4. Regular Expressions

The HEW editor allows you to include special characters in search strings when performing a find, replace or find in files operation.

These characters are listed in the table below and explained underneath.

Character	Function
?	This character matches any single character, except the newline character. For example, t?p matches “top”, “tip” but not “trap”.
*	This character matches any number of occurrences (0 or more) of any character except a newline. Thus, this character will not match across new lines. The * character will match as few occurrences as are necessary to make the rest of the pattern match. For example, t*o matches the “to” of “too”, the “tro” of “trowel” and the “ty o” of “sporty orange” but not “smart orange” because the * character does not match across a new line.
\n	This character matches the newline character. \n would be used to search for line endings or in patterns that cross line boundaries. Example 1: ;\n matches every occurrence of a newline following a semicolon Example 2: ;\nif searches for a semicolon, a new line and a line beginning with “if”.
\t	This character matches the tab character. Example 1: \t8 Finds every occurrence of a tab character followed by an 8. Example 2: init\t Finds every occurrence of a tab character following “init”.
[ ]	This matches any one character or a range of single characters listed within the brackets. Brackets cannot be nested. [-] specifies a range of characters e.g. [a-z] or [0-9]. The beginning character in the range must have a lower ASCII value than the ending character of the range. [~] matches a single character if it is not any one of the characters between [~ and]. This pattern also matches newline characters, unless the newline character is included within the brackets. Example 1: [AEIOU] Finds every uppercase vowel. Example 2: [<>?] Finds a literal <, > or ?. Example 3: [A-Za-z0-9_] Matches an upper or lowercase letter, a digit or an underscore. Example 4: [~0-9] Matches any character except a digit. Example 5: [\t\n] Matches a space, a tab or newline. Example 6: [\]] Matches a literal] if] is placed after \.
\	This is the regular expression override character. If the character following the backslash is a regular expression character, it is treated as a normal character. The backslash is ignored if it is followed by a normal (non-regular expression) character. Example 1: * Searches for every occurrence of an asterisk. Example 2: \\ Searches for every occurrence of a backslash.

5. Placeholders

This section describes how to use the placeholders, a feature provided by several of the High-performance Embedded Workshop components.

5.1 What is a placeholder?

A placeholder is a special string, inserted into text, which is replaced at some subsequent time for the actual value. For example, one of the HEW placeholders is $(FULLFILE) which represents a file with a full path.

Suppose that you have an editor in c:\myedit\myeditor.exe, which can accept the file to be edited as a parameter. When invoking the editor (for example, you may want to open the file 'FILE1.C' from the directory 'c:\files'), the following shortcut could be made:

```
c:\myedit\myeditor.exe c:\files\FILE1.C
```

However, what happens if you want to open any file through this editor? The problem is that the command above is specific to 'c:\files\file1.c'. What we want to be able to do is to tell the HEW to use the editor specified but to open the file that we have chosen at that time. To do this, you can substitute the specific name of the file for a general **Placeholder**:

```
c:\myedit\myeditor.exe $(FULLFILE)
```

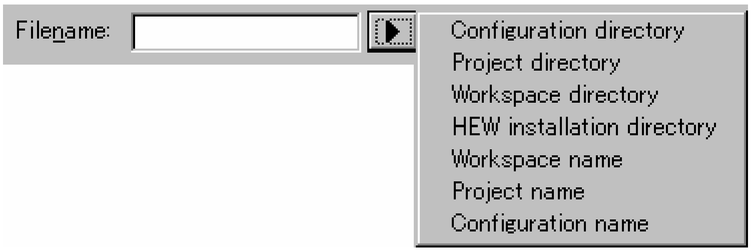
Now whenever the HEW launches the editor with a file, it knows that it has to replace the $(FULLFILE) placeholder with the file you have selected.

5.2 Inserting a placeholder

To insert a placeholder, select in any of the following operations.

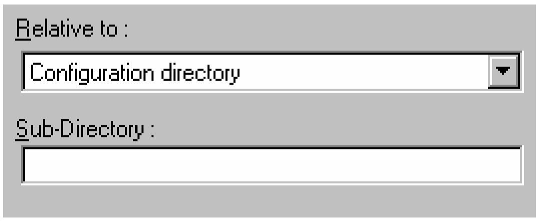
Example 1

1. Place the insertion cursor at the point you would like to insert the placeholder.
2. Click the placeholder button. A pop-up menu will be displayed which lists all of the placeholders that are valid for the associated edit box.
3. Select the desired placeholder from the pop-up menu. The equivalent placeholder will be inserted into the edit box.



Example 2

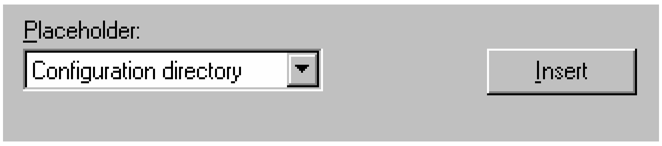
1. Select the required placeholder other than "Custom directory" from the drop-down list box and specify a sub-directory relative to the directory shown by the placeholder.
2. If you select "Custom directory", specify an absolute directory path in the **Sub-Directory** field.


Relative to : Configuration directory

Sub-Directory :

Example 3

1. Place the insertion cursor at the point you would like to insert the placeholder
2. Select the required placeholder from the drop-down list box.
3. Click **Insert**.


Placeholder: Configuration directory

Example 4

1. Alternatively, if you know the placeholder already, type it into the field directly. Ensure that you type the placeholder name in uppercase and that it is preceded by ‘$(’ and followed by ‘)’.

This is correct:

$(FILEDIR)

These are incorrect:

$(Filedir)

$(FILEDIR)

$FILEDIR

5.3 Available placeholders

The table below lists the available placeholders and their meanings, along with an example of their use.

Placeholder	Meaning	Expanded placeholder (example)
$(FULLFILE)	Filename (including full path)	c:\hew\workspace\project\file.src
$(FILEDIR)	File directory	c:\hew\workspace\project
$(FILENAME)	Filename (excluding path, including extension)	file.src
$(FILELEAF)	Filename (excluding path and extension)	file
$(EXTENSION)	File extension	src
$(WORKSPDIR)	Workspace directory	c:\hew\workspace
$(WORKSPNAME)	Workspace name	workspace
$(PROJDIR)	Project directory	c:\hew\workspace\project
$(PROJECTNAME)	Project name	project
$(CONFIGDIR)	Configuration directory	c:\hew\workspace\project\debug
$(CONFIGNAME)	Configuration name	debug
$(HEWDIR)	HEW installation directory	c:\hew
$(TCINSTALL)	Toolchain install directory (on the options setting dialogs and the New Build Phase dialog (step 3))	c:\hew\Tools\Renesas\Sh\9_0_1_1
$(TOOLDIR)	Tool installation directory (Tools Administration)	c:\hew\Tools\Renesas\Sh\9_0_1_1
$(TEMPDIR)	Temp directory	c:\temp
$(WINDIR)	Windows® directory	c:\windows
$(WINSYSDIR)	Windows® system directory	c:\windows\system
$(EXEDIR)	Command directory	v:\vc\win32
$(USERNAME)	User login (Version control)	JHARK
$(PASSWORD)	User password (Version control)	214436
$(VCDIR)	“Virtual” version control directory	“c:\project” is mapped to “x:\vc\project”.
$(COMMENT)	Comment (Version control)	“Please Enter Comment” dialog is invoked.
$(LINE)	Line number of an error/warning	12

In the table above, we are assuming that:

- a file path is “c:\hew\workspace\project\file.src”.
- a workspace named “workspace” is located at “c:\hew\workspace”.
- a project named “project” is located at “c:\hew\workspace\project”.
- a configuration named “debug” has a configuration directory located at “c:\hew\workspace\project\debug”.
- HEW.EXE is installed in “c:\hew”.
- a *.HRF file of a toolchain (i.e. compiler, assembler, linker) is located in “c:\hew\Tools\Renesas\Sh\9_0_1_1”. This is referred to as $(TCINSTALL) on the options setting dialogs and the New Build Phase dialog (step 3) of the Build menu and as $(TOOLDIR) on the **Tools Administration** dialog box.
- the Windows® operating system is installed in “c:\windows” and the Windows® system directory is located at “c:\windows\system”.
- a version control executable path is “v:\vc\win32\ss.exe”; a user name and its password to login to the version control system are “JHARK” and “214436” respectively; $(COMMENT) is specified in a command line to the version control executable; “c:\project” is mapped to “x:\vc\project” on the **Projects** tab of the **Version Control Setup** dialog box, which is invoked via **Tools->Version Control->Configure**.
- an error of compiler or assembler occurred at line 12.

Note:

Not all of the placeholders are relevant in every field. For example, the $(LINE) placeholder has no meaning when specifying a dependent file’s location. $(USERNAME), $(PASSWORD), $(VCDIR), and $(COMMENT) placeholders are acceptable only in version control. If you enter a placeholder into an edit field where it is not acceptable you will be informed.

5.4 Placeholder tips

Placeholders are there to allow you to create flexible paths to the various files used by the system.

- If there is a placeholder pop-up menu (▶) next to an edit field into which you are about to enter a path or file, you should consider how you can use a placeholder to make that path or file definition flexible.
- If you use several configurations, then the $(CONFIGDIR) placeholder is very useful to ensure that files can be written to and from the current configuration's directory.
- Wherever possible, use a placeholder. They can always be removed or added later so don't be afraid to experiment.

6. I/O File Format

HEW formats the **IO** window based on information it finds in an I/O Register definition file. When you select a debugging platform, HEW will look for a “<*device*>.IO” file corresponding to the selected device and load it if it exists. This file is a formatted text file that describes the I/O modules and the address and size of their registers. You can edit this file, with a text editor, to add support for memory mapped registers or peripherals you may have specific to your application e.g. registers in an ASIC device mapped into the microcomputer's address space.

File format

Each module name must be defined in the **Modules** definition section and the numbering of each module must be sequential. Each module corresponds to a register definition section and within the section each entry defines an I/O register.

The **BaseAddress** definition is for devices where the location of I/O registers moves in the address space depending on the CPU mode. In this case, the **BaseAddress** value is the base address of the I/O registers in one specific mode and the addresses used in the register definitions are the address locations of the registers in the same mode. When the I/O register file is actually used, the **BaseAddress** value is subtracted from the defined register address and the resultant offset added to the relevant base address for the selected mode.

Each module has a section that defines the registers forming it along with an optional dependency, the dependency is checked to see if the module is enabled or not. Each register name must be defined in the section and the numbering of each register must be sequential. The dependency is entered in the section as dep=<reg> <bit> <value>.

1. <reg> is the register id of the dependency.
2. <bit> is the bit position within the register.
3. <value> is the value that the bit must be for the module to be enabled.

The [Register] definition entry is entered in the format id=<name> <address> [<size> [<absolute>[<format>[<bitfields>]]]].

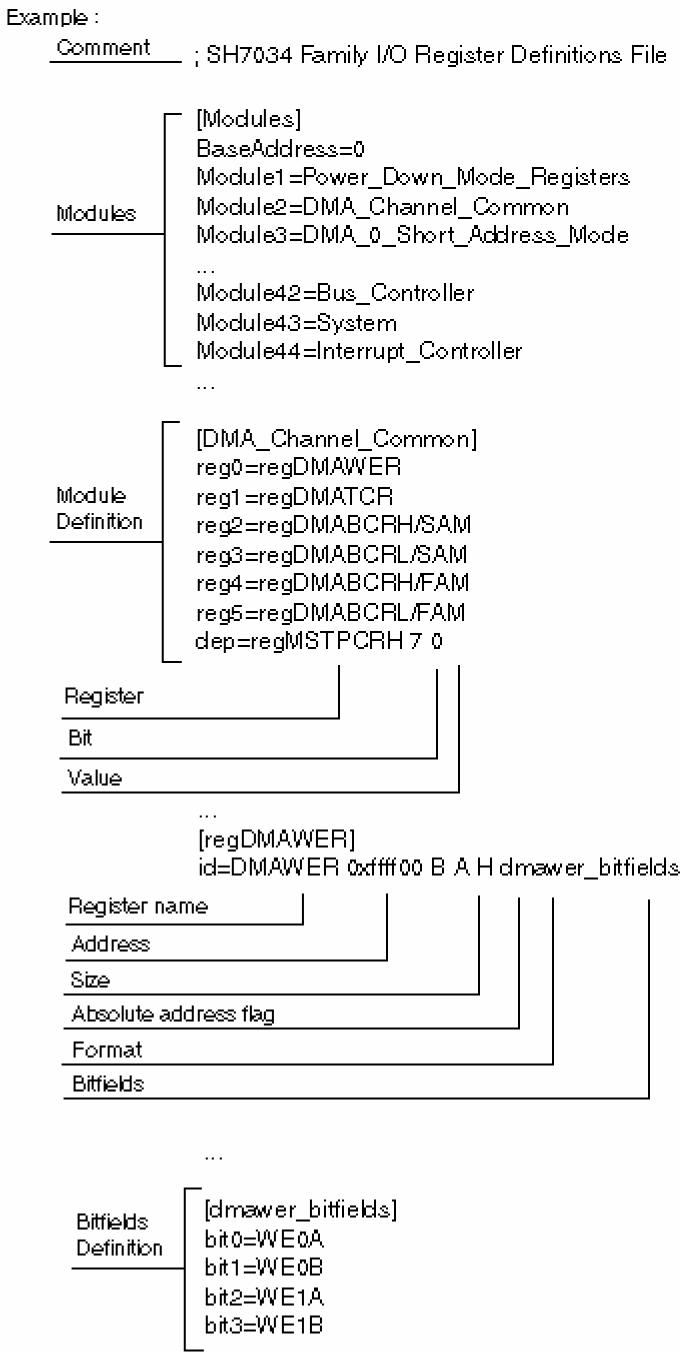
1. <name> register name to be displayed.
2. <address> address of the register.
3. <size> which may be B, W or L for byte, word, or long word (default is byte).
4. <absolute> which can be set to A if the register is at an absolute address. This is only relevant if the I/O area address range moves about on the CPU in different modes. In this case, if a register is defined as absolute the base address offset calculation is not performed and the specified address is used directly.
5. <format> Format for register output. Valid values are H for Hexadecimal, D for decimal, and B for binary.
6. <bitfields> section defining the bits within the register.

Bitfield sections define the bits within a register each entry is of the type bit<no>=<name>.

1. <no> is the bit number.
2. <name> is a symbolic name of the bit.

Comment lines are allowed and must start with a “;” character.

Example :



```
; SH7034 Family I/O Register Definitions File

[Modules]
BaseAddress=0
Module1=Power_Down_Mode_Registers
Module2=DMA_Channel_Common
Module3=DMA_0_Short_Address_Mode
...
Module42=Bus_Controller
Module43=System
Module44=Interrupt_Controller
...

[DMA_Channel_Common]
reg0=regDMAWER
reg1=regDMATCR
reg2=regDMABCRH/SAM
reg3=regDMABCRL/SAM
reg4=regDMABCRH/FAM
reg5=regDMABCRL/FAM
dep=regMSTPCRH 7 0

...
[regDMAWER]
id=DMAWER 0xffff00 B A H dmawer_bitfields

...
[dmawer_bitfields]
bit0=WE0A
bit1=WE0B
bit2=WE1A
bit3=WE1B
```

7. Symbol File Format

In order for HEW to be able to understand and decode the symbol file correctly, the file must be formatted as a Pentica-B file:

1. The file must be a plain ASCII text file.
2. The file must start with the word “BEGIN”.
3. Each symbol must be on a separate line with the value first, in hexadecimal terminated by an “H”, followed by a space then the symbol text.
4. The file must end with the word “END”.

Example:

```
BEGIN
11FAH Symbol_name_1
11FCH Symbol_name_2
11FEH Symbol_name_3
1200H Symbol_name_4
END
```

Note:

Support for this function depends on the debugging platform.

8. Keyboard Shortcuts

All major commands in the HEW application can be driven by the keyboard. Below is a list of all keyboard commands in the application.

Function key	Key	Function
ALT	0-9	Play a macro assigned to keyboard shortcut.
ALT	K	Open the workspace window.
ALT	O	Open the output window.
ALT	F4	Exit the application.
ALT	BACKSPACE	Undo (alternative in the editor to CTRL+Z)
CTRL	0-9	Reserved for use for template insertion.
CTRL	A	Select all in the editor. May also work in other windows.
CTRL	B	Open the Breakpoints dialog box.
CTRL	C	Copy.
CTRL	D	Open the disassembly window.
CTRL	F	Open the Find dialog box.
CTRL	G	Go to source line.
CTRL	H	Replace.
CTRL	I	Open the IO window.
CTRL	K	Open the stack trace window.
CTRL	L	Open the command line window.
CTRL	M	Open the memory window.
CTRL	N	Create a new source file in the editor.
CTRL	O	Open the open file dialog box.
CTRL	P	Print.
CTRL	R	Open the registers window.
CTRL	S	Save the current file.
CTRL	U	Open the status window.
CTRL	V	Paste.
CTRL	X	Cut.
CTRL	Y	Redo.
CTRL	Z	Undo.
CTRL	F2	Toggle bookmark.
CTRL	F3	Search for the next occurrence of the text chosen in the editor or open the Find dialog box.
CTRL	F4	Close the editor window.
CTRL	F6	Switch windows in the editor.
CTRL	F7	Build file.
CTRL	F9	Enable or disable a breakpoint on the current line.
CTRL	BACKSPACE	Delete current word.
CTRL	BREAK	Stop build.
CTRL	INSERT	Copy.
CTRL	TAB	Switch windows in the editor.
CTRL	HOME	Returns the cursor in the editor to the beginning of the current file.
CTRL	END	Sends the cursor in the editor to the end of the current file.
CTRL	UP ARROW	Scroll window up and leave the cursor in the same place in the editor.
CTRL	DOWN ARROW	Scroll window down and leave the cursor in the same place in the editor.
CTRL	LEFT ARROW	Move the cursor to the previous word in the editor.
CTRL	RIGHT ARROW	Move the cursor to the next word in the editor.
CTRL+ALT	1-4	Switch virtual desktop.
CTRL+ALT	PAGE UP	Move to next tab. For example output window.
CTRL+ALT	PAGE DOWN	Move to previous tab. For example output window.
CTRL+SHIFT	8	Show white space characters in the editor.
CTRL+SHIFT	G	Open image window.
CTRL+SHIFT	K	Open TCL toolkit.
CTRL+SHIFT	L	Line deletion in editor.
CTRL+SHIFT	M	Match braces.
CTRL+SHIFT	S	Save all.
CTRL+SHIFT	T	Insert template.
CTRL+SHIFT	U	Change into a uppercase letter all the texts chosen in the editor.
CTRL+SHIFT	V	Open waveform window.
CTRL+SHIFT	Y	Display PC position.

CTRL+SHIFT	F3	Search for the previous occurrence of the text chosen in the editor or open the Find dialog box.
CTRL+SHIFT	TAB	Switch windows in the editor.
CTRL+SHIFT	HOME	Select from the cursor to the beginning of the file.
CTRL+SHIFT	END	Select from the cursor to the end of the file.
CTRL+SHIFT	LEFT ARROW	Select the previous word in the editor.
CTRL+SHIFT	RIGHT ARROW	Select the next word in the editor.
SHIFT	F1	Show context sensitive help.
SHIFT	F2	Previous bookmark.
SHIFT	F3	Search for the previous occurrence of the specified text.
SHIFT	F5	Reset Go.
SHIFT	F8	Displays the editor that generated the previous build error or warning.
SHIFT	F10	Show context menu. Same as right-click pop-up menu.
SHIFT	F11	Step out.
SHIFT	F12	Refresh all windows.
SHIFT	DELETE	Editor cut operation. (alternative in the editor to CTRL+X)
SHIFT	ESC	Hide output window.
SHIFT	TAB	Move the tab back in the editor.
SHIFT	HOME	Selects from the cursor to the beginning of the current line in the editor.
SHIFT	END	Selects from the cursor to the end of the current line in the editor.
SHIFT	UP ARROW	Select the next line up.
SHIFT	DOWN ARROW	Select the next line down.
SHIFT+ALT	BACKSPACE	Redo (alternative in the editor to CTRL+Y)
None	F1	Context sensitive help.
None	F2	Next bookmark.
None	F3	Search for the next of the specified text.
None	F4	Find in files.
None	F5	Go.
None	F7	Build.
None	F8	Display the editor that generated the next build error or warning.
None	F9	Set or remove a breakpoint on the current line.
None	F10	Step over.
None	F11	Step in.
None	F12	Refresh window.
None	DELETE	Clear.
None	INSERT	Toggle insert and overwrite mode.
None	ENTER	Carriage return in the editor.
None	TAB	Insert a tab in the editor.
None	ESC	Halt.
None	HOME	Move the cursor to the beginning of the current line in the editor.
None	END	Move cursor to the end of the current line in the editor.
None	PAGE UP	Move the page in the editor up.
None	PAGE DOWN	Move the page in the editor down.
None	UP ARROW	Move cursor up in the editor.
None	DOWN ARROW	Move cursor down in the editor.
None	LEFT ARROW	Move cursor left in the editor.
None	RIGHT ARROW	Move cursor right in the editor.
None	ALT + Mouse select	Select column in the editor.

Note:

Support for this function depends on the debugging platform.

9. Drag and Drop in the HEW Debugger

When using the HEW debugger it is possible for each debug component to interact with the others. This can be achieved simply by dragging objects from one view to another.

Some examples are listed below

1. It is possible to drag a label from the labels view onto other debug views. So for example if you drag a label onto the disassembly window it will scroll to the address that the label is located at.
2. It is possible to drag a watch variable from the editor into the watch window. This adds the watch variable to the window.
3. Dragging a function name from the editor into the disassembly should jump the disassembly view to the label location.

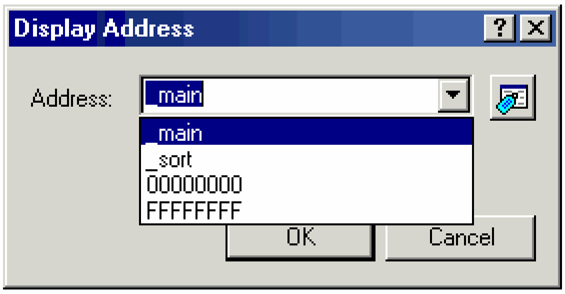
10. Using Labels to View Your Code

Labels are a useful way of navigating through your debug module. It is possible to use labels in any edit field that allows addresses. If you enter a label in such a field then the built in evaluator will check the label and then convert it to an address. This allows you to enter evaluations such as "_main+100" or "_MyFunction+100".

This means that any times that labels are used the addresses which will be evaluated are not fixed. This is especially useful if you are using a command line batch file to set a number of breakpoints. The command line batch file might always need to set a breakpoint on a certain function and this can be achieved by using a label.

Using the label allows the code to change without affecting the batch file contents.

HEW V.3.1 onwards also supports a number of easy ways to use label completion.



Display Address

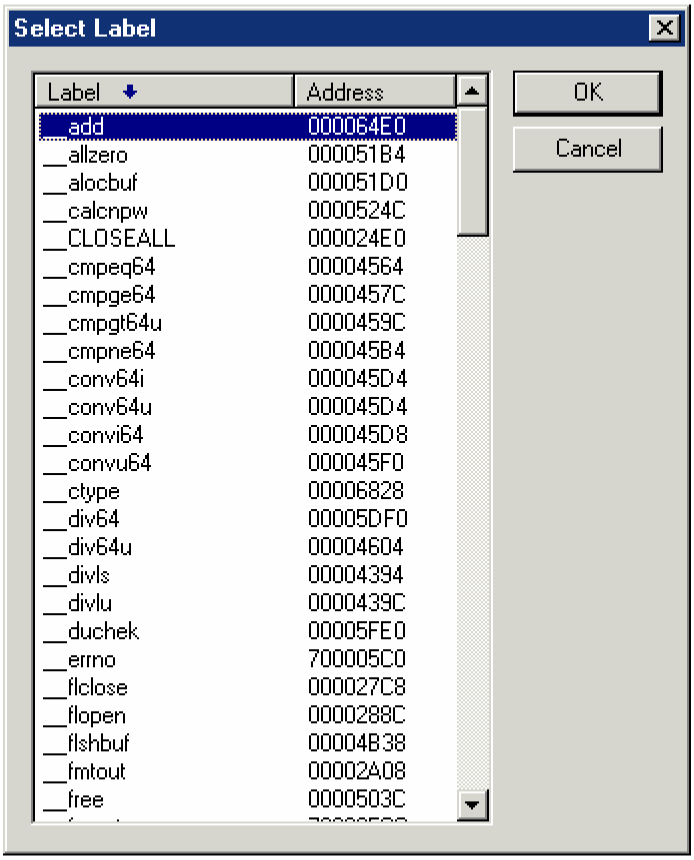
Address: _main

_main
_sort
00000000
FFFFFFFF

The dialog box above also shows an example of a label pick list. This store the last twenty entries made into address fields throughout the entire HEW application. This means if you are entering a label multiple times it should be much faster and efficient if you use this recently used address field list. This control is available for all instances of the address edit field where the input is evaluated.

There is the browse button () right next to the edit field where an address can be entered.

If you click this button, the **Select Label** dialog box will be displayed. A label can be selected from the current list of labels in this dialog box.



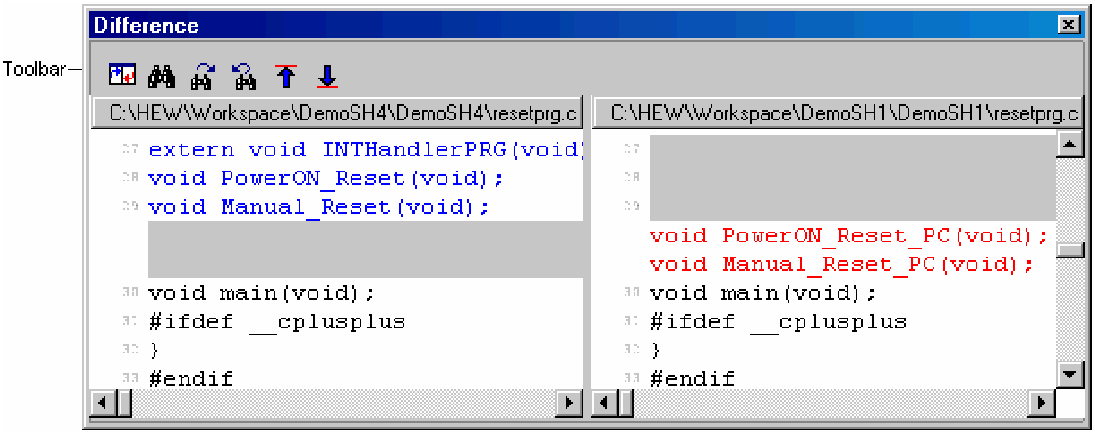
Label	Address
__add	000064E0
__allzero	000051B4
__alocbuf	000051D0
__calcnpw	0000524C
__CLOSEALL	000024E0
__cmpeq64	00004564
__cmpge64	0000457C
__cmpgt64u	0000459C
__cmpne64	000045B4
__conv64i	000045D4
__conv64u	000045D4
__convi64	000045D8
__convu64	000045F0
__ctype	00006828
__div64	00005DF0
__div64u	00004604
__divls	00004394
__divlu	0000439C
__duchek	00005FE0
__errno	700005C0
__flclose	000027C8
__flopen	0000288C
__flshbuf	00004B38
__fmtout	00002A08
__free	0000503C

In this dialog box, the labels are initially listed in alphabetical order and their addresses are displayed on the right. If you click a column header (of names or addresses), the labels will be sorted by the label name or the address value.

11. Integrated Toolbars in a Components View

The HEW 4.0 onwards has the capability to include a toolbar in a views client area. This toolbar allows the views functionality to be accessed quickly from this integrated toolbar.

Various views in the HEW system have this functionality. One example is the **Difference** view. This is shown below:

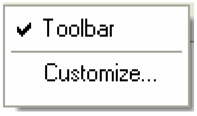

Difference
Toolbar
C:\HEW\Workspace\DemoSH4\DemoSH4\resetprg.c
C:\HEW\Workspace\DemoSH1\DemoSH1\resetprg.c

```
extern void INTHandlerPRG(void)
void PowerON_Reset(void);
void Manual_Reset(void);

void main(void);
#ifdef __cplusplus
}
#endif
```

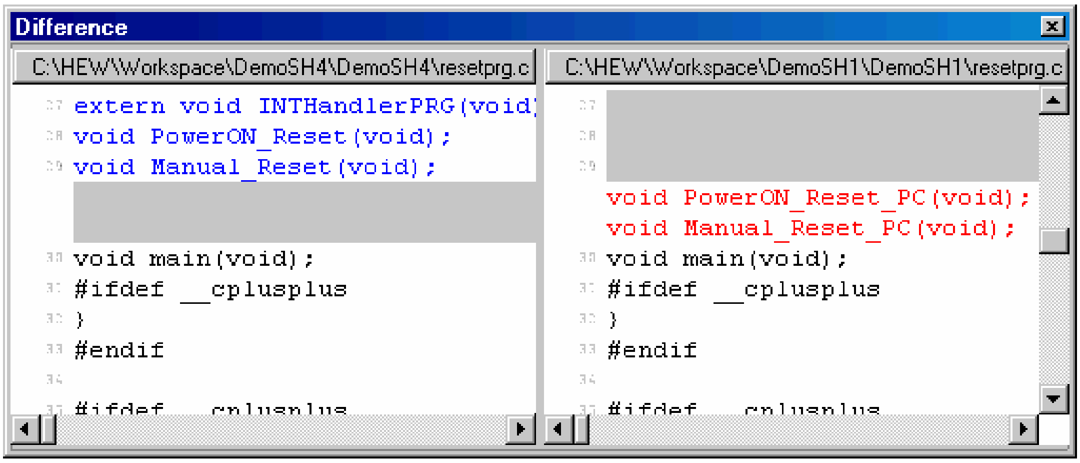
```


void PowerON_Reset_PC(void);
void Manual_Reset_PC(void);
void main(void);
#ifdef __cplusplus
}
#endif
```

The toolbar allows access to key Difference view features.

It is also possible to customize the toolbar further. This can be achieved via the pop-up menu of the toolbar or the component itself. If you right click on the toolbar itself the following menu is displayed:



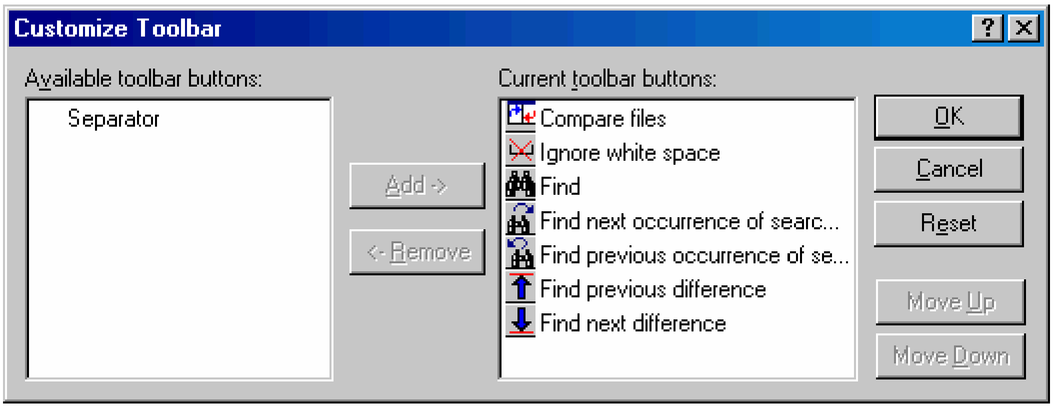
The top menu option named "Toolbar" switches showing/hiding of the toolbar (the toolbar is hidden in the figure below).


Difference
C:\HEW\Workspace\DemoSH4\DemoSH4\resetprg.c
C:\HEW\Workspace\DemoSH1\DemoSH1\resetprg.c

```
extern void INTHandlerPRG(void)
void PowerON_Reset(void);
void Manual_Reset(void);

void main(void);
#ifdef __cplusplus
}
#endif

#ifdef __cplusplus
```

```


void PowerON_Reset_PC(void);
void Manual_Reset_PC(void);
void main(void);
#ifdef __cplusplus
}
#endif

#ifdef __cplusplus
```

The final menu option named "Customize" launches the **Customize Toolbar** dialog box. This is shown below:



This dialog box allows you to modify the displayed buttons and change the ordering. The list on the left named **Available toolbar buttons** displays all toolbar buttons not currently in use on the toolbar. The list named **Current toolbar buttons** displays all of the toolbars currently added to the components' toolbar.

To add the currently selected buttons to a toolbar

1. Select the toolbar button you wish to add from the **Available toolbar buttons** list.
2. Press the **Add** button.
3. Click OK.

To move the currently selected buttons

1. Select the toolbar you wish to move in the **Current toolbar buttons** list.
2. Click **Move Up** or **Move Down** until it is in the desired position.
3. Click OK.

To remove the currently selected buttons from a toolbar

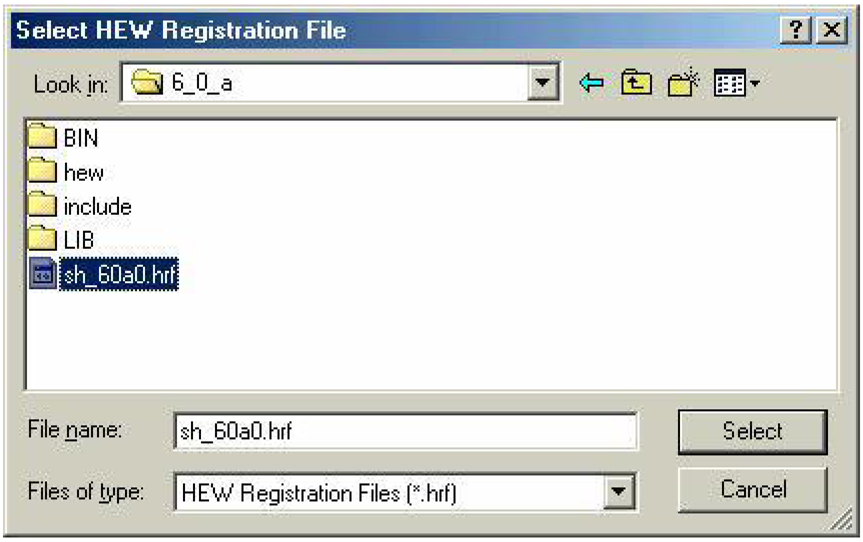
1. Select the toolbar button you wish to remove from the **Current toolbar buttons** list.
2. Press the **Remove** button. The toolbar is added to the **Available toolbar buttons** list.
3. Click OK.

12. To Build in Toolchain for HEW1.x

When a project created in HEW1.x is used without upgrading to new toolchain that has been registered in HEW2.x onwards, the toolchain for the old version must be registered. Select the 'HRF' file for the old toolchain with the **Register** button by selecting [**Tools–>Administration**]. Build can be executed on HEW2.x onwards by using the old toolchain.

However, note that a workspace which has been opened in HEW2.x onwards cannot be opened in HEW1.x.

In HEW2.x onwards, a new project for the old version cannot be created. When a project for the old version in HEW1.x is created, use HEW1.x.



File name: sh_60a0.hrf

Files of type: HEW Registration Files (*.hrf)

13. HMAKE User Guide

13.1 Command line

The following section describes the command line that should be used to execute the hmake program on a file using none or more of the available options.

Basic structure

The command line must be of the following syntax:

```
hmake <make file you wish to execute> <parameter list>
```

If a file is specified without an extension then ".mak" will be appended to it. The parameter list may include none or more of the parameters listed in the following section. The parameters list may appear before the make file name if you wish. Each parameter must be separated by at least one white space character. Parameters are not case sensitive. If no parameters are given and no file is given then help information will be displayed.

Exit codes

If there are any syntax errors in the make file being executed or if any process executed whilst running the make file returns an invalid error code then hmake will exit with code 1. Otherwise hmake will exit with code 0 (See below for file syntax and how to specify exit code conditions).

Parameters

The following table shows the available parameters and their function:

Parameter	Function
/A	Execute all commands regardless of input/output file status. Equivalent to a Build All.
/N	Use status of input/output files to calculate what commands need to be executed (as normal) and then display the commands but do not execute them.
/?	Displays help info.

13.2 File syntax

There are four basic types of statement used in a hmake file, the variable declaration the description block, the comment and the message command. These can be combined in any order to produce a hmake file but a variable must be declared in a variable declaration before it is used in a description block or other variable declaration. The first "all" statement used in nmake files is not required in a hmake file. Commands are executed in order, as they appear in the make file.

Note: the "→" character is used to show were a tab character must be used in order to keep the make file syntactically correct.

Variable declarations

A variable declaration declares a variable which can then be used in any statement throughout the rest of the hmake file. A declaration has the following syntax:

```
<variable name> = <value>
```

Any number of white space characters are allowed between the variable name and the '=' sign and the value and the '=' sign. The value may be split over several lines using a '\' character. If the value contains '\' characters within the main text then these are taken literally. Only '\' characters followed by a new line are considered to indicate a value wrapping over more than one line.

There follows some examples of valid variable declarations:

```
EXECUTABLE = c:\dir\prog.exe
OUTPUT = c:\dir2\file1.out
```

```
INPUT = c:\dir2\file1.c
DEPEND = c:\dir2\file2.h \
       c:\dir2\file3.h \
       c:\dir2\file4.h
```

In order to use a variable later in the hmake file write the variable name with "$(" added to the front and ")" added to the back. The variable name (along with the "$()" characters) will be substituted with the variables value. For examples of this see later under description blocks. Only alphanumeric characters and underscore characters are allowed in variable names. It is possible to use a variable inside the declaration of a different variable but all variables must be declared before they are used.

13.3 Description blocks

Basic outline

A description block specifies one or more targets, zero or more dependants and a list of commands which should be executed if the newest dependent is newer than the newest target. If none of the targets exist and/or none of the dependants exist then the commands will always be executed. It is not necessary to specify any dependants if you wish the commands to always be executed. A description block has the following syntax:

```
<target1> <target2> … : <dependant1> <dependant2> …
→ <command1>
→ <command2>
→ …
→ <commandn>
```

Any number of white space characters are allowed between the last target and the ':' character and the first dependant and the ':' character. No white space is allowed before the first target. Each target and each dependant must be separated by at least one white space character. A tab character must be present at the start of a line containing a command. Variables may be used in a description block using the syntax specified above under variable declarations.

There follows some examples of valid description blocks (one of which uses the variable specified above under variable declarations):

```
c:\dir1\file1.obj : c:\dir1\file1.c c:\dir1\file1.h
→ gcc c:\dir1\file1.c
$(OUTPUT) : $(INPUT) $(DEPEND)
→ $(EXECUTABLE) $(INPUT)
```

Special commands

There are two special commands which can be used in a description block. The "cd" command changes the current directory and the "set" command sets an environment variable which will then be in use for the duration of the make file execution. Both are used in the same way as the DOS equivalents.

There follows some examples of valid description blocks which use these commands:

```
CHANGEDIR :
→ cd c:\dir1\dir2
SETENV:
→ set VAR1=value1
→ set VAR2=value2
→ set VAR3=value3
```

It does not matter that CHANGEDIR and SETENV are not file names. They will be treated as files that do not exist and so the commands will always be executed.

Sub command files

If you wish hmake to generate a sub command file for you then the command part of the description block should be specified as follows (this replaces <commandn> above):

```
→ <command start> <<
→ <sub command1>,
→ <sub command2>,
→ …
→ <sub commandn>
<<<command end>
```

This will generate a sub command file, in the windows temporary directory, which will contain the lines <sub command1>, <sub command2> etc. This command file will be deleted once the make process has completed. The name of the command file will be substituted for all the text between the two "<<"'s. You do not have to worry about the name of the sub command file. This is generated by hmake.

For example:

```
c:\dir1\file1.obj : c:\dir1\file1.c c:\dir1\file1.h
→ gcc @"<<
→ -c -o c:\dir1\file1.obj c:\dir1\file1.c
<<"
```

If the sub command file generated has the name "c:\temp\hmk111.cmd" then the following would be executed by hmake (assuming c:\dir1\file1.obj is out of date):

```
gcc @"c:\temp\hmk111.cmd"
```

The command file (c:\temp\hmk111.cmd) would contain:

```
-c -o c:\dir1\file1.obj c:\dir1\file1.c
```

It is possible to include more than one command in the description block and to use combinations of the standard, and sub command file commands.

13.4 Comments

A '#' character signifies a comment. When this character appears as the first character on a line the rest of the line (up until the next new line character) is ignored. There follows examples of valid comments:

```
# My hmake file
# Variable declaration
OUTPUT= c:\dir1\file1.obj
# Descriptor
$(OUTPUT) : c:\dir1\file1.c c:\dir1\file1.h
→ set VAR1=value1
→ gcc c:\dir1\file1.c
```

A comment must occupy its own line in the hmake file. It is not possible to put comments on the end of other statements.

13.5 Message commands

The message command is used to output a line of text to standard out whilst a make file is executing. These text lines will be output in the order they appear in the make file, in amongst output from any executables being executed as appropriate. No buffering of output text will take place. A message command has the following syntax:

```
!MESSAGE <text to output>
```

A new line character is assumed to come after the last character in <text to output>. Any white space between !MESSAGE and <text to output> will be ignored. There follows an example of a valid message command:

```
!MESSAGE Executing C Compiler
```

High-performance Embedded Workshop V.4.03
User's Manual

Publication Date:	Apr. 20, 2007	Rev.1.00
Published by:	Sales Strategic Planning Div. Renesas Technology Corp.	
Edited by:	Microcomputer Tool Development Department Renesas Solutions Corp.	

High-performance Embedded Workshop V.4.03
User's Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan