

# User's Manual

# ID850NW

Integrated Debugger Ver. 1.10 or Later

Operation (Windows™ Based)

---

Target Devices  
V850 Family™

[MEMO]

**V800 Series, V850 Family, V850/SA1, V850/SB1, V850/SB2, V850/SV1, V850E/MS1, V850E/MA1, V850E/IA1, V851, V852, V853, and V854 are trademarks of NEC Corporation.**

**UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.**

**Pentium is a trademark of Intel Corporation.**

**Windows, Windows NT, and MS-DOS are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.**

**PC/AT is a trademark of International Business Machines Corporation.**

- **The information in this document is current as of August, 2000. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC's data sheets or data books, etc., for the most up-to-date specifications of NEC semiconductor products. Not all products and/or types are available in every country. Please check with an NEC sales representative for availability and additional information.**
  - No part of this document may be copied or reproduced in any form or by any means without prior written consent of NEC. NEC assumes no responsibility for any errors that may appear in this document.
  - NEC 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 NEC semiconductor products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC or others.
  - Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of customer's equipment shall be done under the full responsibility of customer. NEC assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
  - While NEC endeavours to enhance the quality, reliability and safety of NEC semiconductor products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC semiconductor products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment, and anti-failure features.
  - NEC semiconductor products are classified into the following three quality grades:  
 "Standard", "Special" and "Specific". The "Specific" quality grade applies only to semiconductor products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of a semiconductor product depend on its quality grade, as indicated below. Customers must check the quality grade of each semiconductor product before using it in a particular application.
    - "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
    - "Special": 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, life support systems and medical equipment for life support, etc.
- The quality grade of NEC semiconductor products is "Standard" unless otherwise expressly specified in NEC's data sheets or data books, etc. If customers wish to use NEC semiconductor products in applications not intended by NEC, they must contact an NEC sales representative in advance to determine NEC's willingness to support a given application.
- (Note)
- (1) "NEC" as used in this statement means NEC Corporation and also includes its majority-owned subsidiaries.
  - (2) "NEC semiconductor products" means any semiconductor product developed or manufactured by or for NEC (as defined above).

M8E 00.4

# Regional Information

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

## **NEC Electronics Inc. (U.S.)**

Santa Clara, California  
Tel: 408-588-6000  
800-366-9782  
Fax: 408-588-6130  
800-729-9288

## **NEC Electronics (Germany) GmbH**

Duesseldorf, Germany  
Tel: 0211-65 03 02  
Fax: 0211-65 03 490

## **NEC Electronics (UK) Ltd.**

Milton Keynes, UK  
Tel: 01908-691-133  
Fax: 01908-670-290

## **NEC Electronics Italiana s.r.l.**

Milano, Italy  
Tel: 02-66 75 41  
Fax: 02-66 75 42 99

## **NEC Electronics (Germany) GmbH**

Benelux Office  
Eindhoven, The Netherlands  
Tel: 040-2445845  
Fax: 040-2444580

## **NEC Electronics (France) S.A.**

Velizy-Villacoublay, France  
Tel: 01-30-67 58 00  
Fax: 01-30-67 58 99

## **NEC Electronics (France) S.A.**

Madrid Office  
Madrid, Spain  
Tel: 91-504-2787  
Fax: 91-504-2860

## **NEC Electronics (Germany) GmbH**

Scandinavia Office  
Taeby, Sweden  
Tel: 08-63 80 820  
Fax: 08-63 80 388

## **NEC Electronics Hong Kong Ltd.**

Hong Kong  
Tel: 2886-9318  
Fax: 2886-9022/9044

## **NEC Electronics Hong Kong Ltd.**

Seoul Branch  
Seoul, Korea  
Tel: 02-528-0303  
Fax: 02-528-4411

## **NEC Electronics Singapore Pte. Ltd.**

United Square, Singapore  
Tel: 65-253-8311  
Fax: 65-250-3583

## **NEC Electronics Taiwan Ltd.**

Taipei, Taiwan  
Tel: 02-2719-2377  
Fax: 02-2719-5951

## **NEC do Brasil S.A.**

Electron Devices Division  
Guarulhos-SP Brasil  
Tel: 55-11-6462-6810  
Fax: 55-11-6462-6829

J00.7

[MEMO]

## INTRODUCTION

<b>Readers</b>	This manual is intended for user engineers who design and develop application systems of the V850 Family.								
<b>Purpose</b>	This manual is intended for users to understand the functions of the ID850NW in the organization below.								
<b>Organization</b>	<p>This manual consists of the following chapters:</p> <ul style="list-style-type: none"><li>• General</li><li>• Installation</li><li>• Starting and exiting</li><li>• Function of ID850NW</li><li>• Association with project manager</li><li>• Window reference</li></ul>								
<b>How to read this manual</b>	<p>It is assumed that the readers of this manual have general knowledge of electrical engineering, logic circuits, microcontrollers, C language, and assemblers.</p> <p>To understand the functions of the V850 Family → Refer to <b>Hardware User's Manual</b> for each product.</p> <p>To understand the instruction functions of the V850 Family → Refer to <b>V850 Family User's Manual Architecture</b> or <b>V850E/xxx User's Manual Architecture</b>.</p>								
<b>Conventions</b>	<table><tr><td>Data significance:</td><td>Higher digits on the left and lower digits on the right</td></tr><tr><td><b>Note:</b></td><td>Footnote for item marked with <b>Note</b> in the text</td></tr><tr><td><b>Caution:</b></td><td>Information requiring particular attention</td></tr><tr><td>Key descriptions:</td><td>The key descriptions in this manual are explained in terms of the PC-9821 series keyboard. When using a keyboard whose key descriptions differ from the above, use the keys in accordance with the descriptions in <b>APPENDIX D KEY FUNCTION LIST</b>.</td></tr></table>	Data significance:	Higher digits on the left and lower digits on the right	<b>Note:</b>	Footnote for item marked with <b>Note</b> in the text	<b>Caution:</b>	Information requiring particular attention	Key descriptions:	The key descriptions in this manual are explained in terms of the PC-9821 series keyboard. When using a keyboard whose key descriptions differ from the above, use the keys in accordance with the descriptions in <b>APPENDIX D KEY FUNCTION LIST</b> .
Data significance:	Higher digits on the left and lower digits on the right								
<b>Note:</b>	Footnote for item marked with <b>Note</b> in the text								
<b>Caution:</b>	Information requiring particular attention								
Key descriptions:	The key descriptions in this manual are explained in terms of the PC-9821 series keyboard. When using a keyboard whose key descriptions differ from the above, use the keys in accordance with the descriptions in <b>APPENDIX D KEY FUNCTION LIST</b> .								

**Related Documents**

The documents related to this manual are listed below.

The related documents indicated in this publication may include preliminary versions.

However, preliminary versions are not marked as such.

**Documents related to development tools (user's manuals)**

Document Name		Document No.
IE-703002-MC (In-circuit emulator for V851™, V852™, V853™, V854™, V850/SA1™, V850/SB1™, V850/SV1™)		U11595E
IE-V850E-MC (In-circuit emulator for V850E/IA1™) IE-V850E-MC-A (In-circuit emulator for V850E/MA1™)		U14487E
IE-703003-MC-EM1 (Peripheral I/O board for V853)		U11596E
IE-703008-MC-EM1 (Peripheral I/O board for V854)		U12420E
IE-703017-MC-EM1 (Peripheral I/O board for V850/SA1)		U12898E
IE-703037-MC-EM1 (Peripheral I/O board for V850/SB1, V850/SB2™)		U14151E
IE-703040-MC-EM1 (Peripheral I/O board for V850/SV1)		U14337E
IE-703102-MC (In-circuit emulator for V850E/MS1™)		U13875E
IE-703102-MC-EM1, IE-703102-MC-EM1-A (Peripheral I/O board for V850E/MS1)		U13876E
IE-703107-MC-EM1 (Peripheral I/O board for V850E/MA1)		U14481E
IE-703117-MC-EM1 (Peripheral I/O board for V850E/IA1)		U14770E
CA850 (Ver. 2.30 or later) (C compiler package)	Operation	U14568E
	C Language	U14566E
	Project Manager	U14569E
	Assembly Language	U14567E
ID850 (Ver. 2.00 or later) (Integrated debugger)	Operation Windows™ Based	U14217E
ID850 (Ver. 2.20 or later) (Integrated debugger)	Operation Windows Based	U14580E
ID850NW (Ver. 1.10 or later) (Integrated debugger)	Operation Windows Based	This manual
SM850 (Ver. 2.00 or later) (System simulator)	Operation Windows Based	U13759E
SM850 (Ver. 2.20 or later) (System simulator)	Operation Windows Based	U14782E
SM850 (Ver. 2.00 or later) (System simulator)	External Part User Open Interface Specifications	U14873E
RX850 (Real-time OS)	Basics	U13430E
	Installation	U13410E
	Technical	U13431E
RX850 Pro (Real-time OS)	Fundamental	U13773E
	Installation	U13774E
	Technical	U13772E
RD850 (Ver. 3.0) (Task debugger)		U13737E
RD850 Pro (Ver. 3.0) (Task debugger)		U13916E
AZ850 (System performance analyzer)		U14410E



## CONTENTS

<b>CHAPTER 1 GENERAL</b> .....	<b>17</b>
<b>1.1 Overview</b> .....	<b>17</b>
<b>1.2 Features</b> .....	<b>17</b>
<b>1.3 System Configuration</b> .....	<b>18</b>
<b>1.4 Operating Environment</b> .....	<b>19</b>
1.4.1 Hardware environments .....	19
1.4.2 Software environments.....	19
<b>1.5 Note for Debugging at Source Level</b> .....	<b>20</b>
<b>CHAPTER 2 INSTALLATION</b> .....	<b>21</b>
<b>2.1 Installing ID850NW</b> .....	<b>21</b>
<b>2.2 Uninstalling ID850NW</b> .....	<b>27</b>
<b>CHAPTER 3 STARTING AND EXITING</b> .....	<b>31</b>
<b>3.1 Starting</b> .....	<b>31</b>
<b>3.2 Exiting</b> .....	<b>33</b>
<b>CHAPTER 4 FUNCTIONS OF ID850NW</b> .....	<b>35</b>
<b>4.1 Mapping Function</b> .....	<b>35</b>
<b>4.2 Program Execution Function</b> .....	<b>36</b>
4.2.1 Real-time execution function .....	36
4.2.2 Non real-time execution function .....	37
<b>4.3 Event Function</b> .....	<b>38</b>
4.3.1 Using event function.....	38
4.3.2 Event conditions .....	39
<b>4.4 Break Function</b> .....	<b>41</b>
<b>4.5 Trace Function</b> .....	<b>43</b>
4.5.1 Trace operations .....	43
4.5.2 Trace condition setting function.....	43
4.5.3 Trace result display function.....	45
<b>4.6 Load/Save Functions</b> .....	<b>46</b>
4.6.1 Display files .....	46
4.6.2 Information files.....	46
<b>4.7 Register Manipulation Functions</b> .....	<b>47</b>
<b>4.8 Memory Manipulation Functions</b> .....	<b>47</b>
<b>4.9 Real-Time RAM Sampling Function</b> .....	<b>47</b>
<b>CHAPTER 5 ASSOCIATION WITH PROJECT MANAGER</b> .....	<b>49</b>
<b>5.1 Debugger Registration in PM Project</b> .....	<b>49</b>
5.1.1 Debugger selection .....	49
<b>5.2 Starting Up ID850NW from PM</b> .....	<b>50</b>
5.2.1 Reproducing debug environment .....	50

<b>5.3</b>	<b>Correction of Source File and Auto Load .....</b>	<b>51</b>
------------	--	-----------

<b>CHAPTER 6</b>	<b>WINDOW REFERENCE.....</b>	<b>53</b>
------------------	------------------------------	-----------

<b>6.1</b>	<b>Window List.....</b>	<b>53</b>
------------	-------------------------	-----------

<b>6.2</b>	<b>Explanation of Windows .....</b>	<b>55</b>
------------	-------------------------------------	-----------

Main Window .....	56
Configuration Dialog Box .....	81
Extended Option Dialog Box.....	86
Debugger Option Dialog Box .....	90
Interrupt Mask Dialog Box.....	98
Font Dialog Box .....	100
Project File Load Dialog Box.....	103
Project File Save Dialog Box.....	106
View File Load Dialog Box .....	109
View File Save Dialog Box.....	112
Download Dialog Box.....	118
Load Module List Dialog Box .....	122
Upload Dialog Box .....	124
Browse Dialog Box.....	127
Source Move Dialog Box .....	129
Address Move Dialog Box.....	132
Trace Move Dialog Box.....	134
Symbol To Address Dialog Box .....	137
Source Window.....	140
Source Search Dialog Box.....	154
Assemble Window .....	157
Assemble Search Dialog Box.....	166
Memory Window .....	169
Memory Search Dialog Box .....	174
Memory Fill Dialog Box .....	177
Memory Copy Dialog Box .....	179
Memory Compare Dialog Box.....	181
Memory Compare Result Dialog Box.....	183
DMM Dialog Box .....	185
Watch Window .....	188
Quick Watch Dialog Box .....	193
Add Watch Dialog Box .....	198
Register Window.....	201
Register Select Dialog Box .....	204
IOR Window.....	206
IOR Select Dialog Box .....	209
Add I/O Port Dialog Box.....	213
Local Variable Window .....	216
Stack Window .....	218
Trace View Window .....	222
Trace Data Select Dialog Box.....	228
Trace Option Dialog Box.....	231
Event Manager.....	235

Software Break Manager .....	242
Event Dialog Box .....	245
Event Link Dialog Box.....	256
Break Dialog Box .....	266
Trace Dialog Box .....	273
Delay Count Setting Dialog Box .....	281
Reset Debugger Dialog Box .....	283
About Dialog Box .....	285
Exit Debugger Dialog Box.....	286
Error/Warning Dialog Box .....	288
<b>APPENDIX A DEBUGGING WITH ID850NW.....</b>	<b>289</b>
<b>APPENDIX B TERMINOLOGY .....</b>	<b>293</b>
<b>B.1 Terminology .....</b>	<b>293</b>
<b>B.2 Window Type and Configuration .....</b>	<b>302</b>
B.2.1 Windows.....	302
B.2.2 Dialog boxes .....	302
<b>APPENDIX C ERROR MESSAGE LIST.....</b>	<b>303</b>
<b>APPENDIX D KEY FUNCTION LIST.....</b>	<b>327</b>
<b>D.1 Special Function Key Function List .....</b>	<b>327</b>
<b>D.2 Function Key Function List .....</b>	<b>328</b>
<b>D.3 Special Function Key Function List ( <span style="border: 1px solid black; padding: 0 2px;">SHIFT</span> + Key) .....</b>	<b>329</b>
<b>D.4 Function Key Function List ( <span style="border: 1px solid black; padding: 0 2px;">SHIFT</span> + Key) .....</b>	<b>329</b>
<b>D.5 Special Function Key Function List ( <span style="border: 1px solid black; padding: 0 2px;">CTRL</span> + Key) .....</b>	<b>330</b>
<b>D.6 Function Key Function List ( <span style="border: 1px solid black; padding: 0 2px;">CTRL</span> + Key) .....</b>	<b>330</b>
<b>D.7 Control Key Function List ( <span style="border: 1px solid black; padding: 0 2px;">CTRL</span> + Key) .....</b>	<b>330</b>
<b>D.8 Special Function Key Function List ( <span style="border: 1px solid black; padding: 0 2px;">CTRL</span> + <span style="border: 1px solid black; padding: 0 2px;">SHIFT</span> Key) .....</b>	<b>331</b>
<b>APPENDIX E INDEX.....</b>	<b>333</b>

## LIST OF FIGURES (1/2)

Figure No.	Title	Page
1-1	ID850NW System Configuration Example .....	18
3-1	ID850NW Icon .....	31
3-2	Configuration Dialog Box on Starting.....	31
3-3	Main Window.....	32
3-4	comctl32.dll 4.72 Installer .....	32
3-5	Exit Debugger Dialog Box .....	33
5-1	Select Debugger Type Dialog Box (PM) .....	49
6-1	Main Window.....	56
6-2	Toolbar .....	57
6-3	Example of Tool Hint .....	59
6-4	Status Bar .....	60
6-5	<u>F</u> ile Menu Bar .....	63
6-6	<u>E</u> dit Menu Bar.....	64
6-7	<u>V</u> iew Menu Bar .....	66
6-8	<u>O</u> ption Menu Bar.....	71
6-9	<u>R</u> un Menu Bar .....	73
6-10	<u>E</u> vent Menu Bar .....	76
6-11	<u>B</u> rowse Menu Bar .....	77
6-12	<u>J</u> ump Menu Bar .....	78
6-13	<u>W</u> indow Menu Bar.....	79
6-14	<u>H</u> elp Menu Bar.....	80
6-15	Configuration Dialog Box .....	81
6-16	Extended Option Dialog Box .....	86
6-17	Debugger Option Dialog Box.....	90
6-18	Interrupt Mask Dialog Box.....	98
6-19	Font Dialog Box .....	100
6-20	Project File Load Dialog Box .....	103
6-21	Project File Save Dialog Box .....	107
6-22	View File Load Dialog Box .....	109
6-23	View File Save Dialog Box .....	112
6-24	Download Dialog Box .....	119
6-25	Load Module List Dialog Box.....	122
6-26	Upload Dialog Box .....	124
6-27	Browse Dialog Box.....	127
6-28	Source Move Dialog Box .....	129
6-29	Address Move Dialog Box.....	132
6-30	Trace Move Dialog Box .....	134
6-31	Symbol To Address Dialog Box .....	137
6-32	Source Window .....	140

## LIST OF FIGURES (2/2)

Figure No.	Title	Page
6-33	Source Search Dialog Box .....	154
6-34	Assemble Window .....	157
6-35	Assemble Search Dialog Box .....	166
6-36	Memory Window .....	169
6-37	Memory Search Dialog Box .....	174
6-38	Memory Fill Dialog Box .....	177
6-39	Memory Copy Dialog Box .....	179
6-40	Memory Compare Dialog Box .....	181
6-41	Memory Compare Result Dialog Box .....	183
6-42	DMM Dialog Box .....	185
6-43	Watch Window .....	189
6-44	Quick Watch Dialog Box .....	193
6-45	Add Watch Dialog Box .....	198
6-46	Register Window .....	201
6-47	Register Select Dialog Box .....	204
6-48	IOR Window .....	206
6-49	IOR Select Dialog Box .....	209
6-50	Add I/O Port Dialog Box .....	213
6-51	Local Variable Window .....	216
6-52	Stack Window .....	218
6-53	Trace View Window .....	222
6-54	Trace Data Select Dialog Box .....	228
6-55	Trace Option Dialog Box .....	231
6-56	Event Manager .....	236
6-57	Software Break Manager .....	242
6-58	Event Dialog Box .....	246
6-59	Event Link Dialog Box .....	257
6-60	Break Dialog Box .....	267
6-61	Trace Dialog Box .....	274
6-62	Delay Count Setting Dialog Box .....	281
6-63	Reset Debugger Dialog Box .....	283
6-64	About Dialog Box .....	285
6-65	Exit Debugger Dialog Box .....	286
6-66	Error/Warning Dialog Box .....	288
C-1	Display Format of Error Message .....	303

## LIST OF TABLES (2/1)

Table No.	Title	Page
4-1	Relationship Between Event Conditions and Setting Dialog Box .....	38
4-2	Each Condition That Can Be Set in Event Setting Dialog Box.....	39
4-3	Event Condition Type .....	40
4-4	Types of Trace Mode .....	43
4-5	Types of Trace Data to Be Fetched.....	43
4-6	Types of Trace.....	44
4-7	Types of Tracer Control Mode .....	44
4-8	Trace Data Display Contents.....	45
4-9	Display Files Handled by ID850NW .....	46
4-10	Information Files Handled by ID850NW.....	46
6-1	Windows of ID850NW Debugger .....	53
6-2	CPU Status Display Contents.....	61
6-3	IE Status Display Contents .....	61
6-4	Display Contents of Break Cause .....	61
6-5	STEP Modes .....	61
6-6	Trace Memory Size .....	87
6-7	Saved Contents of Project File .....	105
6-8	View File Extension.....	110
6-9	File Types and Current Window.....	115
6-10	Types of Files That Can Be Downloaded.....	120
6-11	Types of Files That Can Be Uploaded.....	125
6-12	Types of Files That Can Be Displayed in Browse Dialog Box.....	128
6-13	Format of Frame No. Specification.....	135
6-14	Format of Symbol Specification .....	138
6-15	Breakpoint Setting/Deleting Function .....	143
6-16	Breakpoint Display Color .....	143
6-17	Meaning of Event Display Mark .....	144
6-18	Program Codes .....	144
6-19	Details of Drag and Drop Function (Line/Address).....	146
6-20	Current PC Line and Breakpoint Setting Line .....	148
6-21	Details of Drag and Drop Function (Character String) .....	152
6-22	Details of Jump Function .....	221
6-23	Contents of Fetch Operation Access Display .....	225
6-24	Contents of Program Fetch Display.....	225
6-25	Contents of Data Operation Access Display.....	225
6-26	Window Link Function .....	227
6-27	Trace Triggers of Execution Trace Data .....	234
6-28	Event Icon Marks .....	237
6-29	Character Color of Event Icon .....	238
6-30	Key Information in Detailed Event Display Mode .....	240
6-31	Number of Valid Event Conditions .....	247

## LIST OF TABLES (2/2)

Table No.	Title	Page
6-32	Meanings of Event Icons (Event Condition) .....	248
6-33	List of Status Conditions in Event Dialog Box .....	249
6-34	List of Access Size Conditions in Event Dialog Box .....	250
6-35	List of Data Conditions in Event Dialog Box .....	252
6-36	Number of Valid Event Link Conditions .....	258
6-37	Meanings of Event Icons (Event Link Condition) .....	259
6-38	Number of Events That Can Be Set in Link Condition Setting Area .....	260
6-39	Number of Events That Can Be Set in Disable Condition Setting Area .....	261
6-40	Number of Valid Break Event Conditions.....	268
6-41	Meanings of Event Icons (Break Event Condition).....	269
6-42	Number of Events That Can Be Set in Break Condition Setting Area .....	269
6-43	Total Number of Events That Can Be Set to Each Area in Trace Dialog Box.....	275
6-44	Meanings of Event Icons (Trace Event Condition).....	276
B-1	Delimiter Symbols.....	294
B-2	Register Set of V850 Family .....	296
C-1	Error Message Type .....	303
C-2	Error Message List.....	304
D-1	Special Function Key Function List .....	327
D-2	Function Key Function List.....	328
D-3	Special Function Key Function List ( <span style="border: 1px solid black; padding: 2px;">SHIFT</span> + Key) .....	329
D-4	Function Key Function List ( <span style="border: 1px solid black; padding: 2px;">SHIFT</span> + Key).....	329
D-5	Special Function Key Function List ( <span style="border: 1px solid black; padding: 2px;">CTRL</span> + Key) .....	330
D-6	Function Key Function List ( <span style="border: 1px solid black; padding: 2px;">CTRL</span> + Key).....	330
D-7	Control Key Function List ( <span style="border: 1px solid black; padding: 2px;">CTRL</span> + Key).....	330
D-8	Special Function Key Function List ( <span style="border: 1px solid black; padding: 2px;">CTRL</span> + <span style="border: 1px solid black; padding: 2px;">SHIFT</span> Key).....	331

[MEMO]



## CHAPTER 1 GENERAL

### 1.1 Overview

The V850 Family integrated debugger ID850NW (hereafter referred to as "ID850NW") is a software tool to debug the user programs developed for NEC's V850 Family 32-bit microcontroller for embedded control applications efficiently.

The ID850NW features an easy-to-understand GUI (Graphical User Interface) on a host machine using Windows as the operating system.

Moreover, the commands that are frequently used can be input using the mouse, providing an environment with excellent operability.

### 1.2 Features

The features of the ID850NW are as follows.

#### (1) GUI function

The ID850NW runs in a Windows environment and its debugging operations can be controlled with a mouse. Buttons and menus are displayed in each window, and you can select other related information from the displayed information.

#### (2) Source debug

C source and assembly language programs can be debugged at source level.

#### (3) Use of in-circuit emulator

Breakpoints can be set and user programs can be traced<sup>Note</sup> by using the detailed event setting functions of an in-circuit emulator.

**Note** The trace function is valid only for CPU cores with the TRCU incorporated.

#### (4) Monitor function (automatic display updating function while execution is stopped)

When execution of the user program is stopped, the values displayed in the window are automatically updated.

#### (5) Saving/restoring debugging environment

The debug environment can be saved in a file. The saved environment can be restored and can start debugging again.

#### (6) Function expansion by TIP (Tool Interface Protocol)

By linking up with a task debugger (RD), system performance analyzer (AZ), etc., it is possible to vastly improve the debugging efficiency of applications that use a real-time OS (RX).

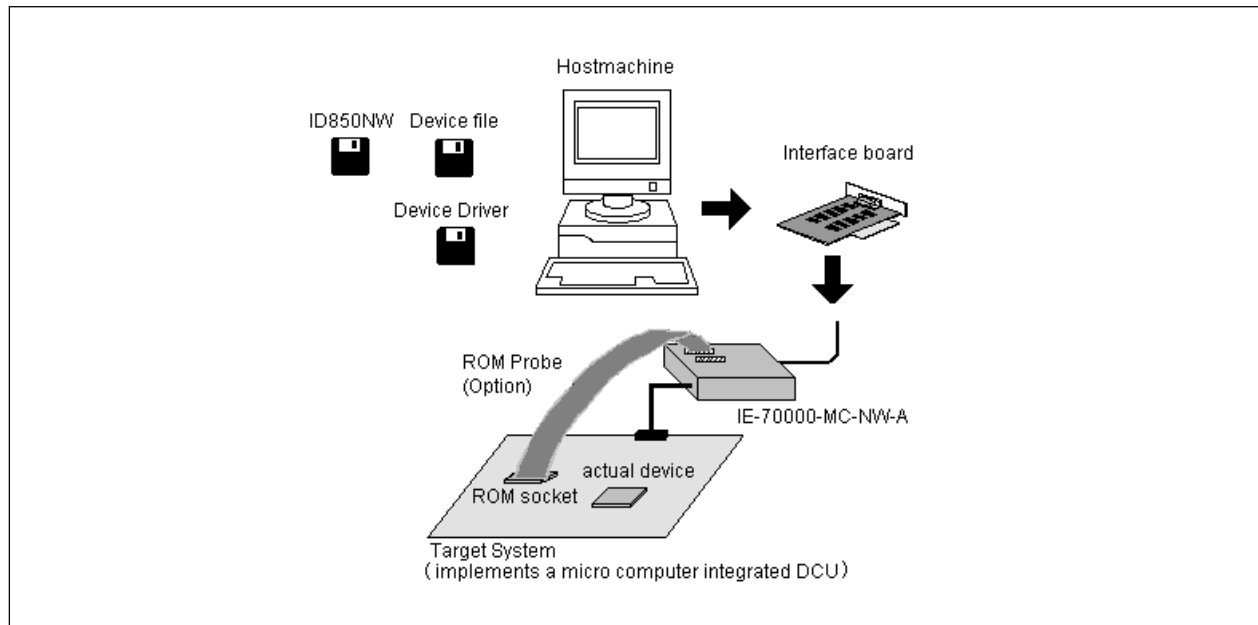
### 1.3 System Configuration

The ID850NW offers a comfortable debugging environment for user programs and target boards developed for the V850 Family by connecting the host machine and the in-circuit emulator with a dedicated parallel interface board.

The in-circuit emulator (IE-70000-MC-NW-A) used by the ID850NW provides a debugging function by connecting a target board on which a V850 Family microcontroller with the DCU (Debug Control Unit) incorporated is mounted.

Figure 1-1 shows an example of the system configuration of the ID850NW.

**Figure 1-1. ID850NW System Configuration Example**



## 1.4 Operating Environment

To use the ID850NW, the following hardware and software environments are necessary.

### 1.4.1 Hardware environments

#### (1) Host machine

- PC-9821 series
  - PC98-NX series
  - IBM PC/AT™ or compatibles
- CPU: Pentium™ 100 MHz or more (120 MHz or more is recommended)  
Memory: RAM: 32 MB or more (64 MB or more is recommended)

#### (2) In-circuit emulator

- IE-70000-MC-NW-A

#### (3) Interface board

- IE-70000-98-IF-C (PC-9821 series)
- IE-70000-PC-IF-C (IBM PC/AT or compatibles)
- IE-70000-PCI-IF

#### (4) ROM probe (option)

- EP-16000C
- EP-16384C

### 1.4.2 Software environments

#### (1) OS

Windows 95, Windows 98, Windows NT™4.0 (Service Pack 3)

#### (2) Device file

Device file of target device to be used

#### (3) Device driver

Device driver for interface board (attached to this product)

### **1.5 Note for Debugging at Source Level**

When debugging at source level, add an option (-g option) that creates debugging information after the source files have been compiled. If this is not done, debugging cannot be performed in the source level.

## CHAPTER 2 INSTALLATION

This chapter explains how to install/uninstall the ID850NW.

### 2.1 Installing ID850NW

In order to use the ID850NW, it is necessary to install the ID850NW, a device file, and a device driver. Install the device file according to the installer of the device file (refer to **CA830, CA850 C Compiler Package User's Manual Project Manager**).

Install the device driver according to the Readme-j.txt in the floppy disk "NEC IE-PC Driver Installation Disk".

**Caution** If reinstalling the ID850NW, be sure to uninstall it first. If the ID850NW is not uninstalled and reinstalled in a directory that is different to the one the ID850NW was originally installed in, uninstallation of the originally installed ID850NW may become impossible.

The following floppy disks (FDs) are necessary for installing the ID850NW.

- System disks of ID850NW (DISK 1 to 5)

Copy the system disks to other floppy disks for backup. Use the backed up disks for installation to prevent an accident.

An example of installing the ID850NW is shown below.

The setting for this example is as follows:

- Install directory: C:\nectools32
- Drive of floppy disk: Drive A
- Directory to which Windows is installed: C:\WINDOWS

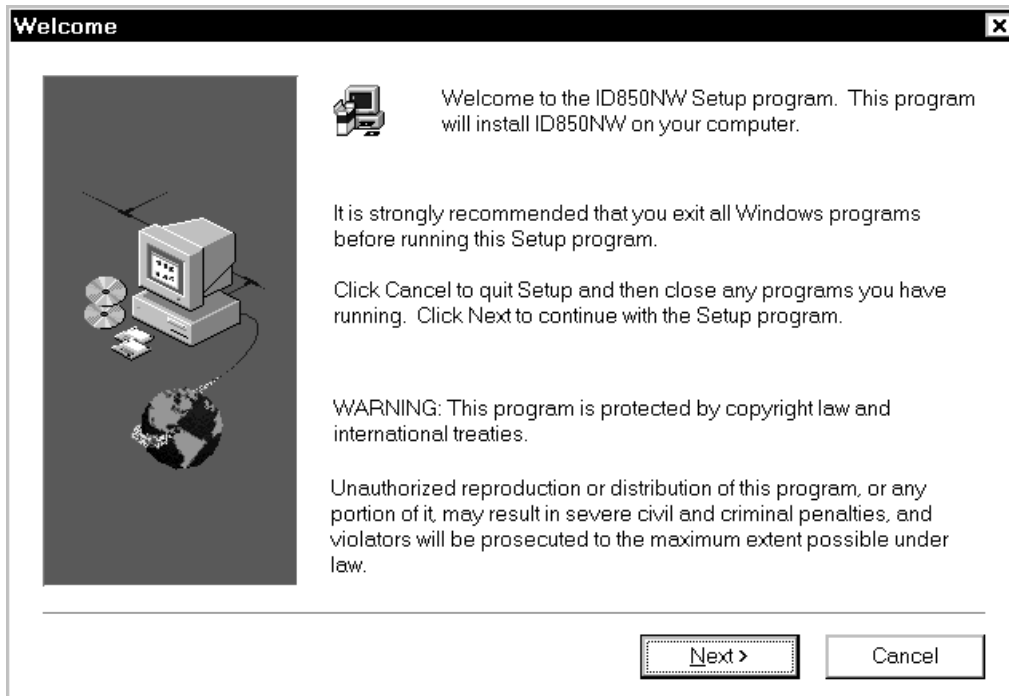
<1> Start up Windows.

<2> Insert the system disk (DISK 1) to a floppy disk drive (drive A in this example).

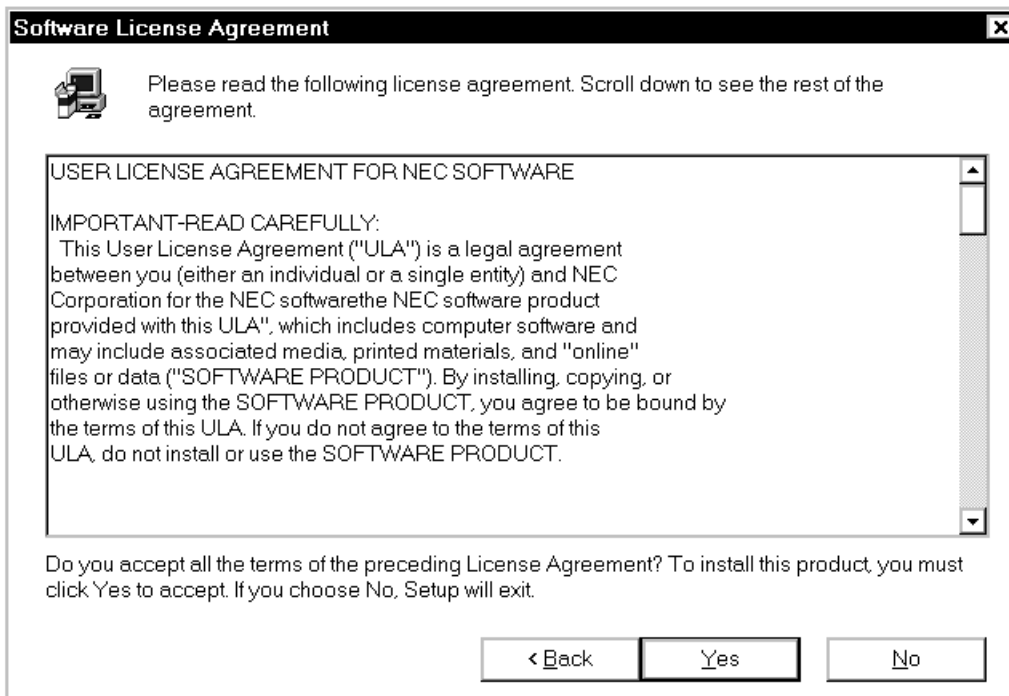
<3> Start up the explorer, select drive A, and double-click "Setup.exe".

<4> The setup program is activated following setup initialization.

Click .



<5> When installing the ID850NW, click  after reading and accepting the terms of the software license agreement. Clicking  will abort installation.



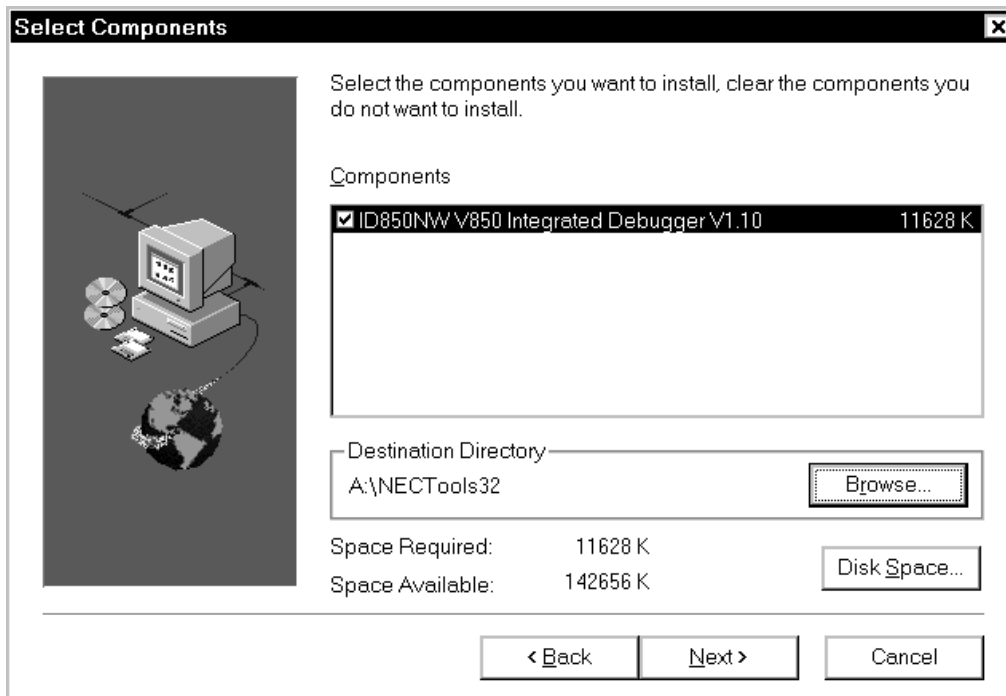
<6> Select the items to be installed and the destination drive or directory.

Clear the check mark from the components that are not to be installed. The default destination drive or directory is C:\nectools32.

To change the drive or directory, click **Browse...** and make appropriate modifications.

After setting each item, click **Next >**.

To abort installation, click **Cancel**.



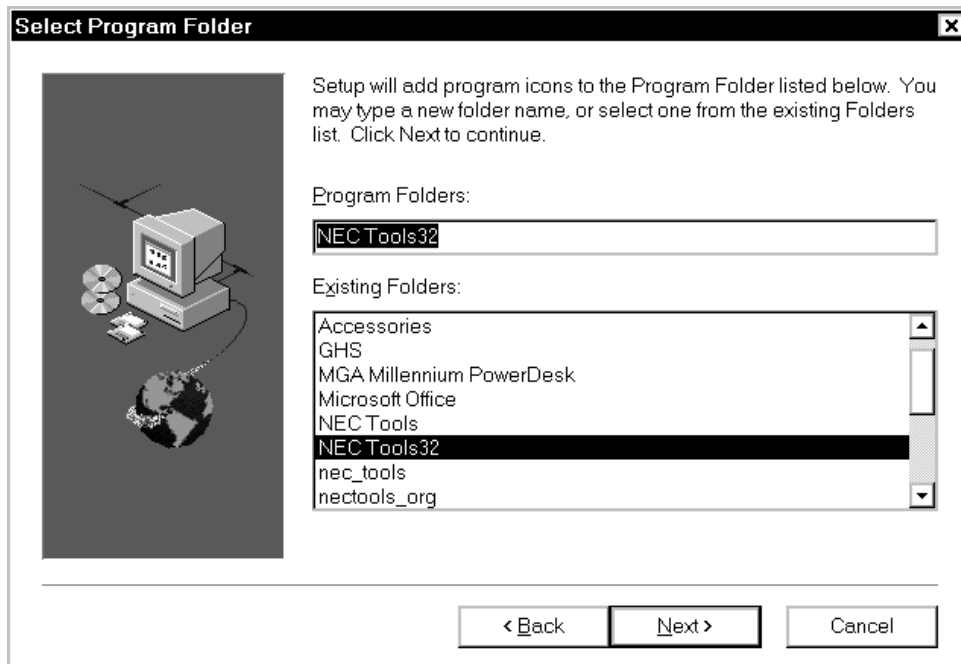
**Caution** If a previously installed ID850NW has not been uninstalled, a dialog box querying whether to uninstall it will be displayed.

<7> Specify the name of the file in which the ID850NW icon is to be registered.

After specifying the group name, click .

To abort installation, click .

The default group name is "NEC Tools32".

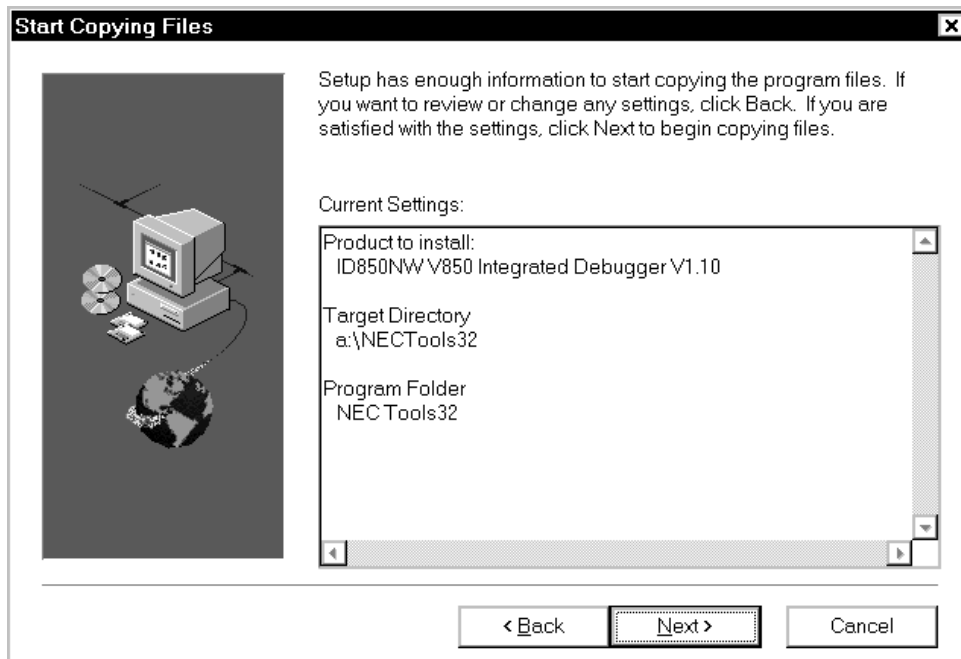


<8> Final confirmation of installation start

Confirm the settings made in <6> and <7>.

If there are no changes to these settings, click .

If changes are necessary, click . If there are any problems, click  and abort installation.

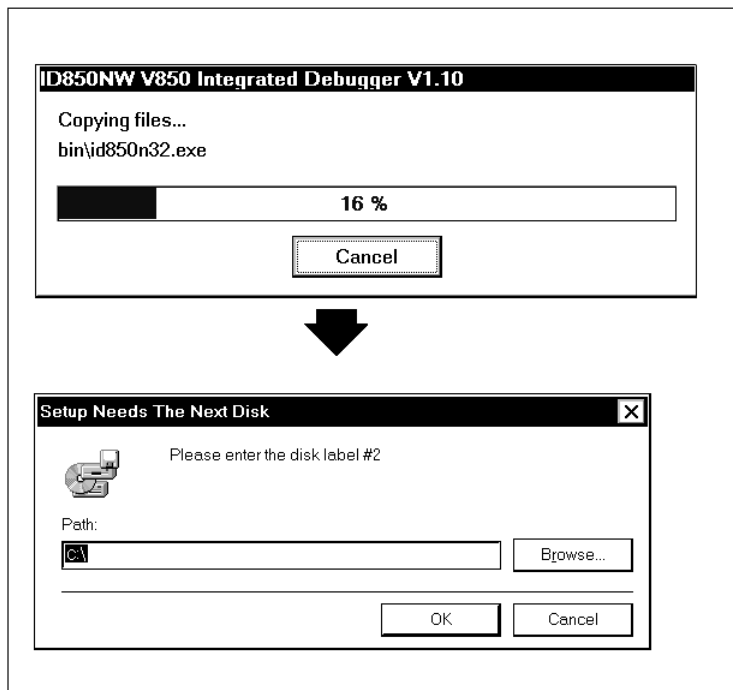




<9> Copying files

Copy the files into the directory set in <6>. The change-of-system-disk message (Setup Needs the Next Disk) dialog box will open, so after changing the disk, click .

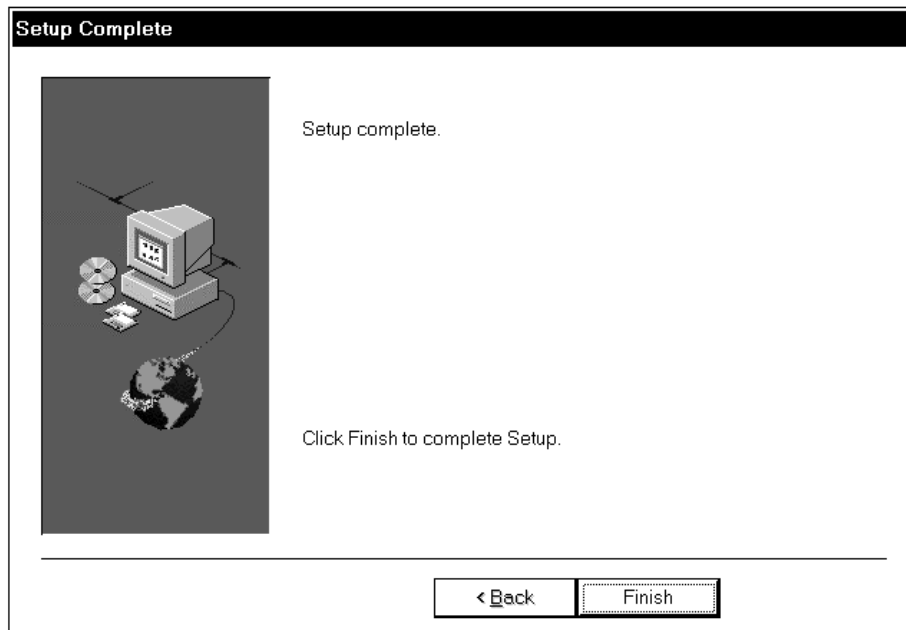
Displays the installation status



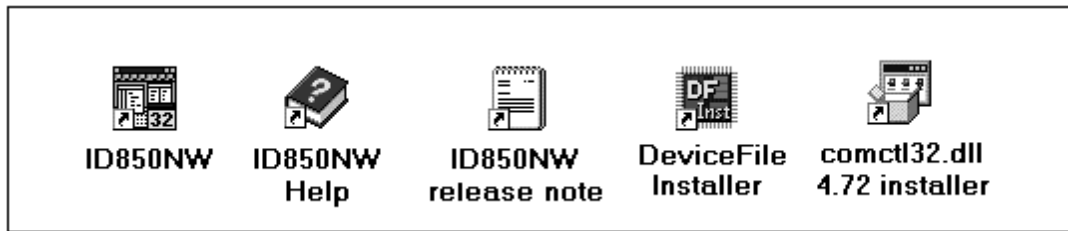
Change-of-system-disk message

<10> Completing installation

After installation is completed, the Setup Complete dialog box will open, and click . If the data file for 16-bit version NEC development tools NECDEV.INI exists in the Windows directory, a message querying whether to move the device file data to the registry will be displayed. To move this data, click . If you do not desire this data to be moved, click .



<11> When the system disks have all been copied, the ID850NW icon will be registered in the “NEC Tools” group.



Installation of ID850NW is now complete.

## 2.2 Uninstalling ID850NW

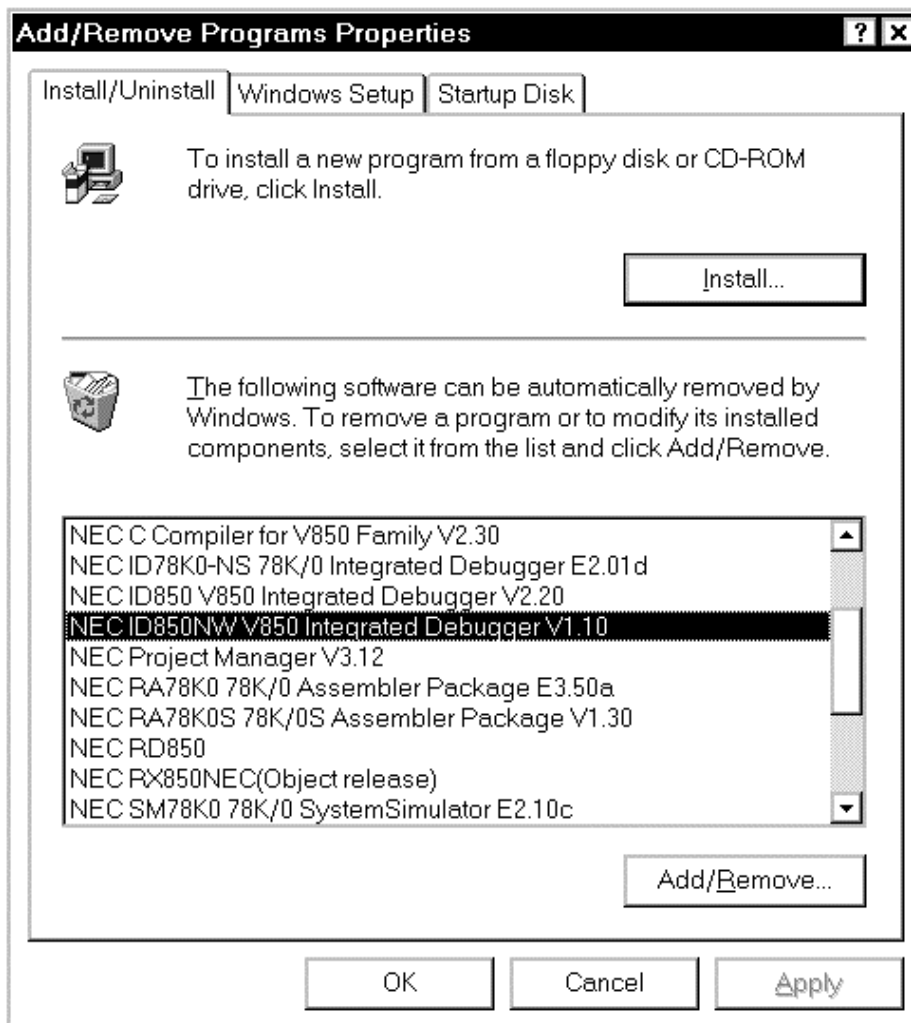
This section explains how to uninstall the ID850NW, using "C:\WINDOWS" as the directory in which Windows is installed.

<1> Start up Windows

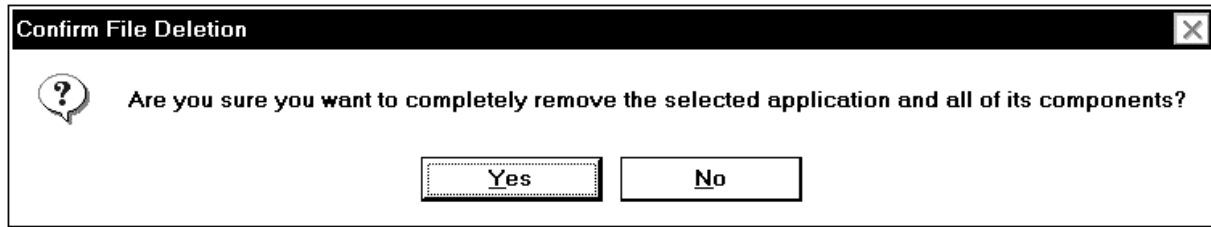
<2> Activate "Add/Remove Programs" from the control panel.



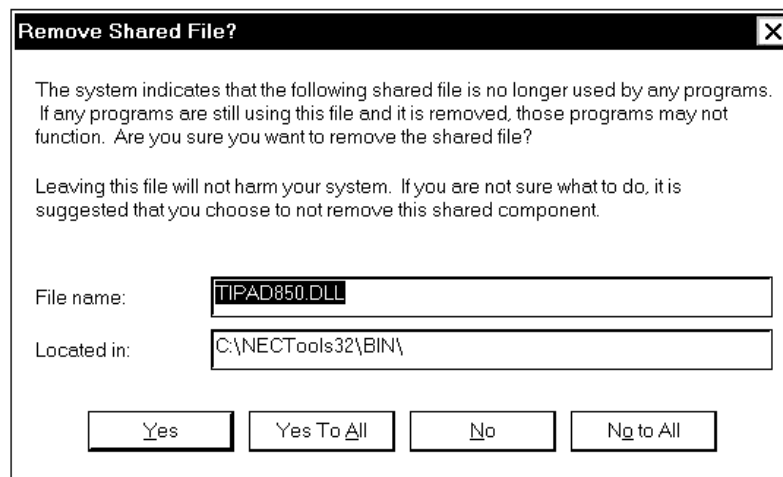
<3> Select "NEC ID850NW V850 Integrated Debugger VX.XX" from the list displayed in "Install/Uninstall" and click .



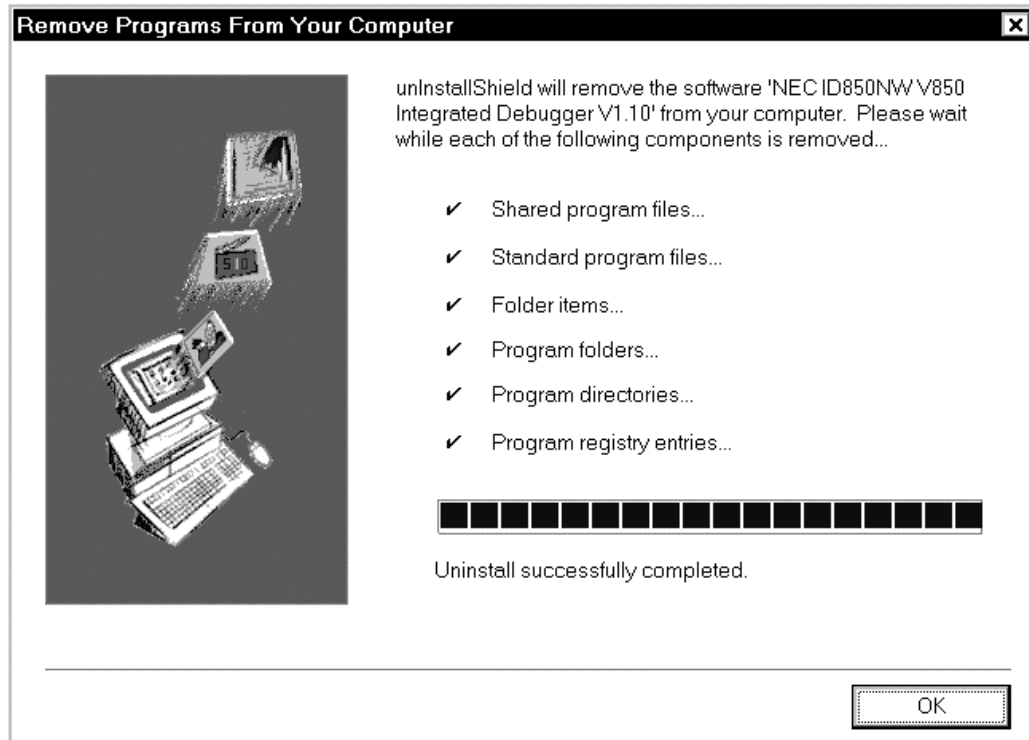
<4> The Confirm File Deletion dialog box will be displayed next, and click .



<5> The following dialog box below will appear during deletion. Click  as to all files confirmed (even if  is selected and files are not deleted here, the latest files with the same name overwrite the old files when the ID850NW is reinstalled).



<6> The program will be deleted. When the message "Uninstall completed" appears, click .



Uninstallation of ID850NW is now complete.

[MEMO]

## CHAPTER 3 STARTING AND EXITING

This chapter explains how to start and exit the ID850NW.

### 3.1 Starting

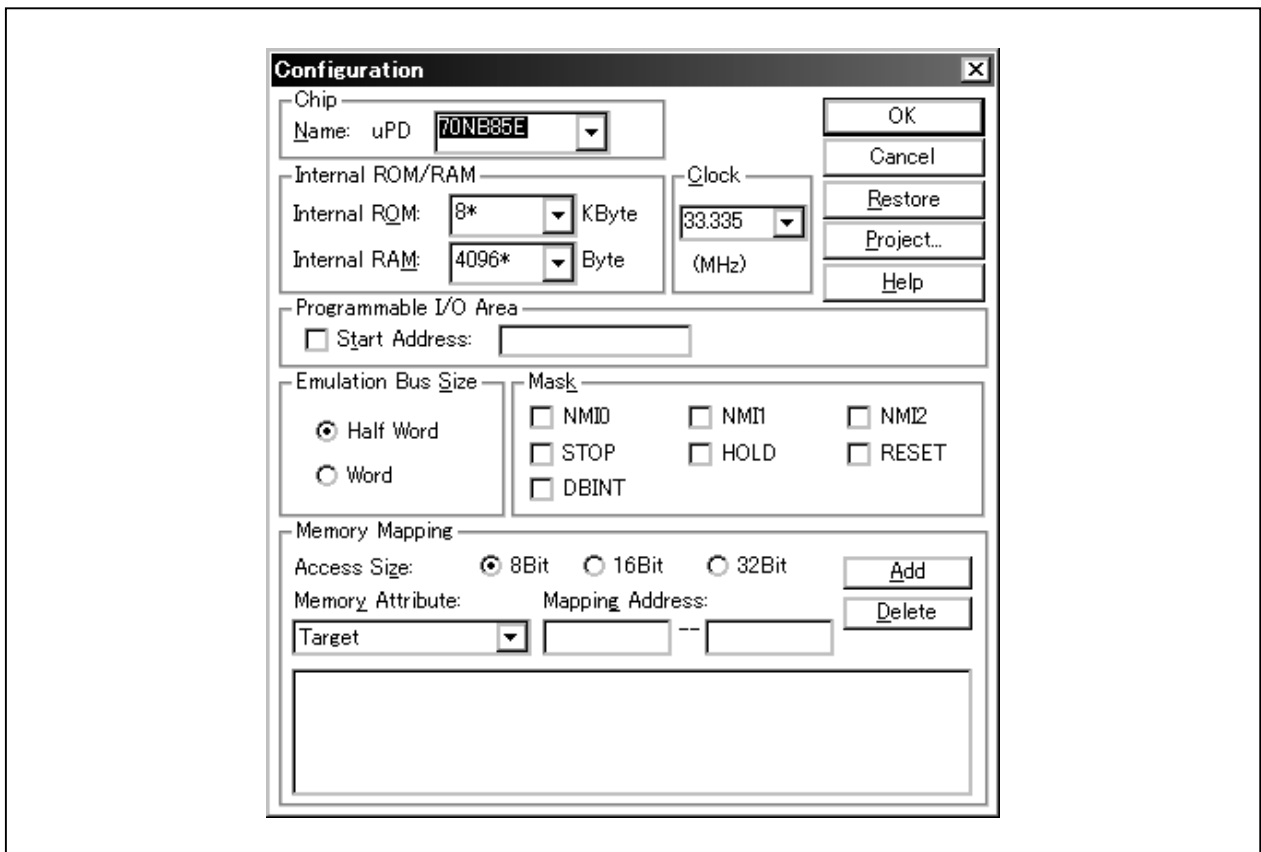
The start method is as follows:

- <1> Turn on power to the in-circuit emulator and target system.
- <2> Start Windows on the PC.
- <3> Double-click the shortcut icon of the ID850NW on the desktop to start up the ID850NW (See **Figure 3-1**).  
The Configuration dialog box shown will be displayed (See **Figure 3-2**).

**Figure 3-1. ID850NW Icon**

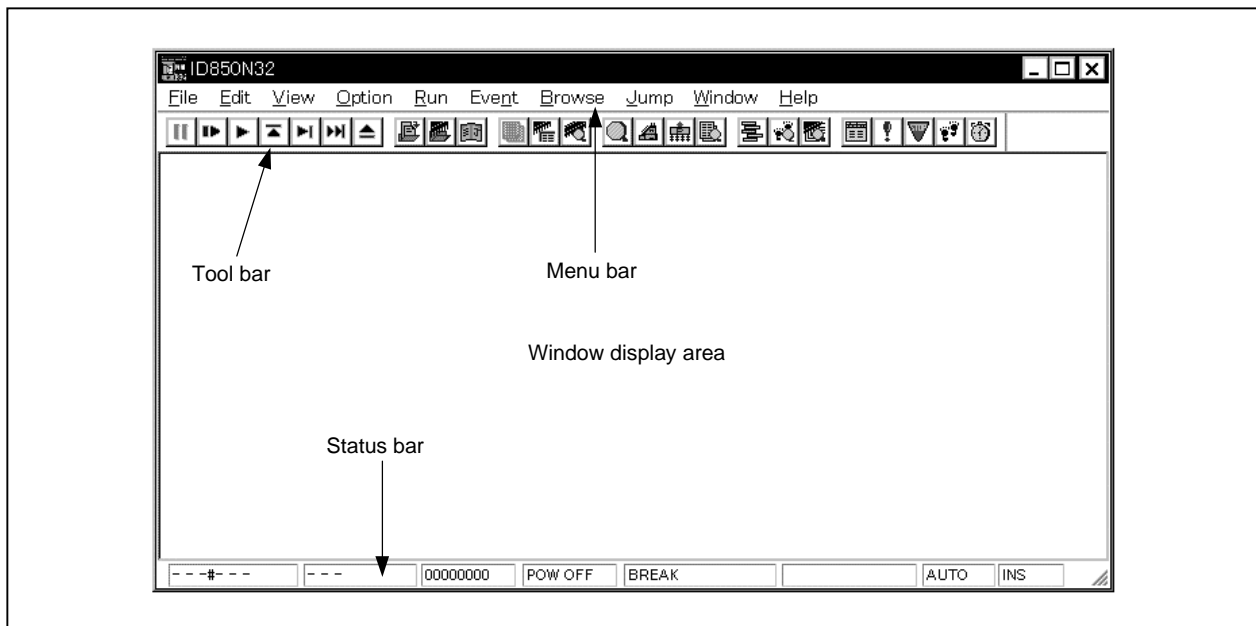


**Figure 3-2. Configuration Dialog Box on Starting**



- <4> Operating environment parameters for the ID850NW are set in this dialog box (For details, refer to **CHAPTER 6 WINDOW REFERENCE**). After setting the parameters, click the  button on the dialog box.
- <5> The main window will then be opened (See **Figure 3-3**) and the operation of the ID850NW is possible. Debugging operations are all carried out from this window.

Figure 3-3. Main Window

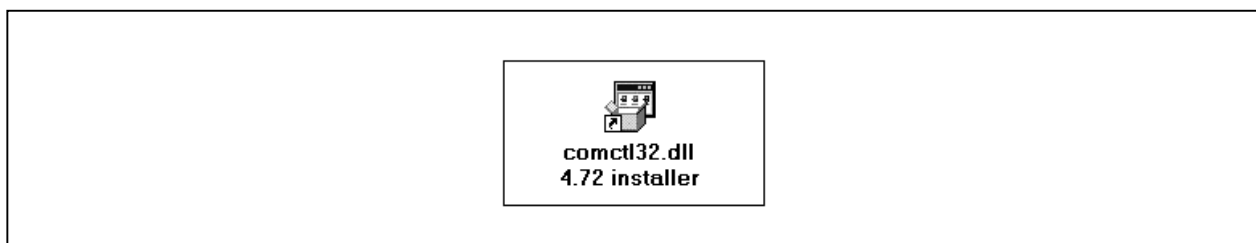


**Caution** Problems such as dialog boxes not displaying data correctly are sometimes experienced when the ID850NW is started up, or if an application error occurs while in use. One cause of such problems may be that an old version of comctl32.dll, which comes with Windows 95, is being used.

If this is the case, it will be necessary to update comctl32.dll using the comctl32.dll 4.72 Installer (40comupd.exe) included in this product.

Note that if Windows 98, Microsoft Internet Explorer 3.X or 4.X has been installed, this update is unnecessary.

Figure 3-4. comctl32.dll 4.72 Installer



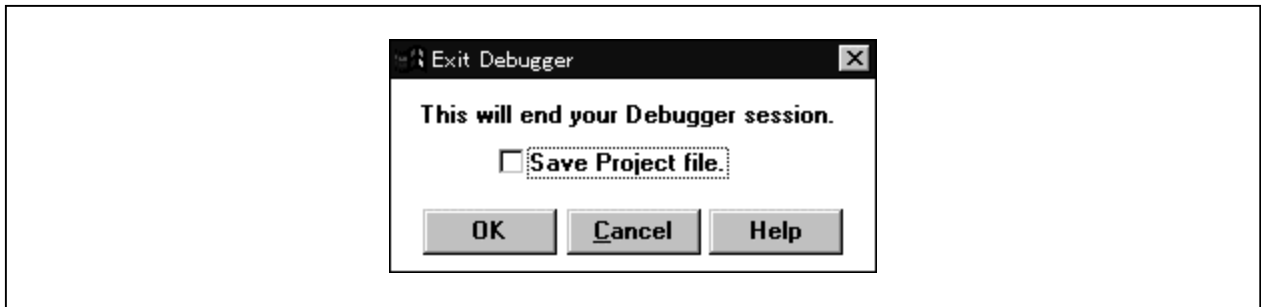


## 3.2 Exiting

The exit method is as follows:

- <1> In the main window, select [File] → [Exit].  
→ The following Exit Debugger dialog box will be displayed.

Figure 3-5. Exit Debugger Dialog Box



- <2> To save the current debugging environment in the project file, check the check box "☐ Save Project file".
- <3> Click the  button.  
If the check box is checked in <2>, the Save As dialog box is opened, the current debugging environment is saved in the project file, all the windows are closed, and then the ID850NW is exited.  
If the check box is not checked, all the windows are closed and then the ID850NW is exited.

[MEMO]

## CHAPTER 4 FUNCTIONS OF ID850NW

This chapter explains the basic functions and processing conventions of the ID850NW.

### 4.1 Mapping Function

The following three types of mapping functions are available.

These mapping functions are set in the Configuration dialog box (Refer to **CHAPTER 6 WINDOW REFERENCE**).

#### (1) Internal ROM

Specify the size of internal ROM inside the target device (core) for the internal ROM size.

Although the size of this area can be changed, the actual memory configuration depends on the target.

#### (2) Internal RAM

Specify the size of internal RAM inside the target device (core) for the internal RAM size.

Although the size of this area can be changed, the actual memory configuration depends on the target.

#### (3) User area mapping (Target)

The memory area specified for user area mapping becomes the area to access either memory in the target system or memory incorporated in the chip. The target device accesses either memory in the target system or memory incorporated in the chip.

#### (4) ROM Emulation RAM

The memory area specified as the ROM emulation RAM is a ROM emulation area in which a ROM probe<sup>Note</sup> is used. The target device accesses the memory inside the in-circuit emulator.

In this area, high-speed downloading and setting pseudo software breaks for the ROM area are possible.

**Note** The ROM probe is optional.

#### (5) I/O protect area (I/O Protect)

The I/O protect area can be set in the area specified as Target. This area is displayed in the same manner as an unmapped area on the Memory window (the symbol ?? is displayed). If this area is mapped with this attribute, it can no longer be read or written easily, protecting it from access from the Memory window.

To read or write a value of the area mapped with this attribute, register in the IOR window or Watch window (Refer to **CHAPTER 6 WINDOW REFERENCE**).

## 4.2 Program Execution Function

The program execution functions start the execution of the user program and the operation of the tracer<sup>Note</sup> by the CPU.

These functions are classified as follows by the execution mode of the user program.

**Note** The trace function is valid only for CPU cores with the TRCU incorporated.

### 4.2.1 Real-time execution function

Real-time execution can be performed by the following commands.

- Go command (  button)
- Start command (Start function) (no button)
- Go & Go command (no button)
- Come command (Come function) (no button)
- Restart command (  button)
- Return Out command (  button)

#### (1) Go command ( button)

This command executes the user program starting from the address indicated by the current PC register. Execution of the user program is stopped when a specified break event condition is satisfied.

Each analyzer gets ready to operate when the user program is executed, and is executed or stopped depending on whether each event condition (break event condition, trace event condition<sup>Note</sup>, etc.) is satisfied or not.

**Note** The trace function is valid only for CPU cores with the TRCU incorporated.

#### (2) Start command (Start function)

This command executes the user program starting from a specified address. Execution of the user program is stopped when a specified break event condition is satisfied.

#### (3) Go & Go command

This command executes the user program starting from the address indicated by the current PC register. Execution of the user program is stopped once if a specified break event condition is satisfied. The contents of each window are updated, and the user program is then executed again starting from the address where it stopped. These operations are repeated until the user issues a Stop command.

#### (4) Come command (Come function)

This command executes the user program starting from the address indicated by the current PC register to the address selected in the line/address display area on the Source window or Assemble window (Refer to **CHAPTER 6 WINDOW REFERENCE**), and then a break occurs.

While the user program is being executed by this command, the break event currently set does not occur.

#### (5) Restart command ( button)

This command resets the CPU and then executes the user program starting from address 0.



This operation is the same as when the Go command is executed following reset of the CPU prior to user program execution.

#### (6) Return Out command ( button)

This command executes until execution returns to the calling function. This command is for functions described in C language.

#### 4.2.2 Non real-time execution function

Non real-time execution can be performed by using the following commands.

- Step In command (  button)
- Next Over command (  button)
- Slow-motion command (no button)

**(1) Step In command (  button)**

This command executes instructions in the program one by one. If functions or subroutines are called, instructions inside them are executed one by one.

**(2) Next Over command (  button)**

This command executes instructions in the program one by one. If functions or subroutines are called, instructions inside them are removed from step execution.

**(3) Slow-motion command**

This command continue to step executing, the contents of each window are updated each time this command is executed. These operations are repeated until the user issues a STOP command.

### 4.3 Event Function

An event is a “specific status of the target system during user program execution” for example “address 0x1000 fetched” and “data written to address 0x2000” showing the specific status of the target system during debugging.

The ID850NW uses these events as triggers of action of break and trace<sup>Note</sup> functions. An event function sets, deletes, or references the “specific status of the target system” that triggers an action (debug action) as an “event condition”.

**Note** The trace function is valid only for CPU cores with the TRCU incorporated.

#### 4.3.1 Using event function

To use an action function that is triggered by an event in accordance with the user’s debugging aims, the event conditions shown in Table 4-1 must be set.

**Table 4-1. Relationship Between Event Conditions and Setting Dialog Box**

Event Condition	Setting Dialog Box	Function
Break event condition	Break dialog box	Condition to execute user program or stop operation of tracer <sup>Note</sup>
Trace event condition <sup>Note</sup>	Trace dialog box	Start/end condition when execution process of user program is saved in trace memory

**Note** The trace function is valid only for CPU cores with the TRCU incorporated.

These event conditions are set by using an “event condition” or “event link condition” alone or in combination.

To use the event function of the ID850NW, therefore, the necessary “event condition” and “event link condition” are first created, and then the event conditions shown in Table 4-1 are set by using these conditions.

### 4.3.2 Event conditions

How to create each event condition is explained next.

#### (1) Creating event conditions

An “event condition” is set in the Event dialog box (Refer to **CHAPTER 6 WINDOW REFERENCE**). An address condition, status condition, data condition, and access size condition are set in this dialog box. A name is given to the combination of these conditions for registration. The maximum number of event conditions that can be registered in the Event dialog box is 256. The maximum number of event conditions that can be simultaneously used for each event condition is 8<sup>Note</sup>.

Table 4-2 lists the contents of each condition that can be set in the Event dialog box.

**Note** The number of execution events differs depending on whether the TEU is incorporated or not.

**Table 4-2. Each Condition That Can Be Set in Event Setting Dialog Box**

Condition	Contents
Address condition	Uses a specified address or address range as an event. A mask value can be set for the specified address.
Status condition	Uses a status for an address condition as an event. The status is selected from the following <sup>Note 1</sup> . <ul style="list-style-type: none"> <li>• Execution: Execution of instruction<sup>Note 2</sup></li> <li>• Before Execution: Before execution of instruction<sup>Notes 2, 3</sup></li> <li>• R/W: Reading/writing of memory</li> <li>• Read: Reading of memory</li> <li>• Write: Writing of memory</li> <li>• Access: All access statuses</li> </ul>
Data condition	Uses the data detected by a status condition as an event. A mask value can be set for the data.
Access size condition	Select the memory access size condition of an access event from the following: <ul style="list-style-type: none"> <li>• Byte: 1-byte access condition</li> <li>• Half Word: 2-byte access condition</li> <li>• Word: 4-byte access condition</li> <li>• No Condition: Access size condition is ignored.</li> <li>• Bit: Bit access condition</li> </ul>

- Notes**
1. The status conditions that can be set differ depending on the connected in-circuit emulator.
  2. If Execution or Before Execution is selected as the status condition, the data, data mask, and access size conditions are invalid.
  3. Up to 2 events can be enabled in Before Execution status, which can be used only for the break event condition. In addition, address range cannot be specified without TEU.

**(2) Event condition type**

Event conditions are classified into the three event types shown in the table below. These event types are automatically determined by selecting the desired status condition when creating an event condition. Table 4-3 lists event condition types.

**Table 4-3. Event Condition Type**

Event Type	Status	Function
Execution event	Execution	Event condition is satisfied when user program begins execution of instruction at specified address.
	Before Execution	Event condition is satisfied just before user program begins execution of instruction at specified address.
Access event	Read Write R/W	Event condition is satisfied when user program accesses specified memory.
	Access	All access statuses of other than above.

**(3) Creating event link condition**

An “event link condition” is a condition under which a sequential rule is applied to the respective events to treat the events as a single event. An event link condition is created in the Event Link dialog box (Refer to **CHAPTER 6 WINDOW REFERENCE**). By arranging the event conditions registered on the Event dialog box in advance in an arbitrary order on the Event Link dialog box, a name is given to the arrangement of the event conditions as an event link condition for registration. The registered event link condition can be used to set various event conditions in the same manner as event conditions.

The maximum number of event link conditions that can be set in the Event Link dialog box is 256. However, the number of event link conditions that can be simultaneously used for each event condition is 3<sup>Note</sup>.

**Note** This is valid only for CPU cores with TEU incorporated.

**(4) Displaying setting status of event condition**

All the registered event conditions and event link conditions are managed or displayed in the Event Manager (Refer to **CHAPTER 6 WINDOW REFERENCE**). The setting status of all the event conditions created by using the event conditions or event link condition are also managed or displayed in the Event Manager.

In the Event Manager (Refer to **CHAPTER 6 WINDOW REFERENCE**), a mark indicating the type of each event condition, and an icon indicating an event name specified for registration are displayed.

**(5) Setting each event condition**

The event conditions shown in Table 4-1 are created in each setting dialog box. Each event condition is created by dragging and dropping the event condition registered in the Event Manager, or the icon of an event link condition, on to the condition item in each setting dialog box.

A created event condition is “set” by clicking the function button  or  in the setting dialog box, or by clicking the mark of the event icon in the Event Manager and display it in red. By setting a created event condition, a debug action occurs as an event condition.

For all the event conditions it is possible to “register” up to 256 conditions, but the number of conditions that can be “set” differs depending on each event condition.



## 4.4 Break Function

The break functions stop the execution of the user program and the operation of the tracer<sup>Note</sup> by the CPU.

**Note** The trace function is valid only for CPU cores with the TRCU incorporated.

The following five types of break functions are available.

- Event detection break
- Break by Come function
- Software break
- Break on satisfaction of condition during step execution
- Forced break

When the Come, Step In, Return Out, or Next Over command of the program execution functions is executed, event detection break and detection of software break are not performed.

Each break function is explained below.

### (1) Event detection break

An event detection break is a function to stop the execution of the user program by detecting a set break event condition. This function is used to stop a user program executed by the Go, Go & Go, and Restart commands.

When the Go & Go command has been executed, the contents of each window are updated and the user program is then executed again after an event detection break has been implemented. Set break events as follows:

- Set a break event in the Source window or Assemble window as an execution event.
- Set a break event in the Break dialog box (Refer to **CHAPTER 6 WINDOW REFERENCE**).

### (2) Break by Come function

The break by Come function is a function to stop a user program executed by the Come command by detecting an address specified on Source window or Assemble window. After user program execution has been stopped, the breakpoint by Come function is eliminated.

While the user program is being executed using this function, the currently set break event does not occur.

### (3) Software break

A software break is a function that replaces the specified address instruction with a debugger-dedicated software break instruction, and is used to stop a user program executed by the Go, Go & Go, or Restart command.

Consequently, although there is no limit to the number of breaks that can be set (only 100 are valid, however), because the software break instruction replaces the specified address instruction, it is not possible to set breaks in the external ROM, or to stop the program using the variable access timing.

While event detection breaks use one hardware resource for one event condition, a software break can set a breakpoint to two or more addresses.

A software break can be set in the Source or Assemble window.

Software breaks are controlled by the Software Break Manager (Refer to **CHAPTER 6 WINDOW REFERENCE**).

### (4) Break on satisfaction of condition during step execution

This is a function to stop the execution of the user program when the end condition of each command (Step In, Next Over, Return Out, or Slowmotion) is satisfied.

Because one instruction at a time is executed, stopped, and condition checked, the processing time of this function is slower than that of real-time execution.

**(5) Forced break**

A forced break is a function to forcibly stop the execution of the user program. This function is valid for all the program execution commands. A forced break is implemented by the following command.

- Stop command  
Forcibly stops the execution of the user program.

## 4.5 Trace Function

The trace function<sup>Note</sup> is to trace the branch instruction of the user program and to save the execution flow in the trace memory. The trace data saved in the trace memory can be displayed on the Trace View window (Refer to **CHAPTER 6 WINDOW REFERENCE**).

**Note** The trace function is valid only for CPU cores with the TRCU incorporated.

### 4.5.1 Trace operations

The tracer operation of the ID850NW is set in the Trace Option dialog box.

In this dialog box it is possible to set the mode for when fetching trace data, select the trace data type, specify the address at which to perform an access data trace, and set the execution trace source.

Any size can be specified for the trace memory in the Extended Option dialog box.

In addition, if a trace exceeding the number of the trace memory frames is performed, the operation of the tracer can be selected from the following: overwrite the previous frame, stop the tracer at that timing, or stop the tracer when a delay trigger event occurs.

The trace function is valid as the default. To invalidate the trace function, remove the checkmark from [Tracer ON] on the [Option] menu.

### 4.5.2 Trace condition setting function

The condition under which trace is to be executed can be set by the following procedures.

#### (1) Trace fetch mode setting

The following two modes are available to fetch trace data.

These settings are made in the Trace Option dialog box.

**Table 4-4. Types of Trace Mode**

Trace Mode	Trace Description
Real-time trace mode (Realtime Mode)	Traces the CPU execution in real time. However, not all the trace data may be fetched, depending on the program or trace source (default).
All trace mode (Non Realtime Mode)	If trace data is likely to be lost, the CPU execution is internally stopped temporarily and all the trace data is properly fetched.

#### (2) Trace data setting

Either of the following can be selected as the trace data to be fetched.

These settings are made in the Trace Option dialog box.

**Table 4-5. Types of Trace Data to Be Fetched**

Trace Data	Description
Execution data	Traces the specified branch source to ascertain the program execution flow.
Access data	Traces the data access to the specified address and data access cycle in the specified address range.

**(3) Trace mode setting**

The following two trace modes can be set.

These settings are made by selecting [Run] menu → [Uncond. Trace ON]/[Cond. Trace ON] from the main window (refer to **CHAPTER 6 WINDOW REFERENCE**).

**Table 4-6. Types of Trace**

Trace Mode	Trace Contents
Unconditional trace	Unconditionally traces all execution processes of user program (default).
Conditional trace	Traces only specified zone (selected when using section trace).

**(4) Tracer control mode setting**

The following two tracer control modes are available.

These settings are made by selecting [Run] menu → [Tracer Control Mode] from the main window (refer to **CHAPTER 6 WINDOW REFERENCE**).

**Table 4-7. Types of Tracer Control Mode**

Tracer Control Mode	Trace Description
Non stop	The data in the oldest frame is overwritten after the total trace memory has been traced (default).
Full stop	The tracer is stopped after the total trace memory has been traced.
Delay trigger stop	The tracer is stopped after tracing for the delay count when a delay trigger event occurs.

**(5) Trace event condition setting**

A trace event condition is a condition under which trace execution is started/ended if the conditional trace is selected as the trace mode.

Section trace and delay trace can be selected for the ID850NW.

The trace event condition can be set on the Trace dialog box (refer to **CHAPTER 6 WINDOW REFERENCE**). Up to 256 trace conditions can be registered, of which only one can be set.

**(a) Section trace**

In this mode, trace is started when a specified trace start condition is satisfied, and ends when a specified trace end condition is satisfied.

**(b) Delay trace**

A trace can be performed for the specified delay count after the delay trigger event is made by setting the delay count in the Delay Count Setting dialog box and setting the delay trigger event.

### 4.5.3 Trace result display function

The result of a trace can be checked in the Trace View window (Refer to **CHAPTER 6 WINDOW REFERENCE**). The Trace View window displays each frame of the trace contents of the following items shown in Table 4-8.

**Table 4-8. Trace Data Display Contents**

Displayed Item	Contents
Frame	Frame number of trace memory indicating time sequence
Time	Execution time of each frame
Address	Fetch address
Data	Fetch data
Status	Fetch status
Address	Access address
Data	Access data
Status	Access status
DisAsm	Disassemble result

Displayed Item and Contents can be selected by opening the Trace Data Select dialog box in [View] → [Select...] in the main window.

## 4.6 Load/Save Functions

The load/save functions are to load and save each file.

The ID850NW individually loads/saves each file. The following two types of files are loaded/saved.

- Display files: Files recording the screen information when they are saved. By loading these files, the reference file is opened in the Source window.
- Information files: Files recording various set data and debugging environments as the internal information of the ID850NW.

### 4.6.1 Display files

Table 4-9 lists the display files handled by the ID850NW.

**Table 4-9. Display Files Handled by ID850NW**

File Name (Extension)	Outline	Operation Window
Source text display file (.SVW)	Shows source text in Source window	View File Load dialog box View File Save dialog box
Variable display file (.WCH)	Shows variable information in Watch window	
Assemble display file (.DIS)	Shows assemble information in Assemble window	
Memory display file (.MEM)	Shows memory information in Memory window	
Register display file (.RGW)	Shows register information in Register window	
Stack trace display file (.STK)	Shows stack trace information in Stack window	
I/O register display file (.IOR)	Shows I/O register information in IOR window	
Local variable display file (.LOC)	Shows local variable information in Local Variable window	
Trace display file <sup>Note</sup> (.TVW)	Shows trace information in Trace View window	
Event set file (.EVN)	Stores event set information of Event Manager	

**Note** The trace function is valid only for CPU cores with the TRCU incorporated.

### 4.6.2 Information files

Table 4-10 lists the information files handled by the ID850NW.

**Table 4-10. Information Files Handled by ID850NW**

File Name (Extension)	Outline	Operation Window
Object file (.HEX)	Stores object code of user program	Download dialog box Upload dialog box
Load module file (.OUT)	Stores object code of user program, symbol, and source information	Download dialog box
Project file (.PRJ)	Stores debugging environment	Project File Load dialog box Project File Save dialog box
Binary file (.BIN)	Stores data of binary format	Download dialog box Upload dialog box

The project file sets window/dialog box display information, size, and display area when it is loaded to restore the previous debugging environment (the status when the project file was saved).

#### 4.7 Register Manipulation Functions

The register manipulation functions are for displaying or changing the contents of the program registers, system registers, and peripheral I/O registers.

These functions can be used in the Register window or IOR window (Refer to **CHAPTER 6 WINDOW REFERENCE**).

#### 4.8 Memory Manipulation Functions

The memory manipulation functions are for displaying or changing the contents of the memory by using mnemonic, hexadecimal, and ASCII codes. These functions can be used in the Memory window or Assemble window (Refer to **CHAPTER 6 WINDOW REFERENCE**).

#### 4.9 Real-Time RAM Sampling Function

Real-time RAM sampling is a function used to read memory and update the displayed contents in real time, even during user program execution. All or specified 1 KB of internal RAM area can be subject to real-time RAM sampling. The sampled areas are displayed in the Memory or Watch window in real time.

This function is realized by software emulation, so user program execution is temporarily stopped.

The details for real-time RAM sampling can be set on the Extended Option dialog box (Refer to **CHAPTER 6 WINDOW REFERENCE**).

[MEMO]



## CHAPTER 5 ASSOCIATION WITH PROJECT MANAGER

This chapter explains the procedure and notes on using the functions associated with the Project Manager. The ID850NW can automatically perform a sequence of tasks such as creating a program, compiling, debugging, and correcting the program, in association with the Project Manager (PM).

To debug a load module file created in the PM at the source level, confirm that the item [Option] → [Debug] of the PM is checked for building.

**Caution** When a load module file is created by using an MS-DOS™ prompt, the functions of the ID850NW associated with the PM cannot be used.

### 5.1 Debugger Registration in PM Project

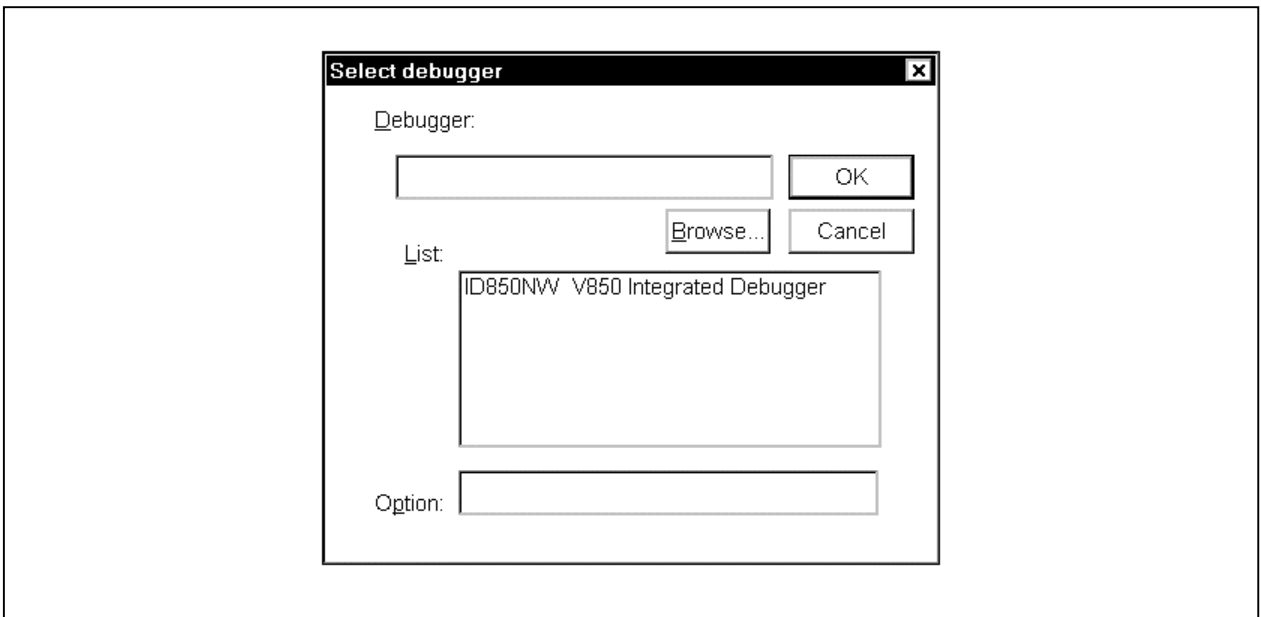
Every debugger for ID850NW can be specified for each project in the PM.  
The selection procedure is explained below.

#### 5.1.1 Debugger selection

The procedure for selecting the debugger is as follows.

- <1> Select [Project] → [New] in the PM and create the new project for PM. Alternatively, read an existing PM project by selecting [Project] → [Open...].  
→ For details of the setting items, refer to the on-line help or User's Manual.
- <2> Select [Option] → [Select Debugger...] from the PM menu bar.  
→ This opens the Select Debugger Type dialog box. Figure 5-1 shows the Select Debugger Type dialog box.

Figure 5-1. Select Debugger Type Dialog Box (PM)



- <3> Either click "ID850NW V850 Integrated Debugger" by mouse from the debugger list, or specify the execution file name ID850n32.EXE (with path) for the debugger file name.
- <4> Click .
- ID850NW will now be registered as the project debugger of the open PM.  
The ID850NW icon will also be displayed on the PM toolbar.

## 5.2 Starting Up ID850NW from PM

One of the following three methods can be used to start up the ID850NW from the PM.

- (1) Click the ID850NW startup button on the PM toolbar.
- (2) Select **[Build]** → **[Debug]** from the PM menu bar.
- (3) Select **[Build]** → **[Build and Debug]** from the PM menu bar. Alternatively, click the batch-processing button on the PM toolbar (however, the ID850NW will start up only when Build has finished normally).

### 5.2.1 Reproducing debug environment

By following the procedure shown below, it is possible to reproduce the debug environment of the previous time the ID850NW was started up from the PM.

- <1> Create a new project file<sup>Note</sup> with the PM (eg: aaa.prj)

**Note** Use the project file to save and reference the environment data for the debugger (ID850NW) and the PM. The project file extension when dealing with the debugger and PM is prj. Refer to the online help or user's manual of each product regarding the data that is saved and reproduced using a project file.

- <2> Start up the ID850NW from the PM. Because the project file is new, set items other than the device classification in the Configuration dialog box, in the same way as when starting up the ID850NW individually.
- <3> Debug the target device with the ID850NW.
- <4> When the ID850NW has finished debugging, check **[Save project file]** in the Exit Debugger dialog box and exit the ID850NW.
  - The debug environment at the completion of ID850NW debugging is saved to the aaa.prj file (the debug environment can also be saved to the aaa.prj file at times other than the completion of ID850NW debugging).
- <5> When the ID850NW is next started up after the aaa.prj file is read by the PM, the debug environment saved in <4> above is automatically reproduced.
  - The project file currently being used by the PM (aaa.prj) is automatically read when the ID850NW is started up.

### 5.3 Correction of Source File and Auto Load

If a bug is found as a result of debugging using the ID850NW, compiling and re-downloading can be automatically executed as well as editing the source file with the ID850NW started with the following procedure.

**Caution** If [Option] → [Project Manager Options...] is selected in the PM, and if an editor other than the standard editor is selected, this processing cannot be performed.

- <1> Read the source file to correct with the ID850NW to the Source window. With the Source window displayed at the foremost position, select [File] → [Open...] on the main window to specify the corresponding file (if the corresponding file has already been displayed in the Source window, activate the window).  
→ The corresponding file is displayed in the Source window.
- <2> Select [Edit] → [Edit Source] in the ID850NW main window.  
→ The corresponding file is automatically read to the standard editor and is opened.
- <3> Correct the source file in the editor.
- <4> Select [File] → [Save as...] in the editor and save the file.
- <5> Select [File] → [Exit] in the editor to terminate the editor or select [Build].  
→ As a result, the PM automatically executes compiling, assembling, and linking in accordance with the settings of the project. When these operations have been completed normally, the created load module file is automatically downloaded to the ID850NW.

- Cautions**
1. When the load module file is automatically downloaded, CPU reset is not executed.
  2. The debug window, which was opened when the editor was called, and all event settings are restored. However, if the previously used lines and symbols have been deleted because of source file correction, observe the following points.
    - The software breakpoint may be deleted.
    - The variables that were displayed as variable display will be displayed in gray.
    - In event conditions, the event mark will be displayed in yellow.

[MEMO]

## CHAPTER 6 WINDOW REFERENCE

This chapter explains in detail the functions of the windows and dialog boxes of the ID850NW. For the configuration and details of the types of the windows and dialog boxes, refer to **APPENDIX B TERMINOLOGY**.

### 6.1 Window List

Table 6-1 lists the windows of the ID850NW debugger.

**Table 6-1. Windows of ID850NW Debugger (1/2)**

Window Name	Outline	Page
Main window	This window is displayed first when the debugger is started. It controls execution of the user program. In addition, various windows are opened from this window.	p.56
Configuration dialog box	Sets/displays debugger environment	p.81
Extended Option dialog box	Sets/displays each extended option	p.86
Debugger Option dialog box	Sets/displays each option	p.90
Interrupt Mask dialog box	Sets mask for external maskable interrupts	p.98
Font dialog box	Sets the type and size of the display font in the Source window	p.100
Project File Load dialog box	Reads debugging environment	p.103
Project File Save dialog box	Saves debugging environment	p.106
View File Load dialog box	Reads the display files of each window and the source text and event setting files	p.109
View File Save dialog box	Saves the display files of the current window and the event setting files	p.112
Download dialog box	Downloads the object files and binary files	p.118
Load Module List dialog box	Displays the downloaded load module file names and file path names in a list	p.122
Upload dialog box	Saves the memory contents to a file	p.124
Browse dialog box	Selects the files to be set in the Debugger Option and Source Search dialog boxes	p.127
Source Move dialog box	Specifies the files to be displayed in the Source window and the display start position	p.129
Address Move dialog box	Specifies the display start address for the Memory and Assemble windows	p.132
Trace Move dialog box <sup>Note</sup>	Specifies the display start position for the Trace View window	p.134
Symbol To Address dialog box	Displays the addresses and symbol values of variables and functions	p.137
Source window	Displays source file or text file	p.140
Source Search dialog box	Searches a character string in a source file	p.154
Assemble window	Displays disassemble text and executes on-line assemble	p.157
Assemble Search dialog box	Searches the contents of the Assemble window	p.166
Memory window	Displays/changes memory contents	p.169
Memory Search dialog box	Searches the contents of the Memory window	p.174
Memory Fill dialog box	Initializes the memory contents by specified data	p.177
Memory Copy dialog box	Copies memory contents	p.179

**Table 6-1. Windows of ID850NW Debugger (2/2)**

Window Name	Outline	Page
Memory Compare dialog box	Compares memory contents	p.181
Memory Compare Result dialog box	Displays result of comparison of memory contents	p.183
DMM dialog box	Sets address and data to perform DMM (Dynamic Memory Modification)	p.185
Watch window	Displays/changes variables	p.188
Quick Watch dialog box	Displays/changes variable values temporarily	p.193
Add Watch dialog box	Registers variables to be displayed in the Watch window	p.198
Register window	Displays/changes register (program register, system register) contents	p.201
Register Select dialog box	Selects register displayed on the Register Window	p.204
IOR window	Displays/changes contents of peripheral I/O register	p.206
IOR Select dialog box	Selects peripheral I/O registers and I/O ports to be displayed in the IOR window	p.209
Add I/O Port dialog box	Registers I/O ports to be displayed in the IOR window	p.213
Local Variable window	Displays/changes local variable in current function	p.216
Stack window	Displays/changes the stack contents of the current user program	p.218
Trace View window <sup>Note</sup>	Displays trace result	p.222
Trace Data Select dialog box <sup>Note</sup>	Selects items to be displayed in the Trace View window	p.228
Trace Option dialog box <sup>Note</sup>	Sets various settings for trace	p.231
Event Manager	Displays, switches enable/disable, or deletes each event	p.235
Software Break Manager	Displays, switches enable/disable, or deletes the software break	p.242
Event dialog box	Registers/displays event condition	p.245
Event Link dialog box	Registers/displays event link condition	p.256
Break dialog box	Registers/sets/displays break event condition	p.266
Trace dialog box <sup>Note</sup>	Registers/sets/displays trace event condition	p.273
Delay Count Setting dialog box <sup>Note</sup>	Sets the delay count of the trace	p.281
Reset Debugger dialog box	Initializes ID850NW and target CPU and symbol information	p.283
About dialog box	Displays version of ID850NW	p.285
Exit Debugger dialog box	Ends ID850NW	p.286
Error/Warning dialog box	Displays error/warning message	p.288

**Note** The windows/dialog boxes related to trace are valid only for CPU cores with the TRCU incorporated.

## 6.2 Explanation of Windows

This section explains each window using the format shown below.

Window Name	Window type
-------------	-------------

The window name and window type (mode type when dialog box) are indicated in the above frame.

### General

Briefly explains the window.

### Opening method

How to open the windows is explained below.

### Window

Shows the screen image of the configuration of the window.

### Function

Explains each function of the window.

### Function buttons

Explains the operations of the function buttons in the window.

### Menu bar

Enumerates the menus pulled down from the menu bar, and explains the function of each menu.

### Caution

Explains points of caution to be noted.

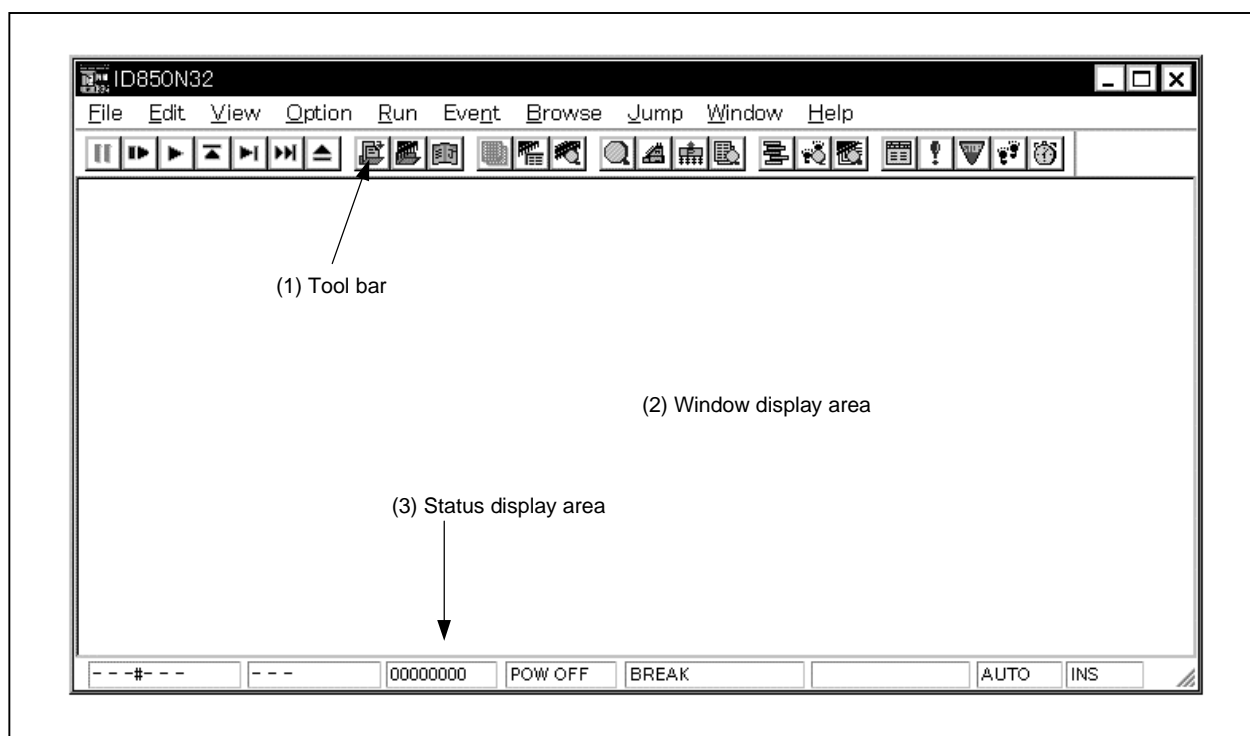
<b>Main Window</b>	<b>Main window</b>
--------------------	--------------------

### General

This window is automatically opened after the ID850NW is started and initialization is complete. It remains displayed until the ID850NW is terminated. Various windows are opened and manipulated from this window. Execution of the user program is also controlled in this window.

### Window

**Figure 6-1. Main Window**



### Function

The Main window consists of the following areas.

- Toolbar
- Window display area
- Status display area

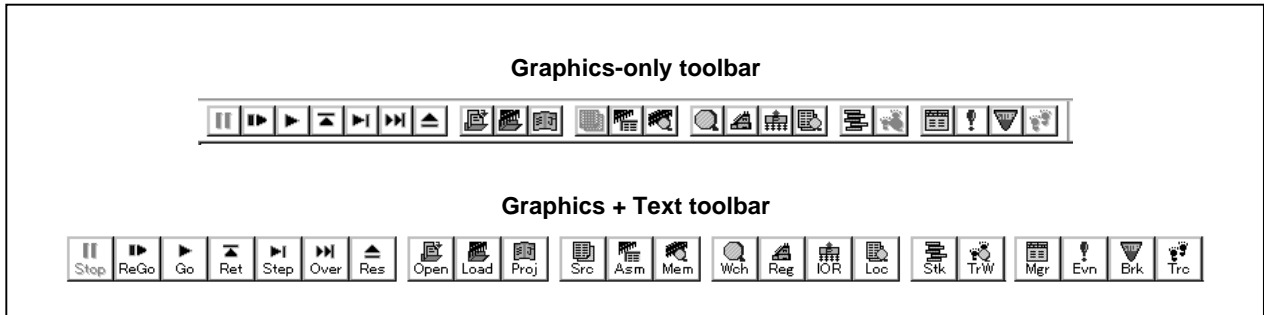


Each function is explained below.

### (1) Toolbar

The toolbar consists of buttons that can execute frequently used commands with a single click. Each button is graphically displayed and easy to understand. This button group can be switched between “Graphics-only buttons” and “Graphics + Text buttons” by the Debugger Option dialog box. The function of each button can be also executed using the menu bar in this window. This toolbar is not shown when it is specified not to be displayed by selecting [Option] → [Tool Bar] from the menu bar of this window.

Figure 6-2. Toolbar



An explanation of each button is given below.



This button stops execution of the user program.  
The same function as [Run] → [Stop] on the menu bar.



This button runs the user program after the CPU is reset.  
The same function as [Run] → [Restart] on the menu bar.



This button runs the user program from the current PC.  
The same function as [Run] → [Go] on the menu bar.



This button executes the program real-time until execution returns to the calling function.  
The same function as [Run] → [Return Out] on the menu bar.  
This command is for functions described in C language.



This button is used for step execution (execution of program instructions one by one).  
If functions or subroutines are called, instructions inside them are executed one by one.  
The same function as [Run] → [Step In] on the menu bar.



This button executes the next step (Executes the program, regarding the function/call statement as one step.).  
If functions or subroutines are called, instructions inside them are removed from step execution.  
The same function as [Run] → [Next Over] on the menu bar.



This button resets the CPU.  
The same function as [Run] → [CPU Reset] on the menu bar.



This button opens the View File Load dialog box.  
The same function as [File] → [Open...] on the menu bar.



This button opens the Download dialog box.  
The same function as [File] → [Download...] on the menu bar.



This button opens the Project File Load dialog box.  
The same function as [File] → [Project] → [Open...] on the menu bar.



This button opens the Source window and displays the contents of the source text.  
The same function as [Browse] → [Source Text] on the menu bar.



This button opens the Assemble window and the user program is disassembled and displayed.  
The same function as [Browse] → [Assemble] on the menu bar.



This button opens the Memory window and displays the memory contents.  
The same function as [Browse] → [Memory] on the menu bar.



This button opens the Watch window and displays the watch contents.  
The same function as [Browse] → [Watch] on the menu bar.



This button opens the Register window and displays the contents of the registers.  
The same function as [Browse] → [Register] on the menu bar.



This button opens the I/O Register window and displays the contents of the peripheral I/O register.  
The same function as [Browse] → [I/O Register] on the menu bar.



This button opens the Local Variable window and displays the local variable contents.  
The same function as [Browse] → [Local Variable] on the menu bar.



This button opens the Stack window and displays the contents of the stack.  
The same function as [Browse] → [Stack Trace] on the menu bar.



This button opens the Trace View window and displays the result of trace<sup>Note</sup>.  
The same function as [Browse] → [Trace] on the menu bar.



This button opens the Event Manager.  
The same function as [Event] → [Event Manager] on the menu bar.



This button opens the Event dialog box and registers or sets events.  
The same function as [Event] → [Event...] on the menu bar.



This button opens the Break dialog box and registers, sets, or displays break events. The same function as [Event] → [Break...] on the menu bar.



This button opens the Trace dialog box and registers, sets, or displays trace events<sup>Note</sup>. The same function as [Event] → [Trace...] on the menu bar.

**Note** This button is valid only for CPU cores with the TRCU incorporated.

#### (a) Tool hint function

If the mouse cursor is pointed at any of the buttons on the toolbar, a tool hint pops up after a few seconds.

**Figure 6-3. Example of Tool Hint**



#### (b) Moving the toolbar

By dragging toolbar with the mouse, it can be moved to any desired position within or outside the Main window.

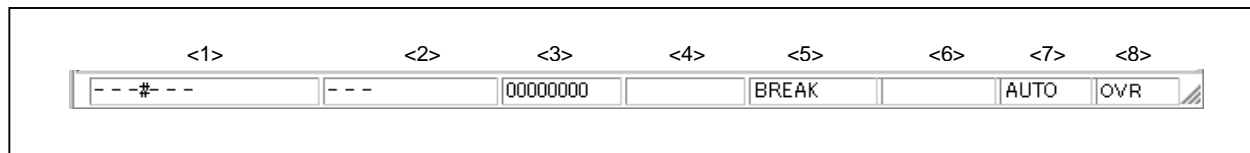
### (2) Window display area

This area displays various debug windows.

The displayed window can be changed in size or registered as an icon in this area. The list of the windows displayed in this area is as follows.

- Source window
- Watch window
- Register window
- Local variable window
- Trace View window<sup>Note</sup>
- Software Break Manager
- Assemble window
- Memory window
- IOR window
- Stack window
- Event Manager

**Note** Valid only for CPU cores with the TRCU incorporated.

**(3) Status display area****Figure 6-4. Status Bar**

This area displays the status of the debugger and in-circuit emulator.

<1> Program name \$ Source name # Line No.

Program name: Displays the program file name shown by the PC value.

Source name: Displays the source file name shown by the PC value.

Line No.: Displays the line number shown by the PC value.

<2> Function name: Displays the function name shown by the PC value.

<3> PC value: Displays the current PC value.

<4> CPU status: Displays the status of the CPU (target device).  
Displays the character "I" as a delimiter if there are multiple states.

<5> IE status: Displays the status of the in-circuit emulator (RUN, BREAK mode).  
Displays the character "I" as a delimiter if there are multiple states.

<6> Break cause: Displays cause of break.

<7> STEP mode: This is the area that displays the step execution mode.

<8> Key input mode: This area displays the key input mode.

The types of CPU statuses are shown below.

**Table 6-2. CPU Status Display Contents**

Status Indication	Meaning
HWSTOP	In hardware stop mode
RESET	Reset status
POW OFF	Power is not supplied to the target

The types of IE statuses are shown below.

**Table 6-3. IE Status Display Contents**

Status Indication	Meaning
RUN	Currently running a user program.
STEP	Currently executing a step.
TRC	Currently operating a trace <sup>Note</sup> .
BREAK	Break

**Note** The trace function is valid only for CPU cores with the TRCU incorporated.

The causes of a break are as follows.

**Table 6-4. Display Contents of Break Cause**

Displayed Cause	Meaning
Manual Break	Forcibly break
Temporary Break	Temporary break (displayed when [Come Here] and [Return] are executed)
Event Break	Break by event
Software break	Break by software break event

The types of STEP modes are shown below.

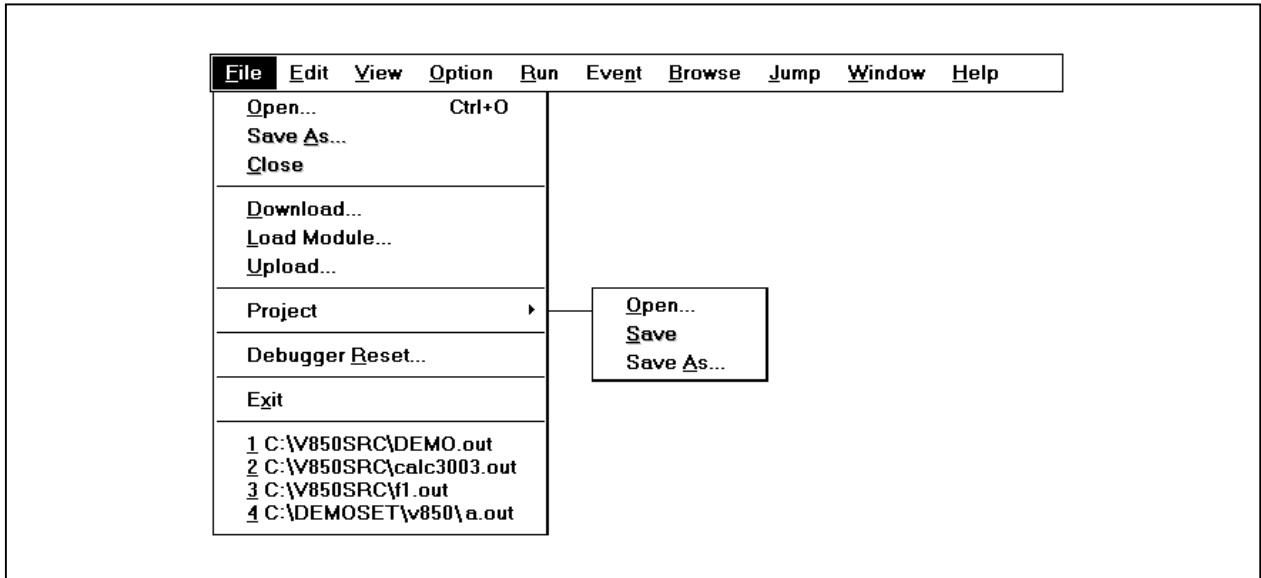
**Table 6-5. STEP Modes**

STEP Mode	Meaning
SRC	STEP execution is in the source mode. This is displayed when [Option] → [Source Mode] in the menu bar is selected.
INST	STEP execution is in the instruction mode. This is displayed when [Option] → [Instruction Mode] in the menu bar is selected.
AUTO	STEP execution is in the automatic mode. This is displayed when [Option] → [Auto Mode] in the menu bar is selected.

The types of key input modes are shown below.

Key Input Mode	Meaning
INS	Displayed when key input is in the insert mode.
OVR	Displayed when key input is in the overwrite mode.

## Menu bar

(1) FileFigure 6-5. File Menu Bar**[O]pen...**

This loads the display file or a source file or text file.

Opens the View File Load dialog box.

Operation differs depending on the file extension selected in the dialog box.

When a source file that symbol information is read:

Files are displayed in the Source window.

When an event setting file (.evn): The Event Manager is opened and the event is set.

When a source file that symbol information is not read, or display file:

Files are displayed in the Source window as text format file.

**[S]ave As...**

Saves the displayed contents of the current window in a display file under another name.

Opens the View File Save dialog box.

**[C]lose**

Closes the current window.

**[D]ownload...**

Downloads the user program.

Opens the Download dialog box.

**[L]oad Module...**

Displays a list of the names of files that have been downloaded.

Opens the Load Module List dialog box.

**[U]pload...**

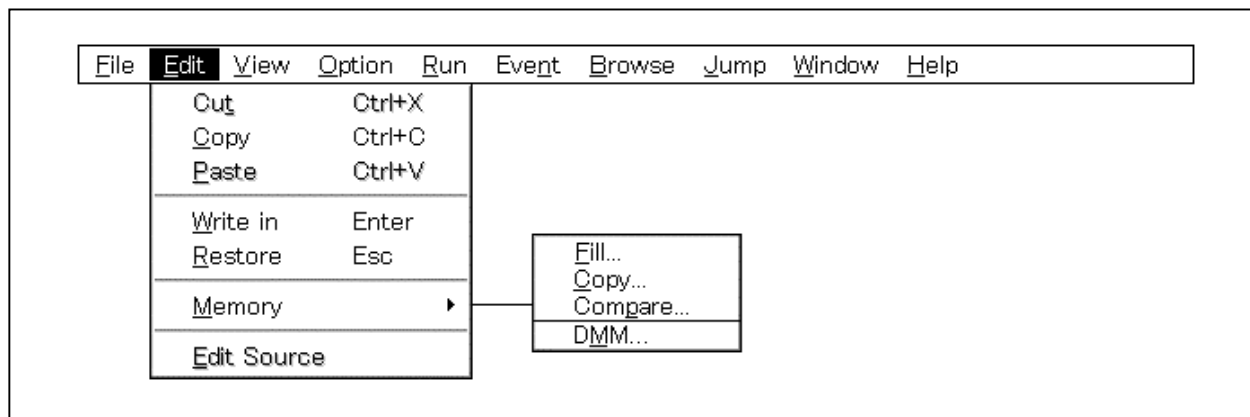
Uploads the user program.

Opens the Upload dialog box.

- [Project >]** Operates a project file.
- [Open...]** Opens the project file.  
Opens the Project File Load dialog box.
- [Save]** Overwrites the current status to the project file.  
The file to be overwritten is a project file that is currently read in the debugger.
- [Save As...]** Saves the current status in the project file.  
Opens the Project File Save dialog box.
- [Debugger Reset...]** Resets the target CPU, symbols and debugger.  
Opens the Reset Debugger dialog box.
- [Exit]** Terminates the debugger. Opens the Exit Debugger dialog box.
- [Open File]** Displays a list of the names of files that can be downloaded.  
If a file name is selected, that file is downloaded.

(2) **E**dit

**Figure 6-6. Edit Menu Bar**



- [Cut]** Cuts the selected character string and places it in the clipboard buffer.
- [Copy]** Copies the selected character string to the clipboard buffer.
- [Paste]** Pastes the contents of the clipboard buffer at the text cursor position.
- [Write in]** Writes corrections to the target.
- [Restore]** Cancels the correction.
- [Memory >]** Manipulates the memory contents.
  - [Fill...]** Initializes the memory contents.  
Opens the Memory Fill dialog box.

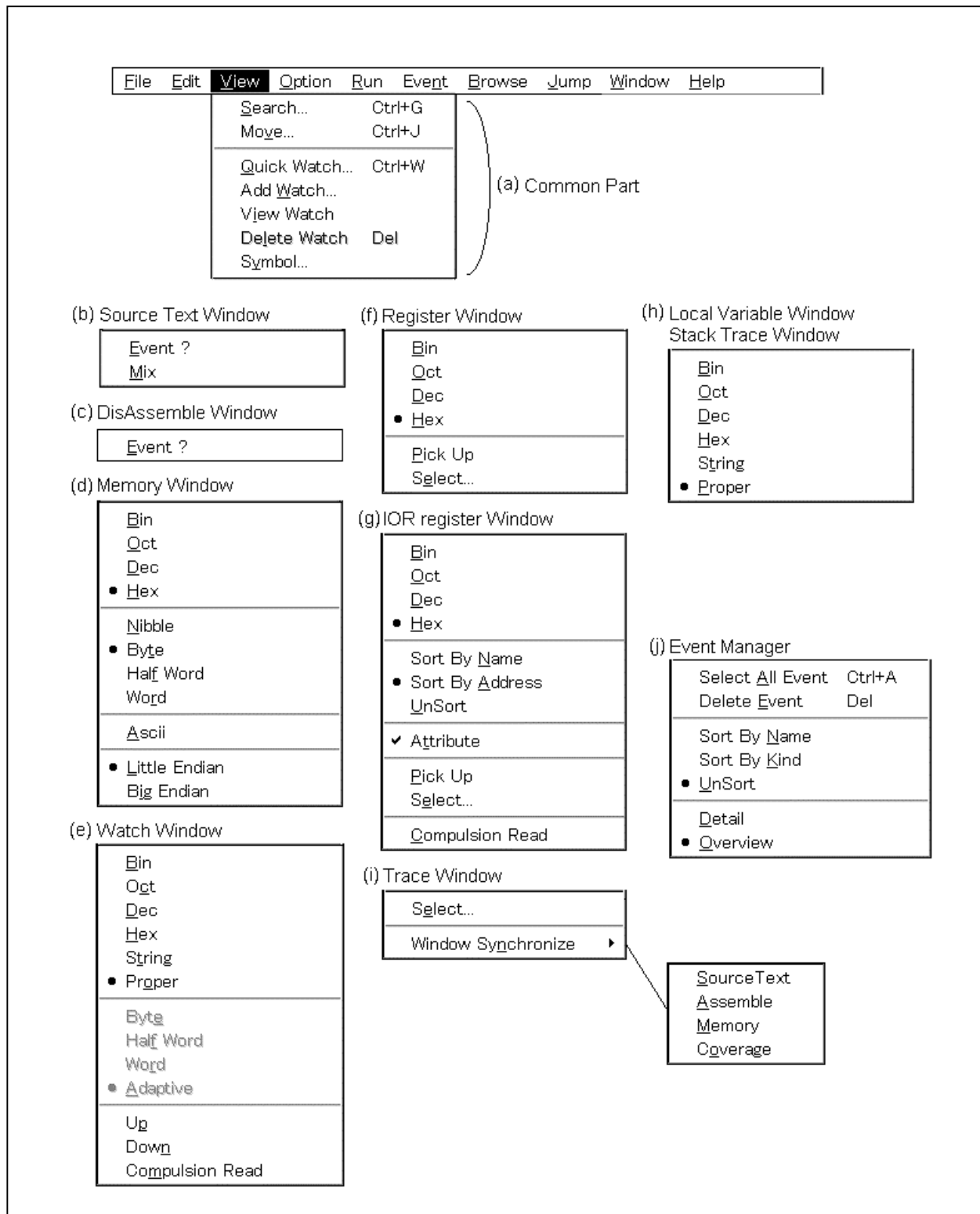


<b>[Copy...]</b>	Copies values from memory to memory. Opens the Memory Copy dialog box.
<b>[Compare...]</b>	Compares values between memories. Opens the Memory Compare Dialog box.
<b>[DMM...]</b>	Sets address and data to perform dynamic modification (DMM). Opens the DMM dialog box.
<b>[Edit Source]</b>	When the Project Manager is operating, opens the source file displayed in an active Source window by using the editor specified by the Project Manager.

(3) **View**

The items displayed in the display menu differ according to whether the window is active or not. An explanation of each dependent part is given in (b) to (j).

Figure 6-7. **View Menu Bar**



**(a) Common part**

<b>[Search...]</b>	Searches character strings and numeric values. Opens the search dialog box according to the current window. Performs the same operation as the <b>[Search]</b> button.
<b>[Move...]</b>	Moves the display position. Opens the Source Move dialog box, Address Move dialog box or Trace Move dialog box corresponding to the current window.
<b>[Quick Watch...]</b>	Displays the contents of the specified data temporarily. Opens the Quick Watch dialog box.
<b>[Add Watch...]</b>	Adds the specified data to the Watch window. Opens the Add Watch dialog box.
<b>[View Watch]</b>	Adds the selected data to the Watch window. In the case of a symbol, it adds it in accordance with the settings in the Debugger Option dialog box.
<b>[Delete Watch]</b>	Deletes the selected watch point from the Watch window. This menu is active only when variables are specified in the Watch window.
<b>[Symbol...]</b>	Displays the address of the specified variable or function and the value of the specified symbol. Opens the Symbol To Address dialog box.

**(b) Source window dependent part**

<b>[Event?]</b>	Displays the cursor position line or the address's event information. If an event is set, it opens the Event dialog box.
<b>[Mix]</b>	Switches between assemble display mixed display and no display. No check mark (" "): No mixed display (default). Check mark ("√"): Mixed display.

**(c) DisAssemble window dependent part**

<b>[Event?]</b>	Displays event information for the cursor position address. If an event is set, it opens the Event dialog box.
-----------------	--

**(d) Memory window dependent part**

<b>[Bin]</b>	Displays the current window in binary.
<b>[Oct]</b>	Displays the current window in octal.
<b>[Dec]</b>	Displays the current window in decimal.
<b>[Hex]</b>	Displays the current window in hexadecimal (default).

<b>[Nibble]</b>	Displays data in 4-bit units.
<b>[Byte]</b>	Displays data in 8-bit units (default).
<b>[Half Word]</b>	Displays data in 16-bit units.
<b>[Word]</b>	Displays data in 32-bit units.
<b>[Ascii]</b>	Turns on/off display of ASCII characters. No check mark ( " " ): Not displayed (default). Check mark ( "✓" ): Displayed.
<b>[Little Endian]</b>	The data is displayed in little endian (default).
<b>[Big Endian]</b>	The data is displayed in big endian.

**(e) Watch window dependent part**

<b>[Bin]</b>	The selected item is displayed as a binary number
<b>[Oct]</b>	The selected item is displayed as an octal number.
<b>[Dec]</b>	The selected item is displayed as a decimal number.
<b>[Hex]</b>	The selected item is displayed as a hexadecimal number.
<b>[String]</b>	The selected item is displayed as a character string.
<b>[Proper]</b>	The selected item is displayed as the standard value for each variable. In the case of symbols, they are displayed in accordance with the settings in the Debugger Option dialog box (default).
<b>[Byte]</b>	The selected item is displayed in 8-bit units.
<b>[Half Word]</b>	The selected item is displayed in 16-bit units.
<b>[Word]</b>	The selected item is displayed in 32-bit units.
<b>[Adaptive]</b>	The selected item is displayed as standard values for each variable. In the case of C language symbols, only this item is valid. In the case of assembler symbols, they are displayed in accordance with the settings in the Debugger Option dialog box (default).
<b>[Up]</b>	The selected data is moved 1 line up.
<b>[Down]</b>	The selected data is moved 1 line down.
<b>[Compulsion Read]</b>	The data in the peripheral I/O registers, where reading is prohibited because the values will change if they are read, the I/O ports added in the Add I/O Port dialog box, and the I/O protected areas is subjected to compulsory reading.

**(f) Register window dependent part**

<b>[Bin]</b>	Displayed as a binary number.
<b>[Oct]</b>	Displayed as an octal number.
<b>[Dec]</b>	Displayed as a decimal number.
<b>[Hex]</b>	Displayed as a hexadecimal number (default).
<b>[Pick Up]</b>	Displays only the register selected in the Register Select dialog box. No check mark (" "): All are displayed (default). Check mark ("√"): The selected register only is displayed.
<b>[Select...]</b>	Opens the Register Select dialog box.

**(g) IOR window dependent part**

<b>[Bin]</b>	Displayed as a binary number.
<b>[Oct]</b>	Displayed as an octal number.
<b>[Dec]</b>	Displayed as a decimal number.
<b>[Hex]</b>	Displayed as a hexadecimal number (default).
<b>[Sort By Name]</b>	Displayed in alphabetical order.
<b>[Sort By Address]</b>	Displayed in address order (default).
<b>[UnSort]</b>	Not sorted.
<b>[Attribute]</b>	Selects whether to display or not to display the Attribute field. No check mark (" "): Not displayed. Check mark ("√"): Displayed (default).
<b>[Pick Up]</b>	Only the peripheral I/O register selected in the IOR Select dialog box is displayed. No check mark (" "): All are displayed (default). Check mark ("√"): The selected register only is displayed.
<b>[Select...]</b>	Opens the IOR Select dialog box.
<b>[Compulsion Read]</b>	The data in the peripheral I/O registers, where reading is prohibited because the values will change if they are read, the I/O ports added in the Add I/O Port dialog box, and the I/O protected areas is subjected to compulsory reading.

**(h) Local Variable window dependent part, Stack window dependent part**

<b>[Bin]</b>	Displayed as a binary number.
<b>[Oct]</b>	Displayed as an octal number.
<b>[Dec]</b>	Displayed as a decimal number.
<b>[Hex]</b>	Displayed as a hexadecimal number.
<b>[String]</b>	Displayed as a character string.
<b>[Proper]</b>	Displayed as the standard value for each variable (default).

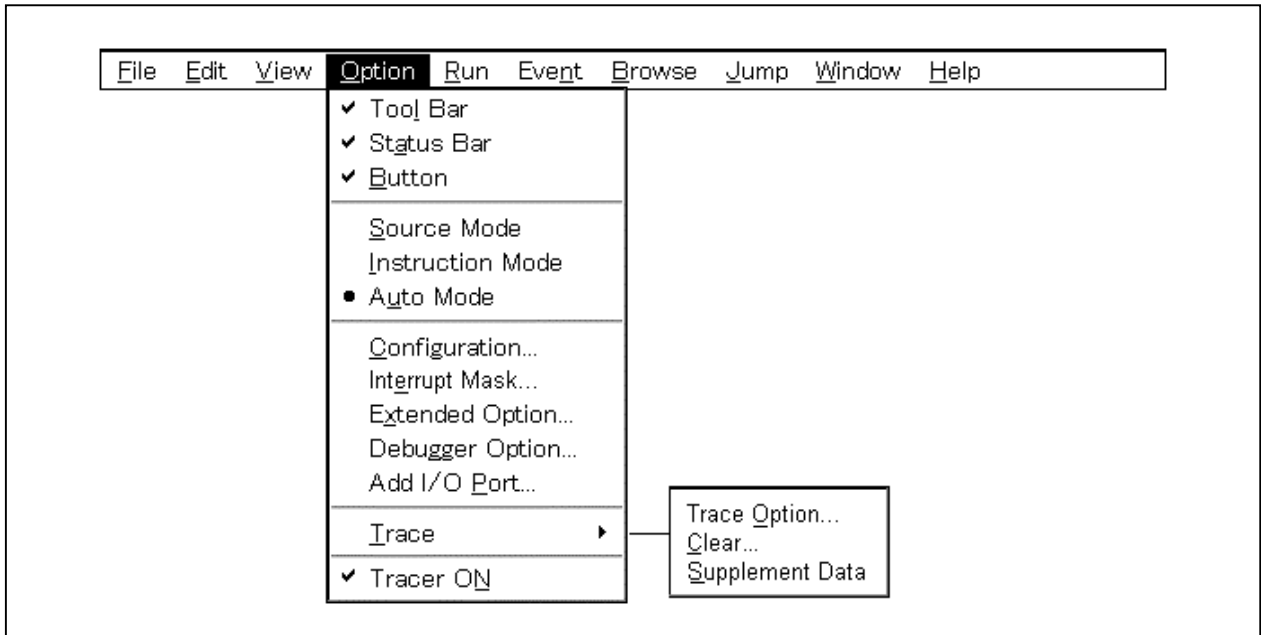
**(i) Trace View window dependent part**

This part is valid only for CPU cores with the TRCU incorporated.

<b>[Select...]</b>	Selects the content of the display. Opens the Trace Data Select dialog box.
<b>[Window Synchronize &gt;]</b>	This links together the Trace View window and Source window, Assemble window, and Memory window.
<b>[Source Text]</b>	Selects whether the Source window will be linked or not linked. No check mark (" "): Not linked (default). Check mark ("✓"): Linked.
<b>[Assemble]</b>	Selects whether the Assemble window will be linked or not linked. No check mark (" "): Not linked (default). Check mark ("✓"): Linked.
<b>[Memory]</b>	Selects whether the Memory window will be linked or not linked. No check mark (" "): Not linked (default). Check mark ("✓"): Linked.
<b>[Coverage]</b>	Selects whether the Coverage window will be linked or not linked. No check mark (" "): Not linked (default). Check mark ("✓"): Linked.

**(j) Event Manager dependent part**

<b>[Select All Event]</b>	Selects all the events entered in the Event Manager.
<b>[Delete Event]</b>	Deletes the selected event.
<b>[Sort By Name]</b>	Displays the icons sorted by event name.
<b>[Sort By Kind]</b>	Displays the icons sorted by event kind.
<b>[UnSort]</b>	Icons are not sorted (default).
<b>[Detail]</b>	Displays details.
<b>[Overview]</b>	Displays a list.

(4) **Option**Figure 6-8. **Option Menu Bar****[Tool Bar]**

Selects whether the toolbar is displayed or not.

No check mark (" "): Not displayed.

Check mark ("√"): Displayed (default).

**[Status Bar]**

Selects whether the status bar is displayed or not.

No check mark (" "): Not displayed.

Check mark ("√"): Displayed (default).

**[Button]**

Selects whether the function buttons on each window are displayed or not.

No check mark (" "): Not displayed.

Check mark ("√"): Displayed (default).

**[Source Mode]**

Performs step execution at the source level (line units).

**[Instruction Mode]**

Performs step execution at the instruction level (instruction units).

**[Auto Mode]**

Switches automatically between source level step execution and instruction level step execution, and executes step execution (default).

When the Source window is active, source level step execution (except in the mixed display mode) is performed, and when the Assemble window is active, instruction level step execution is performed.

When neither window is active, source level step execution is performed.

**[Configuration...]**

Sets the environment.

Opens the Configuration dialog box.

**[Interrupt Mask...]**

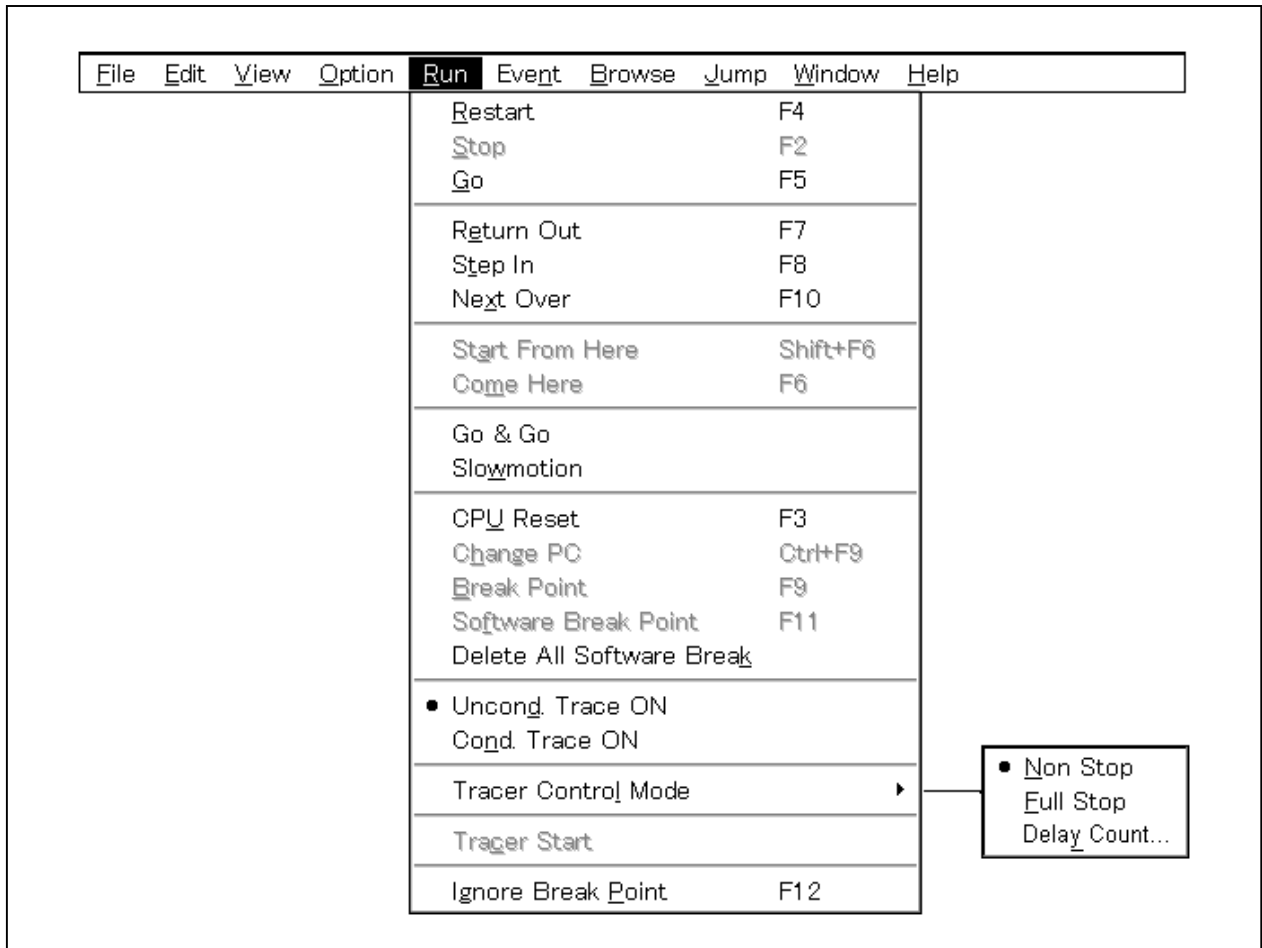
Sets the external interrupt mask.

Opens the Interrupt Mask dialog box.

<b>[Extended Option...]</b>	Sets the extended functions. Opens the Extended Option dialog box.
<b>[Debugger Option...]</b>	Sets the debugger. Opens the Debugger Option dialog box.
<b>[Add I/O Port...]</b>	Adds a user-defined I/O port. Opens the Add I/O Port dialog box.
<b>[Trace &gt;]<sup>Note</sup></b>	Sets various settings for trace.
<b>[Trace Option...]<sup>Note</sup></b>	Opens the Trace Option dialog box.
<b>[Clear...]<sup>Note</sup></b>	Clears the trace data.
<b>[Supplement Data]<sup>Note</sup></b>	Selects trace data supplement on/off. If this is checked, the debugger Supplements the already fetched data. To supplement correctly, select the all trace mode in the Trace Option dialog box to ensure that all the data is fetched (this is because all the data is necessary for data supplementation); otherwise, data may be not supplemented correctly. No check mark (""): Supplementation is not performed (default). Check mark ("√"): Supplementation is performed.
<b>[Tracer ON]<sup>Note</sup></b>	Selects whether the tracer is turned on or off. While the program is executed, this setting cannot be changed. No check mark (" "): Trace is not performed. Check mark ("√"): Trace is performed (default).


**Note** Valid only for CPU cores with the TRCU incorporated.




(5) RunFigure 6-9. Run Menu Bar**[Restart]**

This performs program execution after resetting the CPU.  
Operates the same as the  button.


**[Stop]**

Forcibly stops execution of the program. Performs the same operation as the  button.


**[Go]**





Executes the program from the current PC. Performs the same operation as the  button.

**[Return Out]**

Executes the program real-time until execution returns to the calling function.  
Performs the same operation as the  button.  
This command is for functions described in C language.

**[Step In]**

Executes instructions in the program one by one (step execution). Performs the same operation as the  button. If functions or subroutines are called, instructions inside them are executed one by one.

<b>[Next Over]</b>	Executes instructions in the program one by one (Next step execution). If functions or subroutines are called, instructions inside them are removed from step execution. Performs the same operation as the  button.
<b>[Start From Here]</b>	Executes the program from the cursor position of the Source window or Assemble window (Start function).
<b>[Come Here]</b>	Executes the program until the cursor position of the Source window or Assemble window (Come function).
<b>[Go &amp; Go]</b>	Continues execution of the program. If a break occurs because a break condition is satisfied, updates the window and then executes the user program again. Performs the same operation as clicking the  button each time a break occurs.
<b>[Slowmotion]</b>	Continues step execution of the program. After updating the window for each step execution, it performs step execution again. Performs the same operation as clicking the  button each time a break occurs.
<b>[CPU Reset]</b>	Resets the CPU. Performs the same operation as the  button.
<b>[Change PC]</b>	Sets the address at the cursor position of the Source window or Assemble window to PC.
<b>[Break Point]</b>	Sets/cancels a breakpoint at the cursor position of the Source window or Assemble window.
<b>[Software Break Point]</b>	Sets/cancels a software breakpoint at the cursor position of the Source window or Assemble window.
<b>[Delete All Software Break]</b>	Cancels all software breakpoints that are set.
<b>[Uncond. Trace ON]<sup>Note</sup></b>	Validates the unconditional trace function. Always traces while the user program is executed (default). When the tracer is started, the trace mode cannot be changed.
<b>[Cond. Trace ON]<sup>Note</sup></b>	Validates the trace function. Traces while the user program is executed according to a trace condition. When the tracer is started, the trace mode cannot be changed.
<b>[Tracer Control Mode &gt;]<sup>Note</sup></b>	Set the tracer mode from the following menu.
<b>[Non Stop]<sup>Note</sup></b>	The total trace memory is traced and then overwritten from the oldest frame (default).
<b>[Full Stop]<sup>Note</sup></b>	The total trace memory is traced and then the tracer is stopped.
<b>[Delay Count...]<sup>Note</sup></b>	The tracer is stopped after tracing for the delay count when a delay trigger event occurs.

**[Tracer Start or  
Tracer Stop]<sup>Note</sup>**

When trace measurement is stopped, this starts it.

When trace measurement is being performed, this stops it.

“Tracer Start”: Means that measurement is stopped. If selected, measurement will be started.

“Tracer Stop”: Measurement is in progress. If selected, measurement will be stopped.

If program is currently not executed, and during trace off (no “√” mark next to the [Option] → [Tracer ON] item), this item is invalid (it is displayed in gray).

Trace measurement starts operating immediately after trace has been turned on and program execution is started.

**[Ignore Break Point]**

This selects whether to make a breakpoint valid or invalid.

No check mark (“ ”): Valid (default)

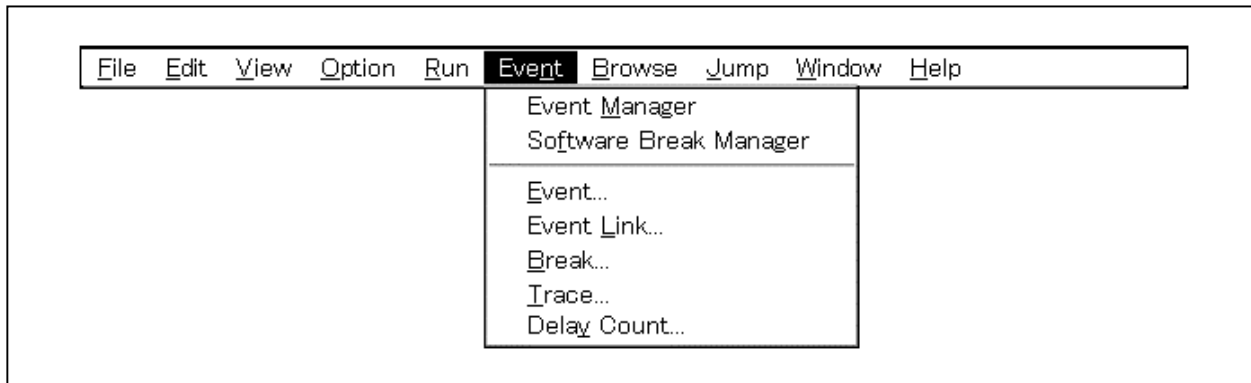
Check mark (“√”): Invalid





If it is made invalid, the breakpoint does not cause a hardware or software break at the specified point. During program execution, this setting cannot be switched.

**Note** Valid only for CPU cores with the TRCU incorporated.

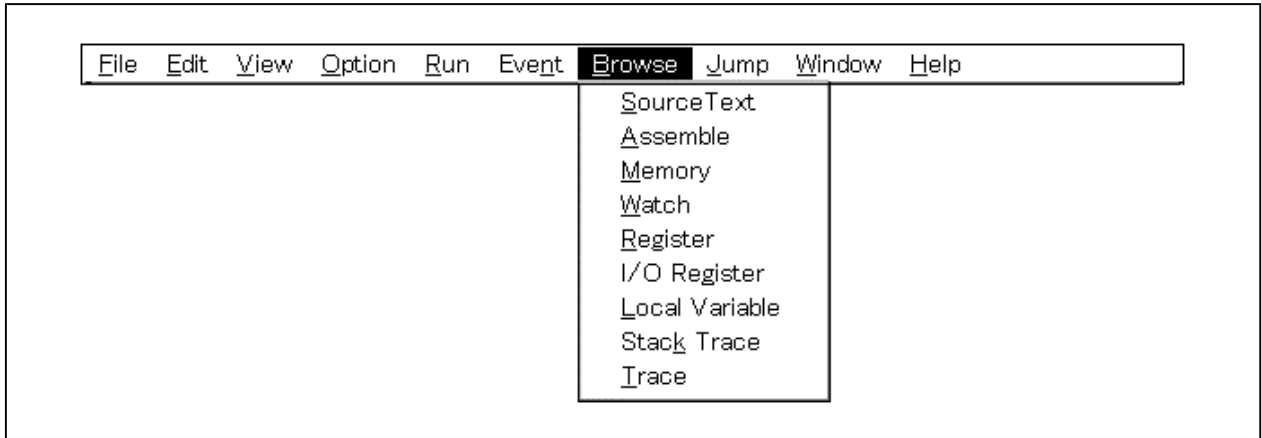
## (6) Event










Figure 6-10. Event Menu Bar



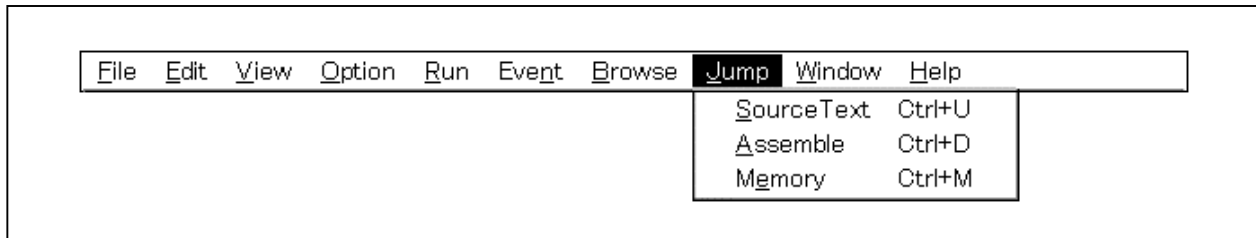
<b>[Event Manager]</b>	<p>Manages each type of event condition.          Opens the Event Manager.          Operates the same as the  button.</p>
<b>[Software Break Manager]</b>	<p>Manages software break conditions.          Opens the Software Break Manager.</p>
<b>[Event...]</b>	<p>Enters event conditions.          Opens the Event dialog box.          Operates the same as the  button.</p>
<b>[Event Link...]</b>	<p>Enters event link conditions.          Opens the Event Link dialog box.</p>
<b>[Break...]</b>	<p>Enters and sets break conditions.          Opens the Break dialog box.          Operates the same as the  button.</p>
<b>[Trace...]<sup>Note</sup></b>	<p>Enters and sets trace event conditions.          Opens the Trace dialog box.          Operates the same as the  button.</p>
<b>[Delay Count...]<sup>Note</sup></b>	<p>Sets the delay count and displays the delay value.          Opens the Delay Count Setting dialog box.</p>

**Note** Valid only for CPU cores with the TRCU incorporated.

(7) **B**rowse**Figure 6-11. Browse Menu Bar**

- [Source Text]** Displays the source text. Opens the Source window. Performs the same operation as the  button.  
If a window exists that is already in the active state, it opens the window in the static state.
- [Assemble]** Disassembles and displays the user program. Opens the Assemble window. Performs the same operation as the  button.  
If a window exists that is already in the active state, it opens the window in the static state.
- [Memory]** Displays the memory contents. Opens the Memory window. Performs the same operation as the  button.  
If a window exists that is already in the active state, it opens the window in the static state.
- [Watch]** Displays the watch contents (variable and other data). Opens the Watch window. Performs the same operation as the  button.
- [Register]** Displays the register contents. Opens the Register window. Performs the same operation as the  button.
- [I/O Register]** Displays the contents of the peripheral I/O registers. Opens the IOR window. Performs the same operation as the  button.
- [Local Variable]** Displays the local variable. Opens the Local Variable window. Performs the same operation as the  button.
- [Stack Trace]** Displays the stack contents. Opens the Stack window. Performs the same operation as the  button.
- [Trace]<sup>Note</sup>** Displays the result of trace. Opens the Trace View window. Performs the same operation as the  button.

**Note** Valid only for CPU cores with the TRCU incorporated.

**(8) Jump****Figure 6-12. Jump Menu Bar****[SourceText]**

Sets the data selected in the current window as the jump address, and displays the source text and source line starting from that address. No jump can be performed if the jump address contains no line information. Opens the Source window.

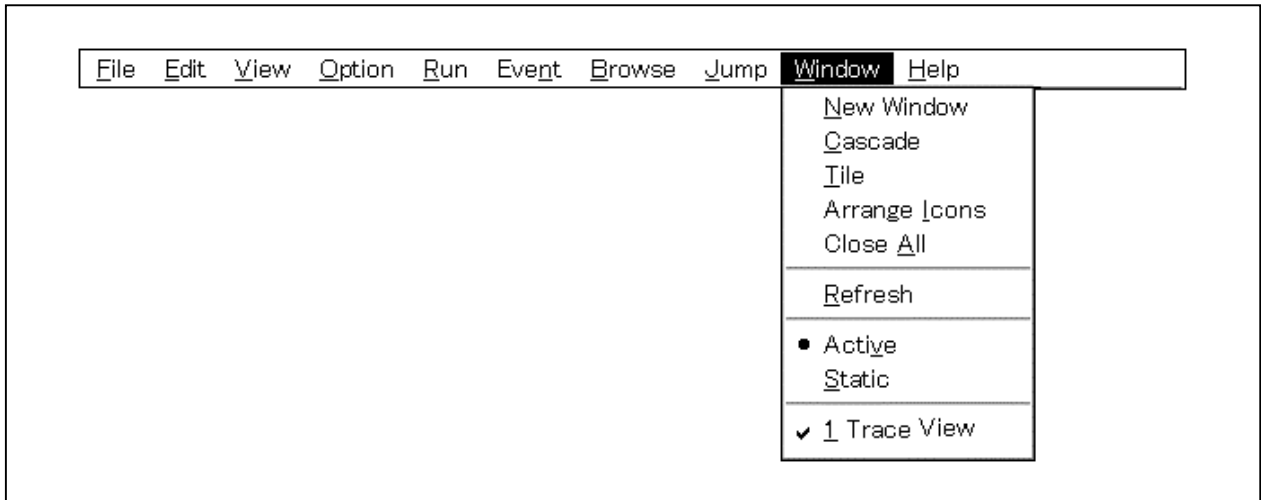
If the Source window is being displayed in the active state, that window is displayed on the front plane (operation object).

**[Assemble]**

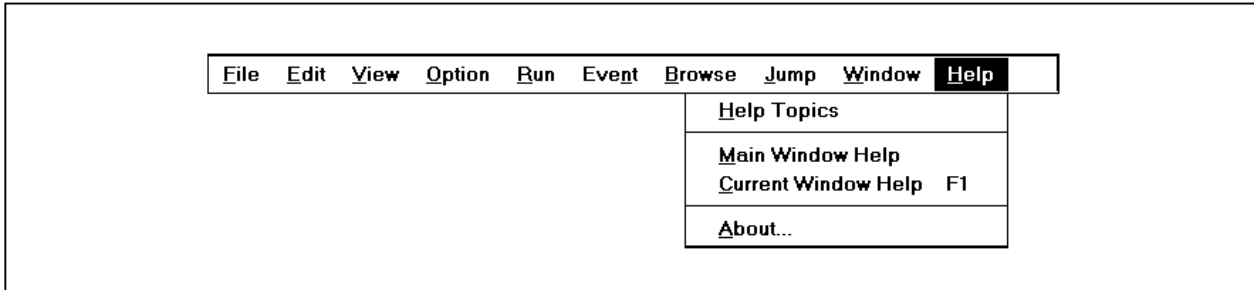
Sets the data selected in the current window as the jump address, and displays the disassembled text starting from that address. Opens the Assemble window. If the Assemble window is being displayed in the active state, that window is displayed on the front plane (operation object).

**[Memory]**

Sets the data selected in the current window as the jump address, and displays the memory contents starting from that address. Opens the Memory window. If the Memory window is being displayed in the active state, that window is displayed on the front plane (operation object).

(9) WindowFigure 6-13. Window Menu Bar

<b>[New Window]</b>	Opens a new window that displays the same contents as the current window. This is valid only in cases where the current window is the Source window, Assemble window, and Memory window.
<b>[Cascade]</b>	Cascades the windows in the main window.
<b>[Tile]</b>	Tiles the windows in the main window.
<b>[Arrange Icons]</b>	Rearranges the icons in the main window.
<b>[Close All]</b>	Closes all the windows except the main window.
<b>[Refresh]</b>	Updates the contents of the window with the latest data.
<b>[Active]</b>	Switches the window to the active state. If it is currently in the active state, a check mark is entered on the left of this item.
<b>[Static]</b>	Switches the window to the static state. If it is currently in the static state, a check mark is entered on the left of this item.
<b>[Open window]</b>	Lists the opened windows. The window with a check mark added at the side of the number is the current window. The window selected by selecting a window name is displayed as the current window.

(10) HelpFigure 6-14. Help Menu Bar

**[Help Topics]** Displays the table of contents of the Help window.

**[Main Window Help]** Displays the help for the main window.

**[Current Window Help]** Displays the help for the current window.

**[About...]** Displays the version of the debugger.



<b>Configuration Dialog Box</b>	<b>Modal</b>
---------------------------------	--------------

**General**

Displays and sets the operating environment of the in-circuit emulator.

This dialog box is displayed automatically after the debugger is started. To use the debugger, the operating environment of the in-circuit emulator must be set first in this dialog box.

When a project file is read, however, it doesn't need to be set. The result of reading the project file is reflected in the Configuration dialog box.

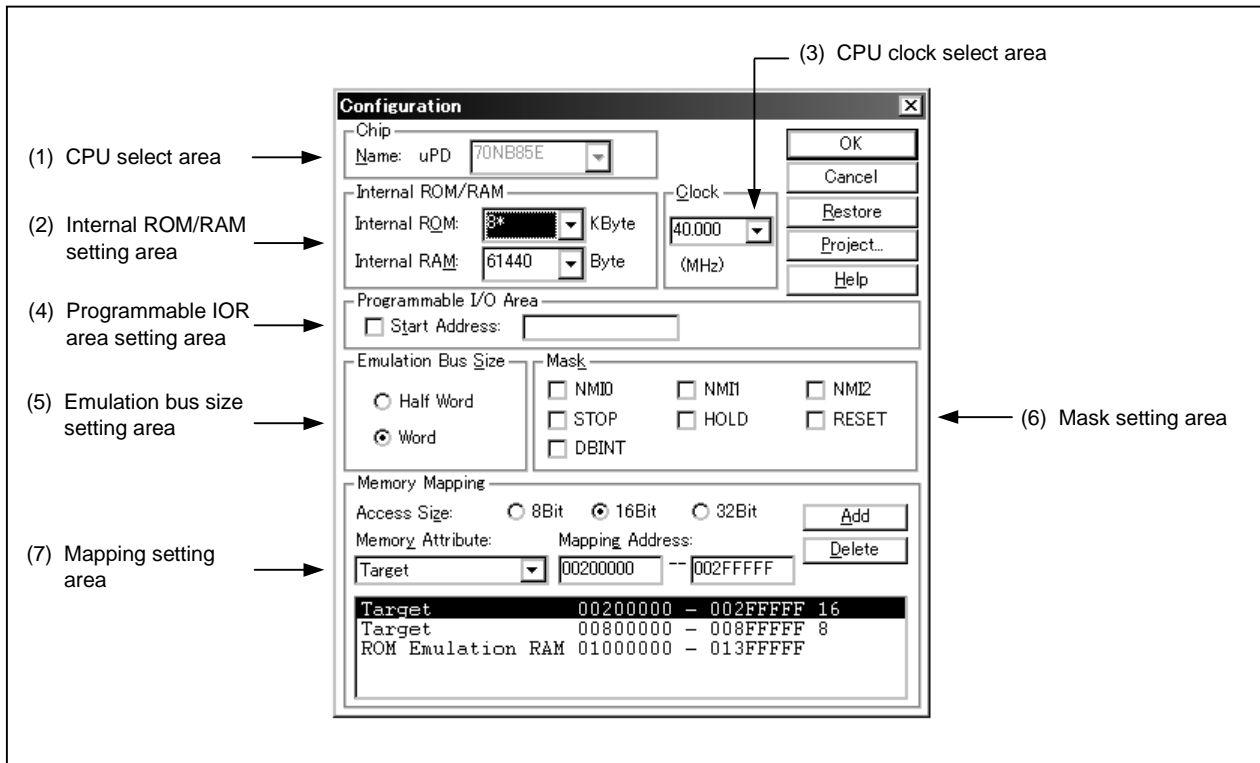
**Opening method**

This dialog box can be opened in the following methods.

- When the debugger is started  
Automatically opened
- In the main window  
Select [Option] → [Configuration...].  
Press the [GRPH] + [O], and [C] keys in that order.

**Window**

**Figure 6-15. Configuration Dialog Box**



<b>Function</b>
-----------------

The Configuration dialog box consists of the following areas.

- CPU select area
- Internal ROM/RAM setting area
- CPU clock select area
- Programmable IOR area setting area
- Emulation bus size setting area
- Mask setting area
- Mapping setting area

The function of each area is explained below.

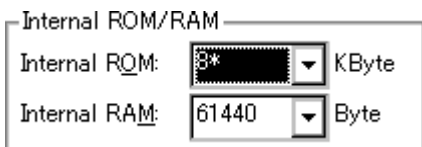
### (1) CPU select area



This area selects a CPU or core name. Note that this area can be selected only when the ID850NW is started.

A chip name can be specified by inputting a name using the keyboard or selecting one from the Name drop-down list.

### (2) Internal ROM/RAM setting area



This area displays the size of the internal ROM and RAM of the CPU.

The default size is automatically displayed with the asterisk (\*) attached at the end when the CPU is selected. To change the size of the internal ROM and RAM, select a size from the internal ROM and RAM drop-down list.

The actual memory configuration depends on the target.

If the selected size cannot be realized due to in-circuit emulator limitations, the debugger automatically adjusts the size. For that reason, if the Configuration dialog box is displayed again, it may be because the size being displayed is greater than the size specified the previous time.

The specifiable size for both alignment and setting range is 4096, 12288, 28672, and 61440 bytes.

**(3) CPU clock select area**

This area specifies the number of system clocks. The unit is MHz.

The number of clocks can be specified either via keyboard input, or selected from a drop-down list.

**(4) Programmable IOR area setting area**

This area is used to specify the start address of the programmable IOR area.

When using programmable IOR in devices that support this function, the start address of the programmable I/O area can be set by selecting the check box.

The address is aligned to 16 KB. Do not change the start address after startup.

**(5) Emulation bus size setting area**

This area sets the bus size of the area mapped in the ROM emulation RAM.

Although it is possible to reset after starting up the debugger, specify in accordance with the memory configuration on the target board.

**Half Word:** Accesses the area mapped by the ROM emulation RAM in halfwords (16 bits). Select when configuring a 16-bit bus using the ROM1 probe of the IE-70000-MC-NW-A.

**Word:** Accesses the area mapped by the ROM emulation RAM in words (32 bits). Select when configuring a 32-bit bus using both the ROM1 and ROM2 probes of the IE-70000-MC-NW-A.

**(6) Mask setting area**

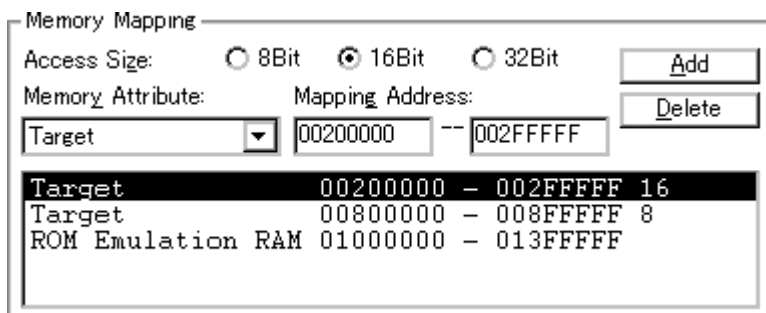
This area specifies a mask for the signal sent from the target.

The signal of the masked pin cannot be input to the CPU or core.

Only mask pins when the operation of the target is unstable at the debugging stage.

The maskable pins are as follows.

NMI0 pin, NMI1 pin, NMI2 pin, RESET pin, STOP pin, HLD pin, DBINT pin

**(7) Mapping setting area**


Memory Mapping

Access Size:  8Bit  16Bit  32Bit

Memory Attribute:  Mapping Address:  --

Target	00200000	-	002FFFFF	16
Target	00800000	-	008FFFFF	8
ROM Emulation RAM	01000000	-	013FFFFF	

This area specifies the access size, mapping attribute, and address, and sets mapping.

**(a) Selecting memory access size**

The following memory access sizes can be selected.

However, they cannot be selected when ROM Emulation RAM is selected for mapping attribute specification.

- 8Bit: The debugger accesses the memory with the ld.b instruction/st.b instruction.
- 16Bit: The debugger accesses the memory with the ld.h instruction/st.h instruction.
- 32Bit: The debugger accesses the memory with the ld.w instruction/st.w instruction.

Because these settings specify the access size of the debugger software, the hardware of the external bus operates in accordance with the settings of the MODE pin or the IOR register.

**(b) Mapping attribute specification**

The following types of mapping attributes can be selected.

Memory Attribute

Target: Selects a target memory.

The target device accesses memory in the target system or inside the chip.

ROM Emulation RAM: Selects a ROM emulation area using a ROM probe. In this area, high-speed downloading and pseudo software break setting to a ROM area are possible.

I/O Protect: Selects an I/O protect area.

The I/O Protect area can only be set inside the area set in the Target. The area set as I/O Protect cannot be read unless it has been registered as an I/O port in the IOR window, or registered in the Watch window. If it is necessary to read this area, execute a forcible read in these windows.

**(c) Mapping address specification**

Mapping Address: Specify the address to be mapped.

Input the higher address and the lower address from the keyboard.

**(d) Setting of memory mapping**

- **To add memory mapping**

After specifying the Memory Attribute and Mapping Address, click the  button.

Memory mapping corresponding to the Memory Attribute and to the Memory Mapping address range is added.

The mapping units are adjusted when the  button is clicked. If the mapping units do not match, the minimum range subject to mapping that can be set, which includes the specified address, is used.

- **To delete memory mapping**

Select the displayed area where you desire to delete mapping, then click the  button. The currently selected mapping will be deleted.

- **Memory mapping unit**

The mapping units of each area are as follows.

Target area: 1 MB units

ROM Emulation RAM area: 1 MB units

(only one continuous area of up to 4 MB)

I/O Protect area: 1 MB units

**Function buttons**

Validates the current environments. Sets environments and closes the Configuration dialog box.

If an error occurs when the  button is clicked, the debugger stops because it is impossible to continue.

Cancels change and closes this dialog box.

Returns the environmental settings to the state they were in before the Configuration dialog box was opened.

Opens the Project File Load dialog box.

If a project file is open or if an error occurs during reading, the debugger stops because it is impossible to continue.

Opens the help window.

**Caution**

Apply power to the target before clicking  button in the Configuration dialog box.

<b>Extended Option Dialog Box</b>	<b>Modal</b>
-----------------------------------	--------------

### General

This dialog box displays and sets the various extended options of the debugger.

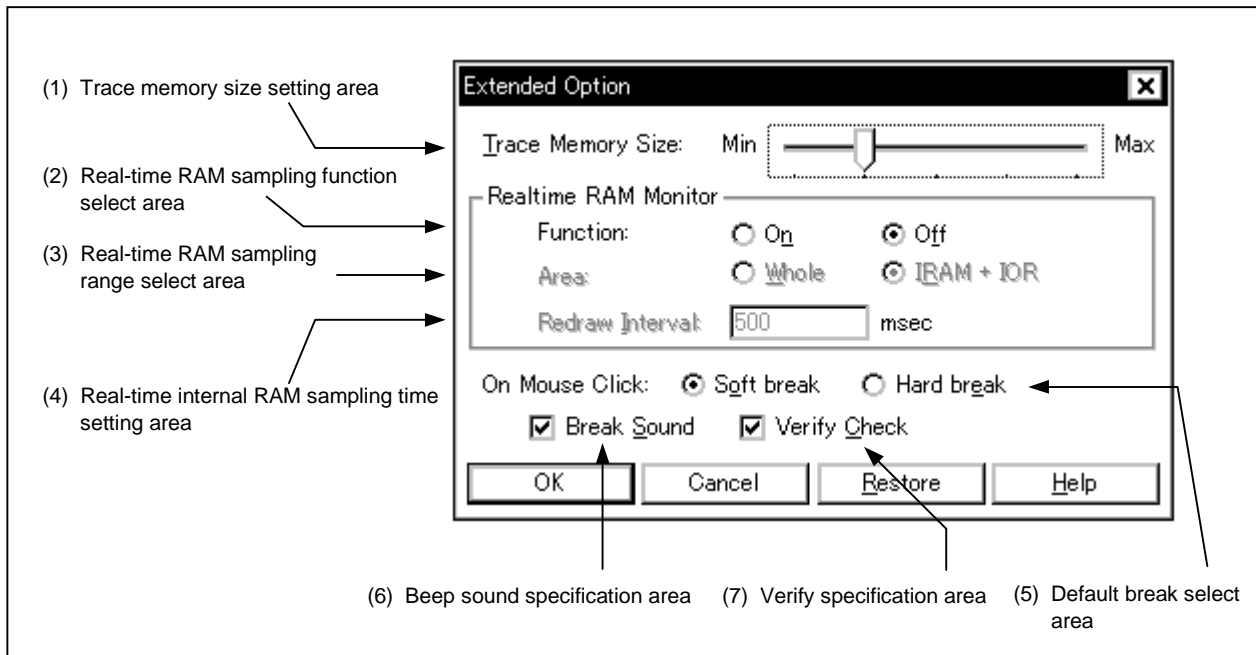
### Opening method

This dialog box can be opened in the following methods.

- In the main window  
Select [Option] → [Extended Option...] from the menu bar.  
Press the [GRPH] + [O], and [E] keys in that order.

### Window

Figure 6-16. Extended Option Dialog Box



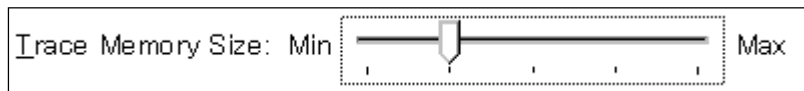
<b>Function</b>
-----------------

The Extended Option dialog box consists of the following areas.

- Trace memory size setting area
- Real-time RAM sampling function select area
- Real-time RAM sampling range select area
- Real-time internal RAM sampling time setting area
- Default break select area
- Beep sound specification area
- Verify specification area

The function of each area is explained below.

### (1) Trace memory size setting area



This area sets the trace memory size to trace user program execution.

This area is valid only for CPU cores with the TRCU incorporated.

The specifiable size is selected from five stages from Min (level 1) to Max (level 5).

The trace memory size in each level is as follows.

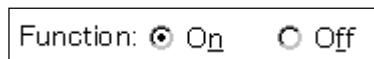
**Table 6-6. Trace Memory Size**

Level	Number of Frames
Level 1 (Min)	4 K
Level 2	32 K (default)
Level 3	64 K
Level 4	90 K
Level 5 (Max)	128 K

Note that these values are for reference and differ from the actual number of trace frames.

In addition, as the memory size gets larger, the number of trace frames to be registered increases, but the response when reading out the trace data becomes worse.

### (2) Real-time RAM sampling function select area



This area makes various settings for the real-time RAM sampling function.

Function:

Selects the real-time RAM sampling function as valid/invalid.

When set as valid, all the software breakpoints and hardware breakpoints already set become invalid.

In addition, because this function is realized by software emulation, the user program execution is temporarily stopped. This is invalid as the default.

**(3) Real-time RAM sampling range select area**

Area:	<input type="radio"/> Whole	<input checked="" type="radio"/> IRAM + IOR
-------	-----------------------------	---

**Area:** Selects the address range in which real-time RAM sampling is to be performed. Variables and data allocated to this range can be displayed in the Watch window or Memory window in close to real time.

**Whole:** The entire memory space is the target.  
The user program execution is stopped for a long time when a large number of windows are opened because the range from which memory is read out is wide. In this case, mnemonics can be displayed in the Trace View window when tracer is stopped during program execution.

**IRAM + IOR:** Only the internal RAM area and IOR area are the target (default).

**(4) Real-time internal RAM sampling time setting area**

Redraw	<input type="text" value="500"/>	msec
--------	----------------------------------	------

This area specifies the sampling time (msec) of the real-time RAM sampling.

Variables, data, etc. allocated to the range specified at Area: are used for the timing to update the data in the Watch window and the Memory window.

**(5) Default break select area**

On Mouse Click:	<input checked="" type="radio"/> Soft break	<input type="radio"/> Hard break
-----------------	---	----------------------------------

Whether to make the breakpoint setting a software break or a hardware break can be selected by clicking the mouse in the point mark area (the most left column) on the Source window or Assemble window.

- **Soft Break:** Sets a software break.  
The breakpoint mark is displayed in blue.
- **Hard Break:** Sets a hardware break.  
The breakpoint mark is displayed in red.

**(6) Beep sound specification area**

<input checked="" type="checkbox"/> Break Sound
---

Specify whether there will be a beep sound when there is a break.

If the check box is checked, a beep sound is emitted when there is a break (the default is with the beep sound activated).



**(7) Verify specification area**

Verify: <input checked="" type="checkbox"/> Check
---

This area specifies whether the memory verification check is to be performed when it is written to. If the check box is checked, when download, memory fill, and memory copy is to be performed the memory is verified. A verify check is also performed when variables or data, etc. have been changed in the Watch window or Memory window, and even when writing to memory (the default is for the check to be performed).

**Function buttons**

OK
----

Makes the currently set environment active.  
Sets the environment and closes the Extended Option dialog box.

Cancel
--------

Cancels the change contents and closes the Extended Option dialog box.

Restore
---------

Returns to the environmental setting state that existed before the Extended Option dialog box was opened.

Help
------

Opens the help window.

<b>Debugger Option Dialog Box</b>	<b>Modal</b>
-----------------------------------	--------------

### General

Displays and sets each type of extended option in the debugger.

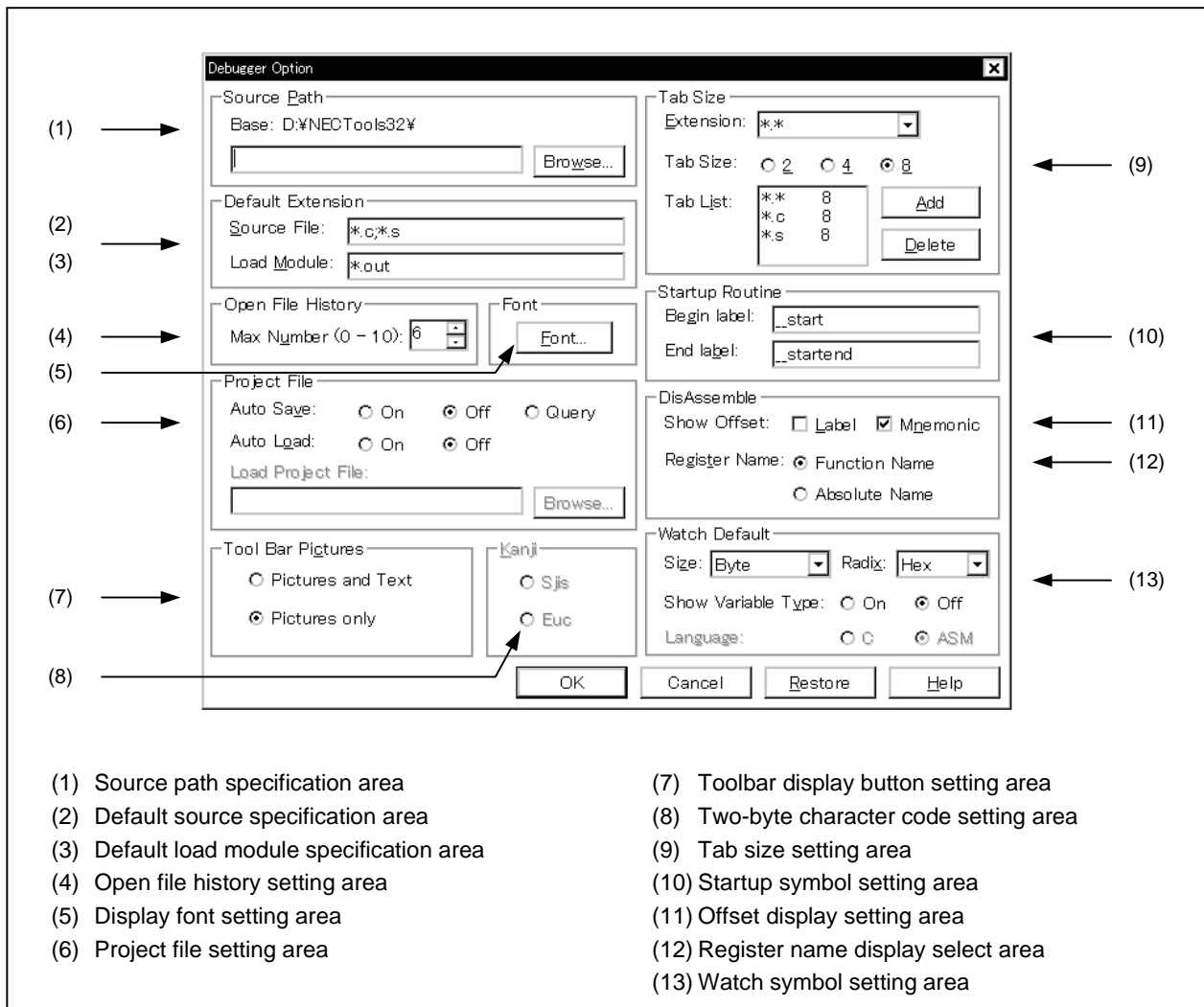
### Opening method

This dialog box can be opened by either of the following methods.

- In the main window  
Select [Option] → [Debugger Option...] from the menu bar.  
Press the [GRPH] + [O], and [D] keys in that order.

### Window

Figure 6-17. Debugger Option Dialog Box



- |  |  |
|--|--|
| (1) Source path specification area         | (7) Toolbar display button setting area  |
| (2) Default source specification area      | (8) Two-byte character code setting area |
| (3) Default load module specification area | (9) Tab size setting area                |
| (4) Open file history setting area         | (10) Startup symbol setting area         |
| (5) Display font setting area              | (11) Offset display setting area         |
| (6) Project file setting area              | (12) Register name display select area   |
|  | (13) Watch symbol setting area           |

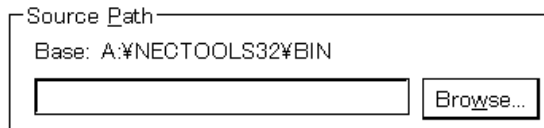
## Functions

The Debugger Option dialog box consists of the following areas.

- Source path specification area
- Default source specification area
- Default load module specification area
- Open file history setting area
- Display font setting area
- Project file setting area
- Toolbar display button setting area
- Two-byte character code setting area
- Tab size setting area
- Startup symbol setting area
- Offset display setting area
- Register name display select area
- Watch symbol setting area

Each function is explained below.

### (1) Source path specification area



Specify the directory in which the source file or text files will be searched for.

The path is specified by inputting it from the keyboard or by using the **Browse...** button.

When the **Browse...** button is clicked, the Browse dialog box is displayed and the source path can be added.

A relative path can also be specified. The directory that is the reference for the relative path is displayed to the right of "Base:". The reference directory is decided using the following procedure.

- The directory from which the project files were loaded.
- The directory from which the latest load module or hex file was loaded.
- Windows current directory.

Delimiters in path information are specified by ";" (semi-colon) or "," (comma).

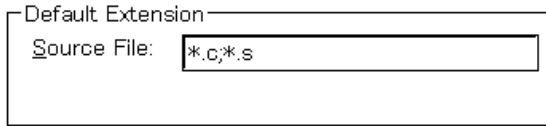
Japanese characters can be specified in the source path (Japanese version only).

**Example** If the source is located in the following directory,

```
b:\src
c:\asm
```

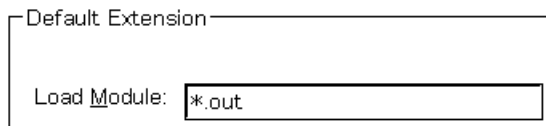
The source path specification becomes as follows.

```
b:\src;c:\asm
```

**(2) Default source specification area**

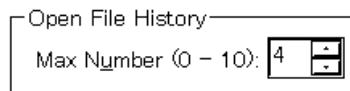
When [File] → [Open...] is selected and the Browse dialog box is opened, the file extension of the displayed source file is set.

File extension delimiters are specified by " " (blank), ";" (semi-colon), or "," (comma).

**(3) Default load module specification area**

Set the file extension of the load module displayed when the Download dialog box is opened.

File extension delimiters are specified by " " (blank), ";" (semi-colon), or "," (comma).

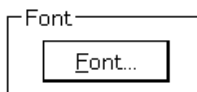
**(4) Open file history setting area**

Set the number of items in the history displayed at the bottom of the [File] menu.

The number of items can be set in a range of 0 to 10.

When 0 is set, the history is not displayed on the menu.

The default is 4.

**(5) Display font setting area**

Specify the display font in the Source window. If the button is clicked, the Font dialog box opens and the display font can be selected.

**(6) Project file setting area**

Project File

Auto Save:  On  Off  Query

Auto Load:  On  Off

Load Project File:

Perform the setting of project file auto save and auto load.

**(a) Project file auto save**

Selection can be made from among the following items.

Auto Save:

- On: Performs project file auto save when closing.
- Off: Does not perform project file auto save when closing.
- Query: Displays the Exit Debugger dialog box when closing (default).

**(b) Project file auto load**

Selection can be made from among the following items.

Auto Load:

- On: Performs project file auto load when starting.
- Off: Does not perform project file auto load when starting (default).

**(c) Auto load project file settings**

Specifies the project file to be loaded during auto load. This can be specified when “Auto Load” has been turned “On.” When it is turned “Off,” the specification column becomes inactive (displayed in gray).

Load Project File: Input the project file name from the keyboard or set it by clicking the  button.

If the  button is clicked, the Browse dialog box is displayed.

**(7) Toolbar display button setting area**

Tool Bar Pictures

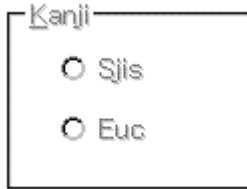
Pictures and Text

Pictures only

Set the toolbar display button.

The display button can be selected from among the following items.

- Pictures and Text: Displays buttons that show both graphics and text.
- Pictures only: Displays buttons that show only graphics (default).

**(8) Two-byte character code setting area**

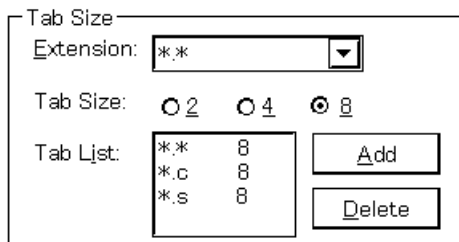
This area specifies the Kanji code (two-byte character code) of the file displayed in the Source or Trace View window<sup>Note</sup>.

Either SJIS or EUC code can be selected. Note that the Kanji code specification is only valid for Japanese Windows.

Sjis: SJIS is used as the Kanji code (default).

Euc: EUC is used as the Kanji code.

**Note** Windows/dialog boxes related to the trace is valid only for CPU cores with the TRCU incorporated.

**(9) Tab size setting area**

Set the tab size for each file extension when displaying files with that file extension.

**(a) File extension setting**

Set the file extension.

Extension: Setting of the file extension can be done by inputting it from the keyboard or by selecting from the drop-down list.

**(b) Tab size selection**

The following tab sizes can be selected.

Tab Size:

- 2: Displays the tab code with 2 spaces.
- 4: Displays the tab code with 4 spaces.
- 8: Displays the tab code with 8 spaces.

**(c) Setting tab size for each file extension**

Tab List: Displays a list of the file extensions and tab sizes.

- If a tab size setting is added, and if a tab size is changed, set it by “Extension” and “Tab Size”, then click the  button.
- To delete a tab size setting,  
Select the file extension setting in the “Tab List,” then click the  button.  
The currently selected file extension setting is deleted.

**(10) Startup symbol setting area**

Specify the beginning and the end symbol of the startup routine.

Be sure to specify this area. If it is left blank, the dialog box cannot be closed.

Downloading an object file in the load module format in the Download dialog box automatically opens the corresponding source file. If the specified symbol is incorrect, the source file will not be able to open up to the PC position in the address range of the corresponding source file. Even the startup routine cannot be skipped using step execution.

- Begin Label: Specify the start symbol.  
The default is \_\_start.
- End Label: Specify the end symbol.  
The default is \_\_startend.

**(11) Offset display setting area**

Set whether there will be an offset display (symbol + offset) during disassemble display.

If there is no offset display, a symbol is displayed only when there is a symbol that matches the numerical value, and when there is not a matching symbol, the numerical value is displayed as is in hex notation.

The areas where the offset display can be specified are of the following two types.

- Label: Set whether or not there will be an offset display in the label column.  
The default is no offset display.
- Mnemonic: Set whether or not there will be offset display in the mnemonic column.  
The default is offset display.

**(12) Register name display select area**

Select the display method of the register name in the mnemonic during disassemble display.

- **Function Name:** Displays the register name as the function name or another name (default).
- **Absolute Name:** Displays the register name as an absolute name.

**(13) Watch symbol setting area**

Specify the watch symbol.

**(a) Default size**

Set the display size of data selected from [View] → [Adaptive] in the Watch window, etc. The size displayed can be selected from among the following items.

Size:

- **Byte:** Displays in 8 bits (default).
- **Half Word:** Displays in 16 bits.
- **Word:** Displays in 32 bits.

**(b) Default radix**

Set the display radix of data selected from [View] → [Proper] in the Watch window, etc. The display radix can be selected from among the following items.

Radix:

- **Hex:** Displayed in hexadecimal (default).
- **Dec:** Displayed in decimal.
- **Oct:** Displayed in octal.
- **Bin:** Displayed in binary.
- **String:** Displayed as a character string.



**(c) Display/non-display of variable type**

Selects the display/non-display of variable type.

Show Variable Type:

- On: Displays the type.
- Off: Not display the type (default).

**(d) Display/non-display of variable type**

Selects the constant display form.

Language:

- C: Displays in C language form (default, fixed).
- ASM: Can't be selected.

**Function buttons**OK

After activating each type of option in the set debugger, the dialog box is closed.

Cancel

Makes each of the options set in the debugger on this occasion inactive and closes the dialog box.

Restore

Returns to the environment setting state existing before the Debugger Option dialog box was opened.

Help

Opens the help window.

<b>Interrupt Mask Dialog Box</b>	<b>Modal</b>
----------------------------------	--------------

### General

Sets the mask for external maskable interrupts (INT0 to INT63).

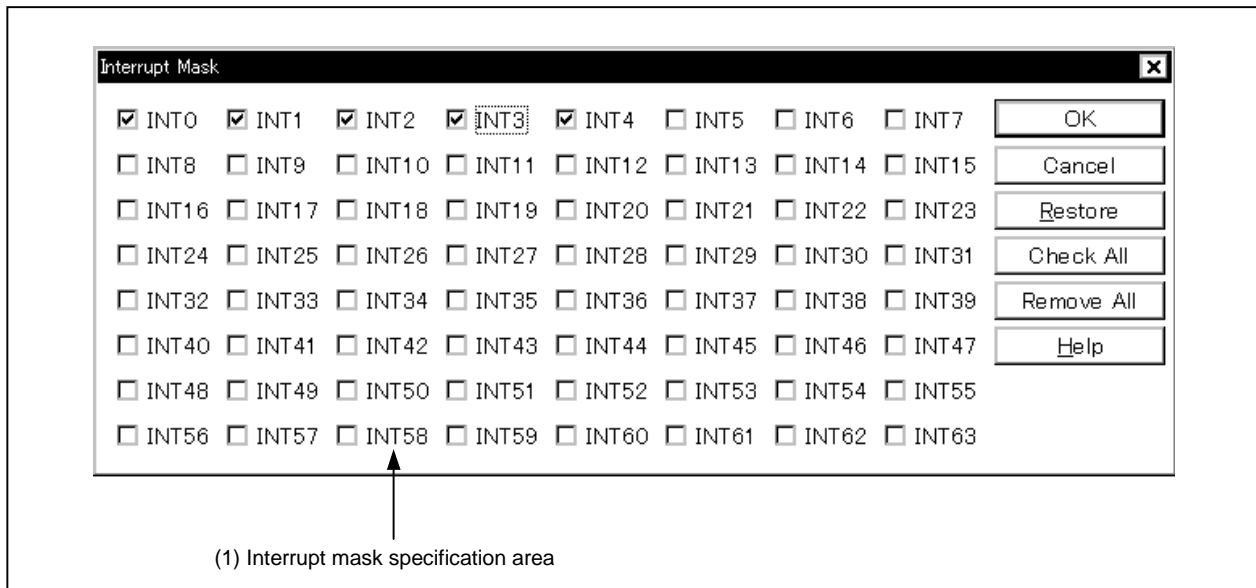
### Opening method

This dialog box can be opened by either of the following methods.

- In the main window  
Select [Option] → [Interrupt Mask...] from the menu bar.  
Press the [GRPH] + [O], and [K] keys in that order.

### Window

**Figure 6-18. Interrupt Mask Dialog Box**



### Function

The Interrupt Mask dialog box consists of the following area.

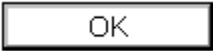
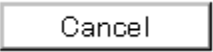

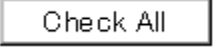
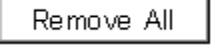
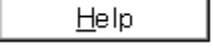
- Interrupt mask specification area

The function of this area is explained below.

#### (1) Interrupt mask specification area

This area specifies whether to mask maskable interrupt signals input from outside the CPU.  
Unchecked interrupts are not masked.

**Function buttons**

	Sets the specified mask condition and closes the dialog box.
	Closes the dialog box without setting data currently specified.
	Returns to the status before the Interrupt Mask dialog box was opened.
	Masks all interrupts.
	Unmasks all interrupts.
	Opens the help window.

<b>Font Dialog Box</b>	<b>Modal</b>
------------------------	--------------

### General

Selects the display font for the Source window and sets the font size.

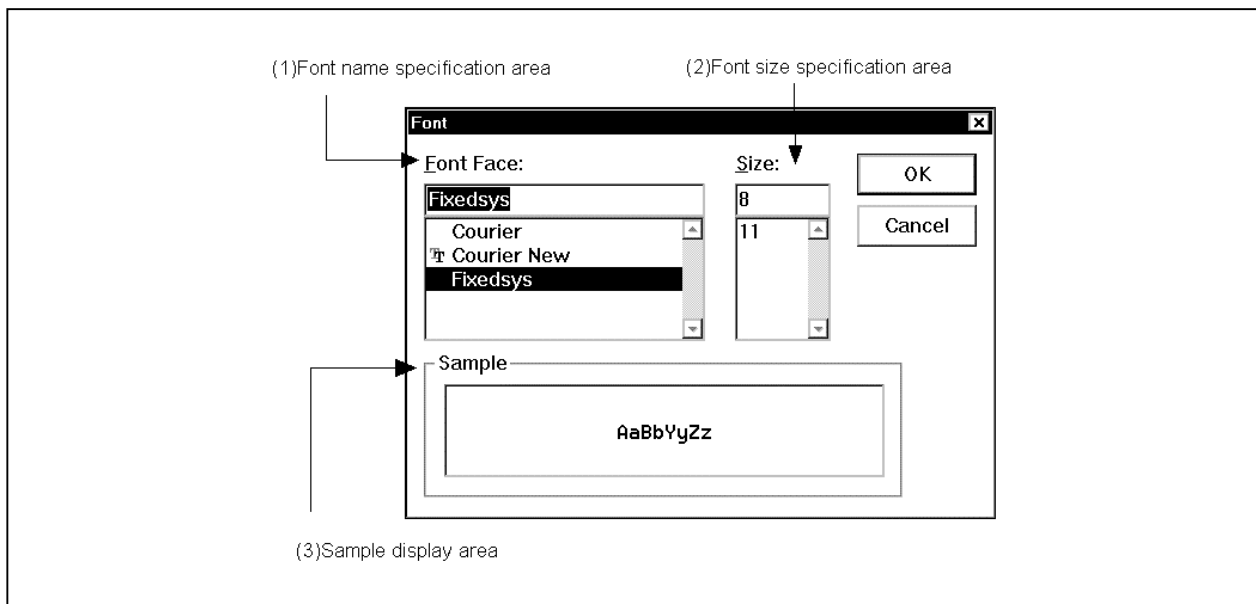
### Opening method

This dialog box can be opened in the following method.

- Click the  button in the Debugger Option dialog box.

### Window

**Figure 6-19. Font Dialog Box**

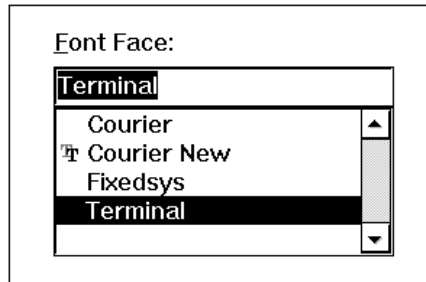


### Functions

The Font dialog box consists of the following areas.

- Font name specification area
- Font size specification area
- Sample display area

Each function is explained below.

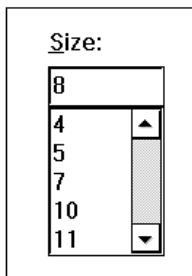
**(1) Font name specification area**

Select the specified font from among the currently usable fonts.

Fixed width fonts (fonts with a constant stroke width and fixed pitch fonts) only are listed.

When the font name is selected from the list, the font name is displayed in the text box and the font sizes with which that font can be used are displayed in the font size specification area.

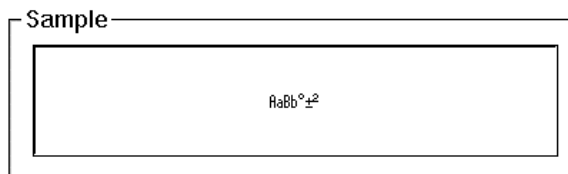
It is also possible to input the font name directly to the text box using the keyboard.

**(2) Font size specification area**

The usable sizes of the font specified in the font name specification area are listed in point units.

When selecting the font size from the list, the size is displayed in the text box.

It is also possible to input the font size directly to the text box from the keyboard.

**(3) Sample display area**

A sample character string showing the specified font and font size is displayed.

**Function buttons**

OK

Changes the font in the Source window to the specified font face and font size and closes the dialog box.

Cancel

Closes the dialog box without changing the font.

<b>Project File Load Dialog Box</b>	<b>Modal</b>
-------------------------------------	--------------


### General

This dialog box restores previous debugging environments.

By loading a project file, the size of the window that displayed the environments and the position of the environments on the window are also restored.

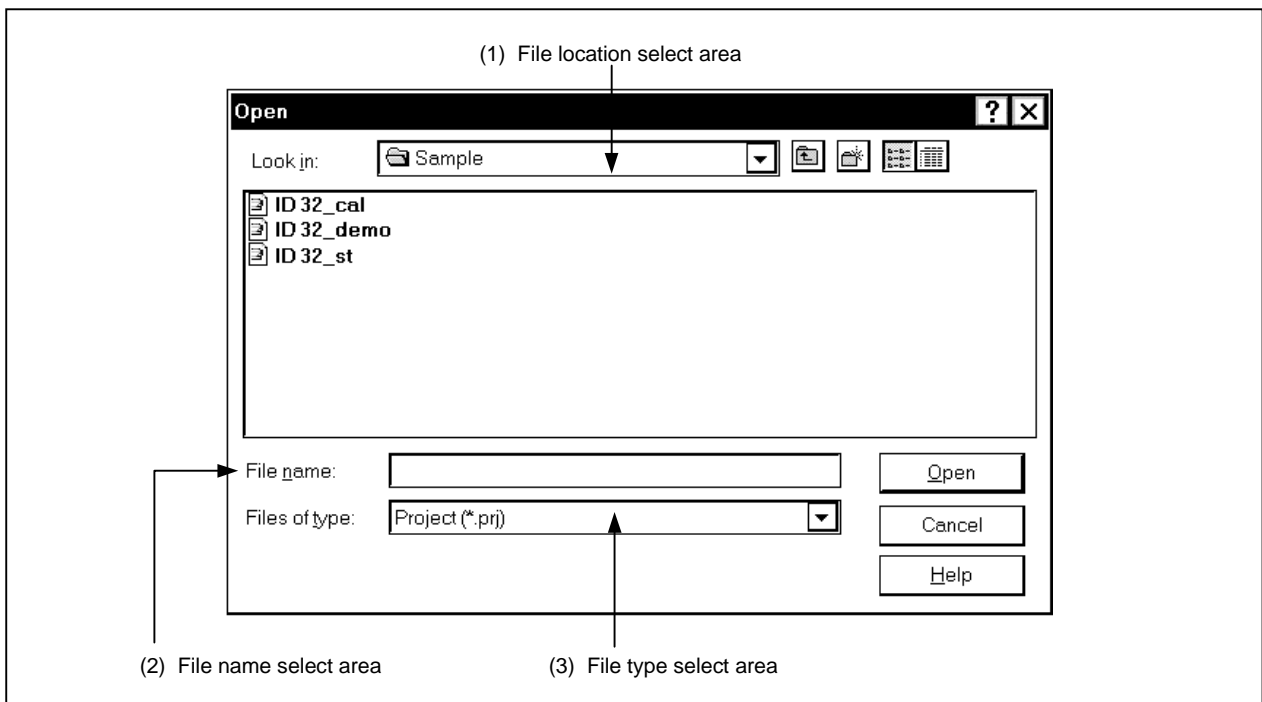
### Opening method

This dialog box can be opened in the following methods.

- In the main window  
Select [File] → [Project...] → [Open Project...] from the menu bar.  
Press the **[GRPH] + [F], [J], and [O]** keys in that order.
- Press the  button on the toolbar.

### Window

**Figure 6-20. Project File Load Dialog Box**



**Function**

Loads project files.

This dialog box consists of the following areas.

- File location select area
- File name select area
- File type select area

The function of each area is explained below.

**(1) File location select area**

Select the drive or directory.

**(2) File name select area**

Specify the file name.

The file name can be input from the keyboard or it can be selected from the list in the top column of the area.

**(3) File type select area**

Displays the type of file displayed in the list. Project (\*.prj) files are displayed.

**Loaded contents**

The items from each window shown in the following table are loaded in the project file. However, if the files are loaded after the debugger is started, when a project file with a different target device is loaded, the debugger itself is reset and the project file is loaded (when the target device differs from the target device specified when the debugger was started, the target device becomes the target device specified in the project file).

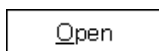


Table 6-7. Saved Contents of Project File

Window	Set Data
Configuration dialog box	All items (target device, clock setting, pin mask setting, mapping information)
Main window	Display position, toolbar/status bar/button display information, execution mode information, trace On/Off information <sup>Note</sup>
Download dialog box	File information to be downloaded
Extended Option dialog box	Setting information
Debugger Option dialog box	Setting information
Interrupt Mask dialog box	Interrupt mask information
Source window	Display information of window, font information
Assemble window	Display information of window, display start address
Memory window	Display information of window, display start address
Stack window	Display information of window
IOR window	Display information of window
Local Variable window	Display information of window
Trace View window <sup>Note</sup>	Display information of window
Event dialog box	Display information of window, event information
Event Link dialog box	Display information of window, link event information
Break dialog box	Display information of window, break event information
Trace dialog box <sup>Note</sup>	Display information of window, trace event information
Event Manager	Display information of window, event information
Software Break Manager	Display information of window, software break information
Register window	Display information of window
Watch window	Display information of window, watch entry information
Trace Option dialog box <sup>Note</sup>	Information of trace option
Delay Count Setting dialog box <sup>Note</sup>	Delay count value

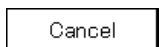
**Note** Valid only for CPU cores with the TRCU incorporated.

#### Function buttons



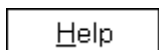
Open

Loads the selected file as the project file.  
Closes the dialog box.



Cancel

Closes this dialog box without loading the project file.



Help

Opens the help window.

<b>Project File Save Dialog Box</b>	<b>Modal</b>
-------------------------------------	--------------

**General**

A project file is a file that stores debugging environments.

To save debugging environments in a project file, the size of the window that displayed the debugging environments and the position of the environments in the window are also saved.

**Opening method**

This dialog box can be opened in either of the following methods.

- In the main window  
Select [File] → [Project] → [Save As...] from the menu bar.  
Press the **[GRPH] + [F], [J], and [A]** keys in that order.

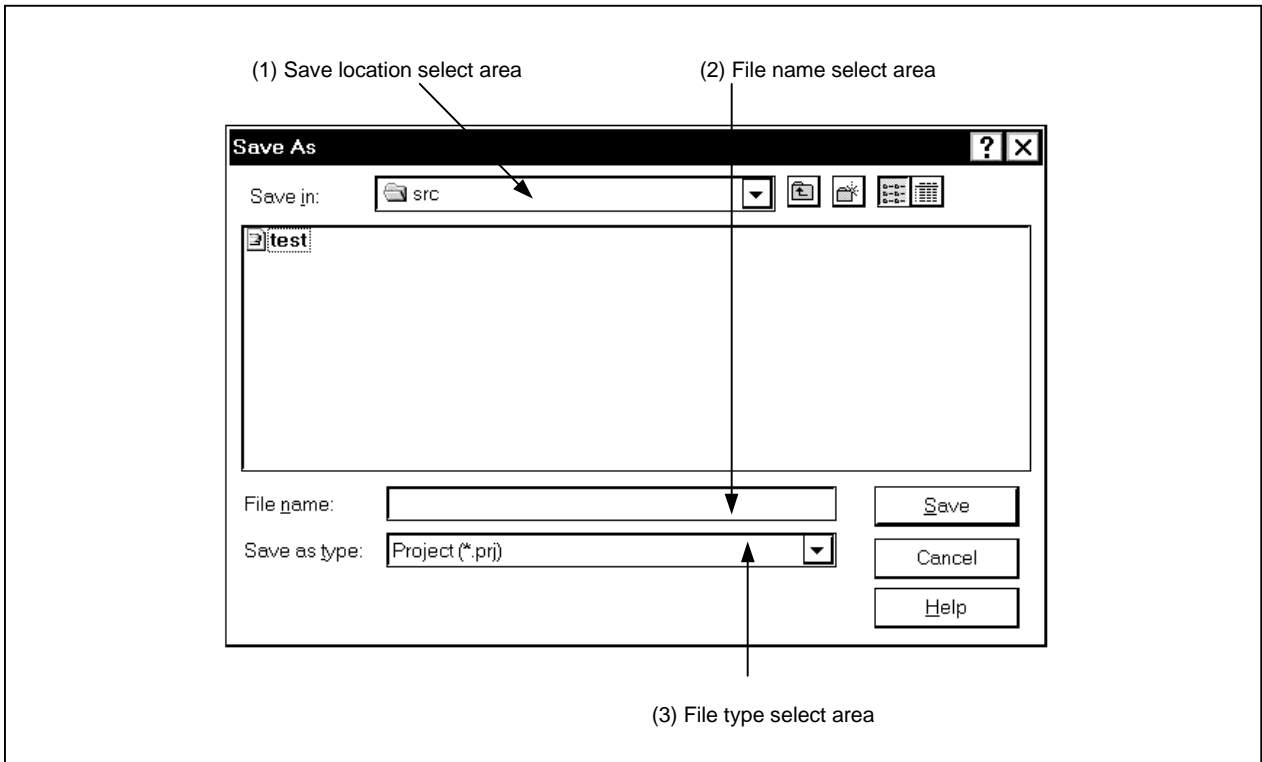
If a project file was loaded or saved previously, and to save a file of the same name, do as follows:

- In the main window  
Select [File] → [Project] → [Save] from the menu bar.  
Press the **[GRPH] + [F], [J], and [S]** keys in that order.

In this way, a file of the same name as the previously loaded or saved project file can be saved without the Project File Save dialog box being opened.

## Window

Figure 6-21. Project File Save Dialog Box



## Function

The Project File Save dialog box is used to save a new project file or a project file whose name has been changed. This dialog box consists of the following areas.

- Save location select area
- File name select area
- File type select area

The function of each area is explained below.

**(1) Save location select area**

Select the drive or directory.

**(2) File name select area**

Specify the file name.

The file name can be input from the keyboard or it can be selected from the list in the top column of the area.

**(3) File type select area**

Displays the type of file displayed in the list. Project (\*.prj) files are displayed.

**Saved contents**

Refer to **Table 6-7 Saved Contents of Project File** for the saved contents of the project file.

**Function buttons**A rectangular button with a thin black border containing the text "Save".

Saves the project information in the selected file.  
After saving, closes the dialog box.

A rectangular button with a thin black border containing the text "Cancel".

Closes this dialog box without saving the project file.

A rectangular button with a thin black border containing the text "Help".

Opens the help window.


<b>View File Load Dialog Box</b>	<b>Modal</b>
----------------------------------	--------------

### General

Reads display files, and also reads source files, other text files and event setting files, and opens a window.

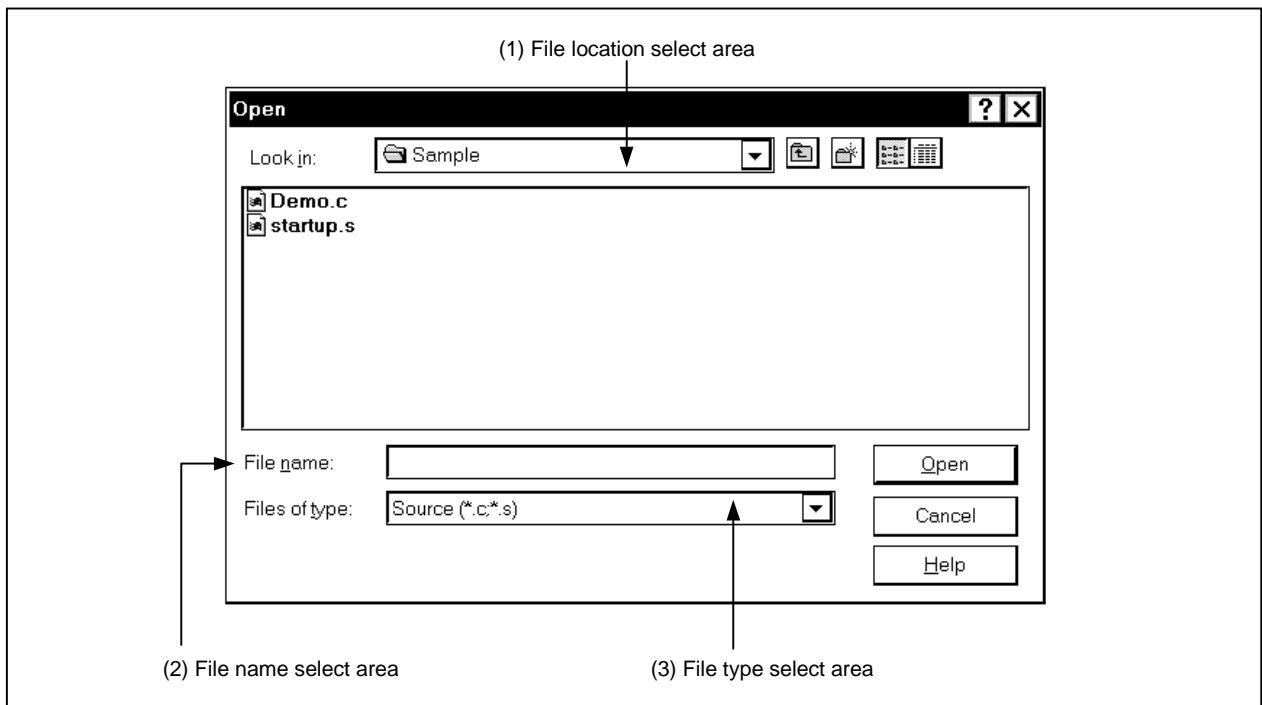
### Opening method

This dialog box can be opened in the following methods.

- In the main window  
 Select [File] → [Open] from the menu bar.  
 Press the **GRPH** + **F**, and **O** keys in that order.  
 Press shortcut keys **CTRL** + **O**.
- Press the  button on the toolbar.

### Window

**Figure 6-22. View File Load Dialog Box**



<b>Function</b>
-----------------

Reads display files saved in View File Save Dialog Box, and also reads source files, other text files and event setting files, and opens a window.

Depending on the file loaded, the window opened and the status differ.

- Loading of a source file (files in which symbol information has been read)  
If a Source window exists in the active state, it is opened in the static state, and there is no Source window in the active state, a Source window in the active state is opened.
- Source file or display file in which symbol information has not been read  
It is opened as a text format file in a Source window in the static state.
- Loading of an event setting file  
The Event Manager is opened and the event setting contents are returned to the previous environment.

This dialog box consists of the following areas.

- File location select area
- File name select area
- File kind select area

The function of each area is explained below.

**(1) File location select area**

Select the drive or directory.

**(2) File name select area**

Specify the file name. The file name can be input from the keyboard or it can be selected from the list in the top column of the area.

**(3) File type select area**

Displays the type of file displayed in the list.

The file types are as shown below.

**Table 6-8. View File Extension (1/2)**

File Kind (File Extension)	Meaning
Source (*.C, *.S) <sup>Note 1</sup>	Source file
Text (*.TXT)	Text file
Source Text (*.SVW)	Source window display file
Assemble (*.DIS)	Assemble window display file
Memory (*.MEM)	Memory window display file
Watch (*.WCH) <sup>Note 2</sup>	Watch window display file

- Notes 1.** The source file extension (C, S) can be changed in the Debugger Option dialog box.
- 2.** If you are using a previous Variable window display file, it can be used as a Watch window display file by changing the file extension from VAR to WCH.

Table 6-8. View File Extension (2/2)

File Type (File Extension)	Meaning
Register (*.RGW)	Register window display file
I/O Register (*.IOR)	IOR window display file
Local Variable (*.LOC)	Local Variable window display file
Stack Trace (*.STK)	Stack window display file
Trace (*.TVW) <sup>Note</sup>	Trace View window display file
Event (*.EVN)	Event setting file
All (*.*)	All files

**Note** Valid only for CPU cores with the TRCU incorporated.

#### Function buttons

**O**pen

Loads the selected file.

**C**ancel

Closes this dialog box without loading the display file.

**H**elp

Opens the help window.

<b>View File Save Dialog Box</b>	<b>Modal</b>
----------------------------------	--------------

### General

Saves the contents of the current window when the dialog box was opened in a display file or event setting file.

### Opening method

This dialog box can be opened in the following methods.

- In the main window, make the window to be saved the current window and perform either of the following operations.
  - Select [File] → [Save As...] from the menu bar.
  - Press the [GRPH] + [F], and [A] keys in that order.

### Window

**Figure 6-23. View File Save Dialog Box (1/3)**

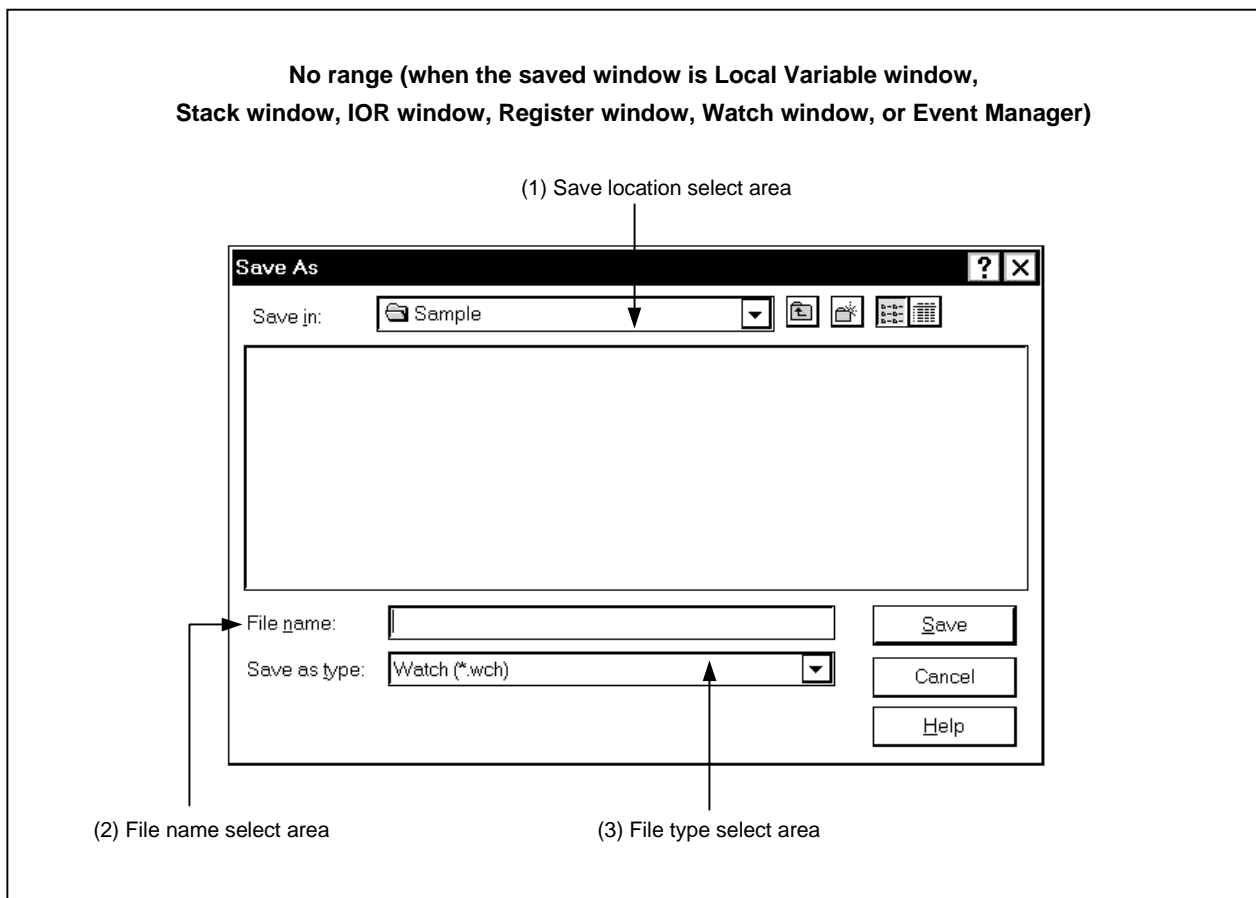
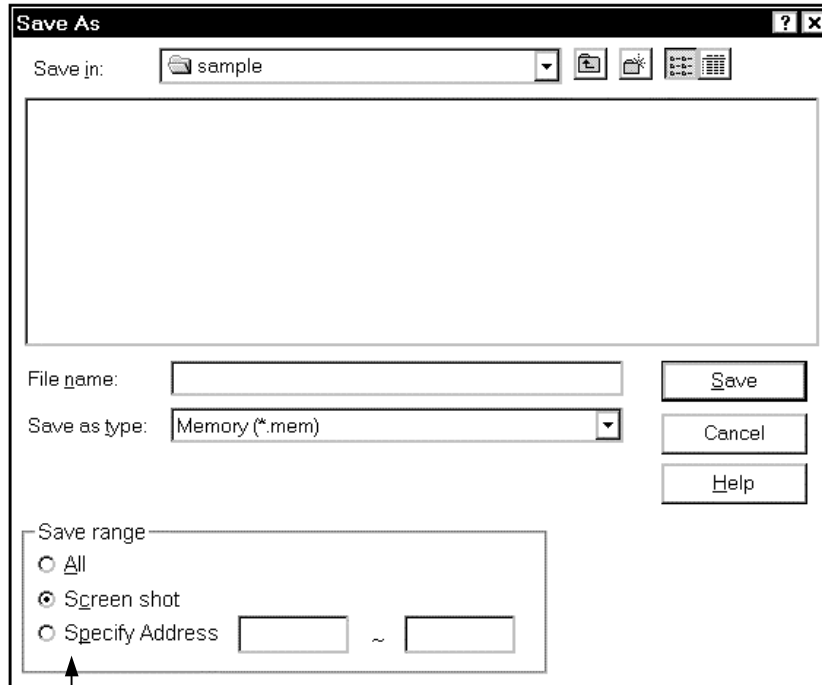




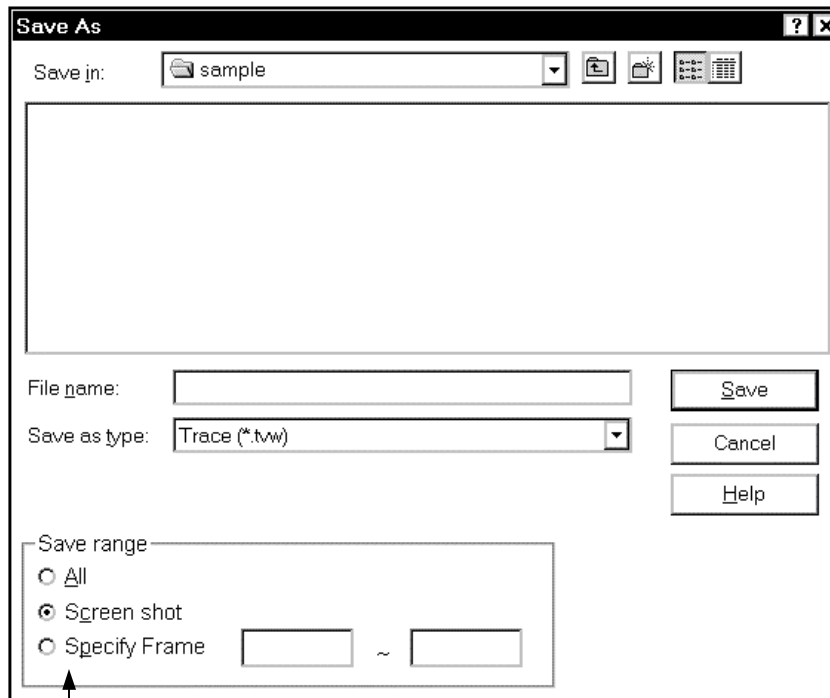
Figure 6-23. View File Save Dialog Box (2/3)

Address range (when the saved window is Assemble window or Memory window)



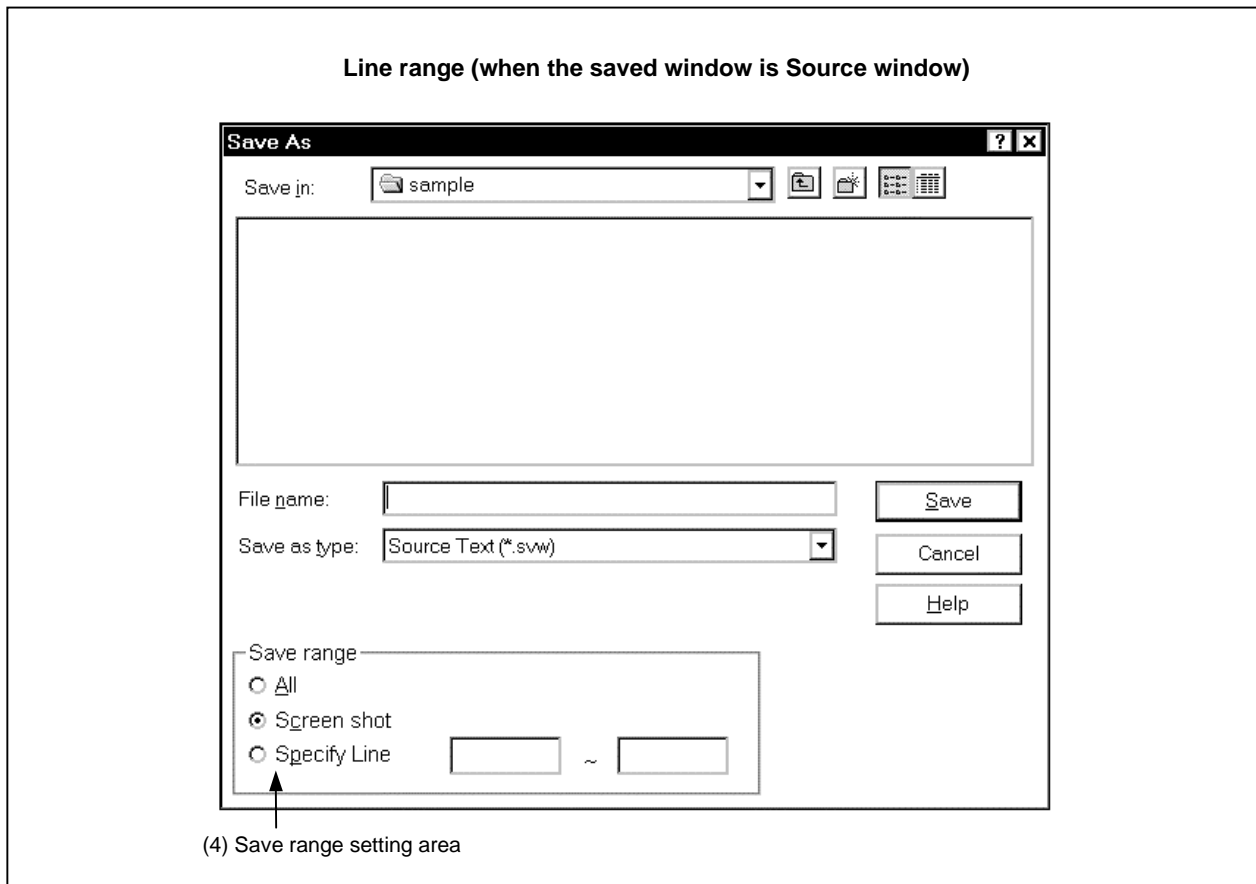
(4) Save range setting area

Frame range (when the saved window is Trace View window)



(4) Save range setting area

Figure 6-23. View File Save Dialog Box (3/3)



### Function

This dialog box saves the contents of the current window in a display file.

This dialog box consists of the following areas.

- Save location select area
- File name select area
- File type select area
- Save range setting area

The function of each area is explained below.

#### (1) Save location select area

Select the drive or directory.

#### (2) File name select area

Specify the file name.

The file name can be input from the keyboard or it can be selected from the list in the top column of the area.

#### (3) File type select area

Displays the type of file displayed in the list. The file types are as shown in the following table.

Table 6-9. File Types and Current Window

File Type (File Extension)	Current Window Name
Source Text (*.SVW)	Source window
Assemble (*.DIS)	Assemble window
Memory (*.MEM)	Memory window
Watch (*.WCH)	Watch window
Register (*.REG)	Register window
I/O Register (*.IOR)	IOR window
Local Variable (*.LOC)	Local Variable window
Stack Trace (*.STK)	Stack window
Trace (*.TVW) <sup>Note</sup>	Trace View window
Event (*.EVN)	Event Manager
All (*.*)	All files

**Note** Valid only for CPU cores with the TRCU incorporated.

#### (4) Save range setting area

This area is displayed only when the current window to be saved is the Assemble window, Memory window, Source window, or Trace View window.

Save range

All  
 Screen shot  
 Specify Address  ~

All: Specifies that all the area from the first line to the last line be saved.

Screen shot: This specifies that the entire visible area from the top line of the screen to the bottom line be saved. However, in cases where there is a mixed display with the Source window, the area saved is the from the source line included in the visible area of the screen.

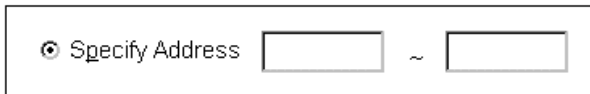
Specify Line (in the case of the Source window),

Specify Frame (in the case of the Trace View window<sup>Note</sup>),

Specify Address (in the case of other windows):

This specifies the starting line and end line of the area to be saved. When the starting line and end line are omitted, it is regarded as if the first line/last line were specified. Details are as shown below.

**Note** Windows/dialog boxes related to the trace are valid only for CPU cores with the TRCU incorporated.

**(a) If current window is Assemble window or Memory window**


Specify Address  ~

Specifies an address range to be saved. Addresses can be also specified by symbols or expressions. (The specification method is the same as in the case of the Address Move dialog box.) The default radix when numbers are input is hexadecimal.

If an area with 256 bytes or more is specified, a dialog box indicating the saving status is displayed. To stop saving midway, click the **Stop** button in this dialog box.



Save

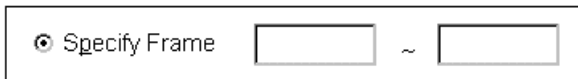
Save Address

start: 0

end: FFFFF

current: 802

Stop

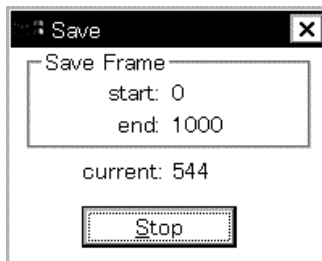
**(b) If current window is Trace View window<sup>Note</sup>**


Specify Frame  ~

Specifies the range of trace frames to be saved. (The specification method is the same as in the case of the Trace Move dialog box.) The default radix when numbers are input is decimal.

If a range of more than 100 frames is specified, a dialog box indicating the saving status is displayed. To stop saving midway, click the **Stop** button in this dialog box.

**Note** Valid only for CPU cores with the TRCU incorporated.



Save

Save Frame

start: 0

end: 1000

current: 544

Stop

**(c) If current window is Source window**

Specify Line  ~

Specify the range by line numbers. The default radix when numbers are input is decimal.

When the Source window is in the mixed display mode, the mixed display portion in the specified lines is also saved.

If a range of more than 100 lines is specified, a dialog box indicating the saving status is displayed. To stop saving midway, click the **Stop** button in this dialog box.

Save

Save Frame

start: 1

end: 2000

current: 85

Stop

**Function buttons**

Save

Saves in the selected file. After that, closes the View File Save dialog box.

Cancel

Closes this dialog box without saving the display file.

Help

Opens the help window.

Restore

(Displayed when the saved window is Assemble window, Memory window, Trace View window, or Source window)

Returns to the status before the View File Save dialog box was opened.

<b>Download Dialog Box</b>	<b>Modal</b>
----------------------------	--------------

**General**

This dialog box selects the name and format of a file to be downloaded, and downloads the contents of memory to the in-circuit emulator or target.

If an object file in the load module format has been loaded, the corresponding source file is searched, and the Source window automatically opens.

Multi-loadable load module files are up to 20.


The files and formats of them that can be downloaded are as follows.

- NEC load module format (ELF (.OUT))
- Intel HEX format file (normal or extended)
- Motorola HEX format file S type (S0, S2, S3, S7 and S8)
- Extended Tektronix HEX format file
- Binary data format file

If a file other than an object file in the load module format is loaded, debugging cannot be executed at the source level.

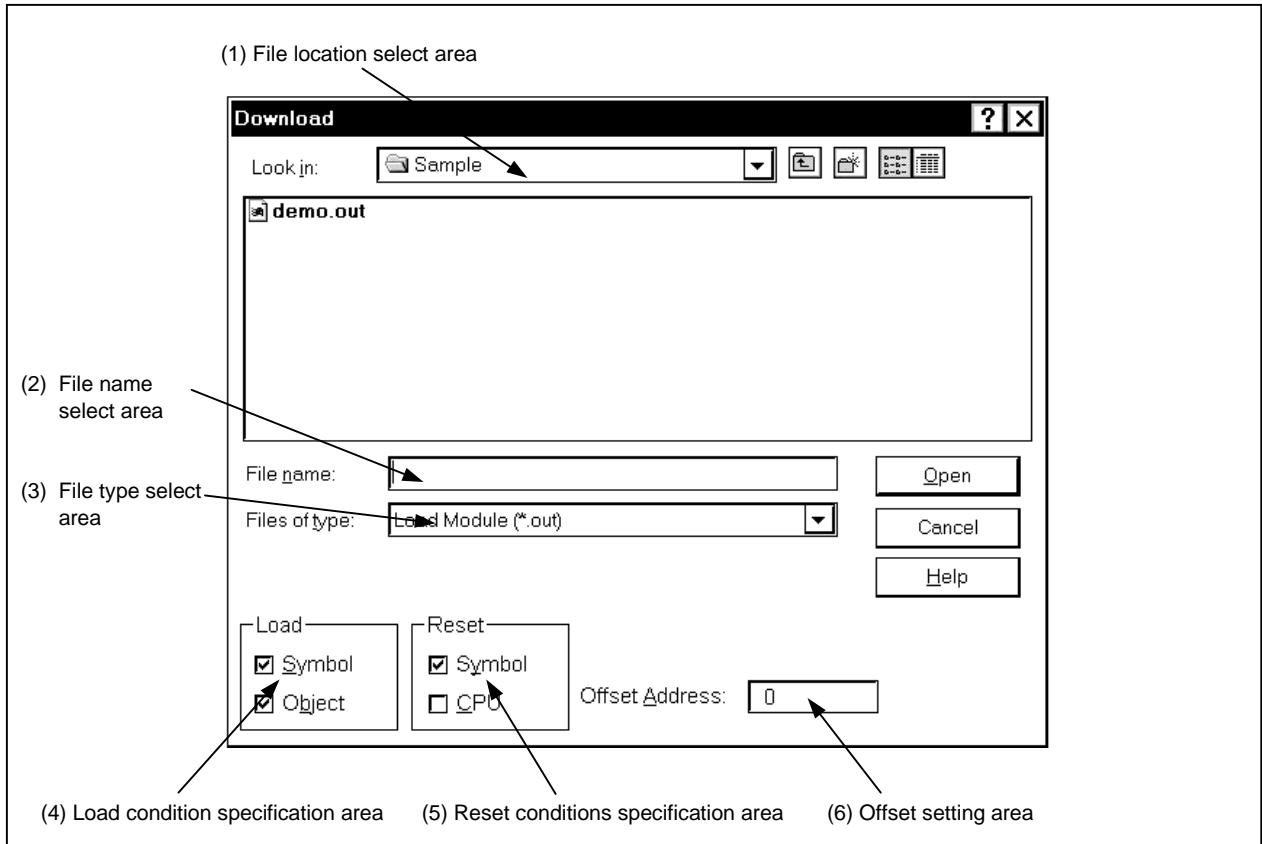
**Opening method**

This dialog box can be opened in the following methods.

- In the main window  
Select [File] → [Download...] from the menu bar.  
Press the **[GRPH]** + **[F]**, and **[D]** keys in that order.
- Press the  button on the toolbar.

Window

Figure 6-24. Download Dialog Box



Function

The maximum number of load module files that can be loaded is 20.

The Download dialog box consists of the following areas.

- File location select area
- File name select area
- File type select area
- Load condition specification area
- Reset conditions specification area
- Offset setting area

The function of each area is explained below.

**(1) File location select area**

Select the drive or directory.

**(2) File name select area**

Specify the file name. The file name can be input from the keyboard or it can be selected from the list in the top column of the area. This dialog box can be used to specify multiple files.

If you are specifying multiple file names, place “;” (semi-colon) on both sides of each file name. It is also possible to specify multiple file names from the top column of the area while pressing the **SHIFT** key or the **CTRL** key.

**(3) File type select area**

Displays the file type to be in the list.

When downloading, specify the same extension of the file specified in (2).

**Example** Specify “Load Module (\*.OUT)” when downloading a load module file, and “Hex Format (\*.HEX)” when downloading a hex file.

The contents displayed for “Files of type” are as shown below.

**Table 6-10. Types of Files That Can Be Downloaded**

Contents (Extension)	Format
Load Module (*.OUT)	Load module format
Hex Format (*.HEX)	Hex format (the format is auto-judged)
Binary Data (*.BIN)	Binary data format

**Remark** The Hex format is judged automatically.

Note that these are the default file extensions; file extensions other than these can also be used.

**(4) Load condition specification area**

Load

Symbol

Object

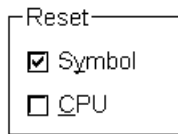
Sets the load conditions.

**Symbol:** Specifies whether the symbol information is read (checked, default) or not. Note that this condition is disregarded for files other than those in the load module format.

**Object:** Specifies whether the object information is read (checked, default) or not.

In cases where a program configured from multiple load module files is debugged, if there are modules where symbol information is not necessary, if the debugger is set so as not to read symbol information, the memory use volume can be reduced.



**(5) Reset conditions specification area**

 A dialog box titled "Reset" containing two checked checkboxes: "Symbol" and "CPU".

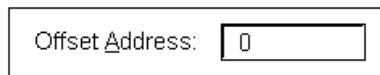
Sets the reset conditions.

Symbol: Specifies whether to reset (checked, default) or not reset symbol information.

CPU: Specifies whether to reset or not reset (checked, default) the CPU.

When debugging a program configured from multiple load module files, load individual load module files without resetting the symbol information. When downloading the multiple load module files, do not overlap the location addresses.

This setting is disregarded for files other than load module format and hex format files.

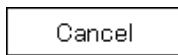
**(6) Offset setting area**

 A dialog box titled "Offset Address:" with a text input field containing the value "0".

Specify the offset address when loading. Address specification can also be by expressions (except symbols). The specification method is the same as in the case of the Address Move dialog box. The default radix when inputting numerical values is hexadecimal.

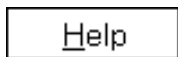
**Function buttons**

 A button labeled "Open".

After the selected file is loaded, the dialog box is closed.


 A button labeled "Cancel".

Closes this dialog box without loading the file.


 A button labeled "Help".

Opens the help window.


 A button labeled "Restore".

Returns the input data to its original form.

<b>Load Module List Dialog Box</b>	<b>Modal</b>
------------------------------------	--------------

### General

The name and file path name of load module files that have been downloaded can be verified. Also, the Download dialog box can be displayed and the load module can be downloaded.

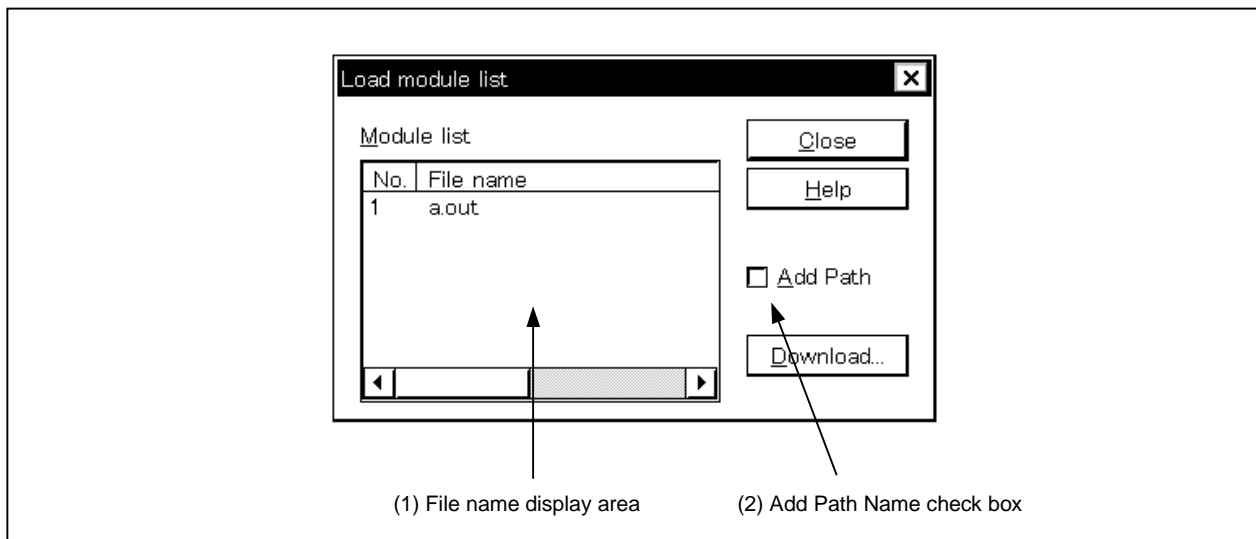
### Opening method

This dialog box can be opened by either of the following methods.

- In the main window  
Select [File] → [Load Module...] from the menu bar.  
Press the [GRPH] + [F], and [L] keys in that order.

### Window

**Figure 6-25. Load Module List Dialog Box**



### Function

The Load Module List dialog box consists of the following areas.

- File name display area
- Add Path Name check box

Each function is explained below.

**(1) File name display area**

Module list

No.	File name
1	a.out
2	f1.out
3	p2tp.out

This area displays the names of load module files that have been downloaded.

The numbers displayed in the “No.” column show the order in which the files were read. When the Add Path Name check box is checked, the full path is displayed, and when it is not checked, the file name only is displayed.

When the file name does not fit on the screen, the mouse or the arrow keys on the keyboard can be used to scroll in order to verify the full name.

If the symbol information has been reset in the Reset Debugger dialog box, or when the symbol information has been reset and downloaded in the Download dialog box, the load module file name with which it was downloaded previously is cleared.

**(2) Add Path Name check box**
 Add Path

This enables you to switch the display of file names displayed in the file name display area between display with the path (checked) or display with the file name only (unchecked, default).

**Function buttons**


Closes the dialog box.

Opens the help window.

Displays the Download dialog box.

The load module can be newly downloaded.

The file name for the newly downloaded file is added in the file name display area when the Download dialog box is closed.

<b>Upload Dialog Box</b>	<b>Modal</b>
--------------------------	--------------

### General

This dialog box selects the name and format of the file to be saved, and saves the contents of the target memory in this file.

The following files and formats of them to be saved can be selected.

- Intel HEX format file (extended only)
- Motorola HEX format file S type (S0, S3, and S7 only)
- Extended Tektronix HEX format file
- Binary data format file

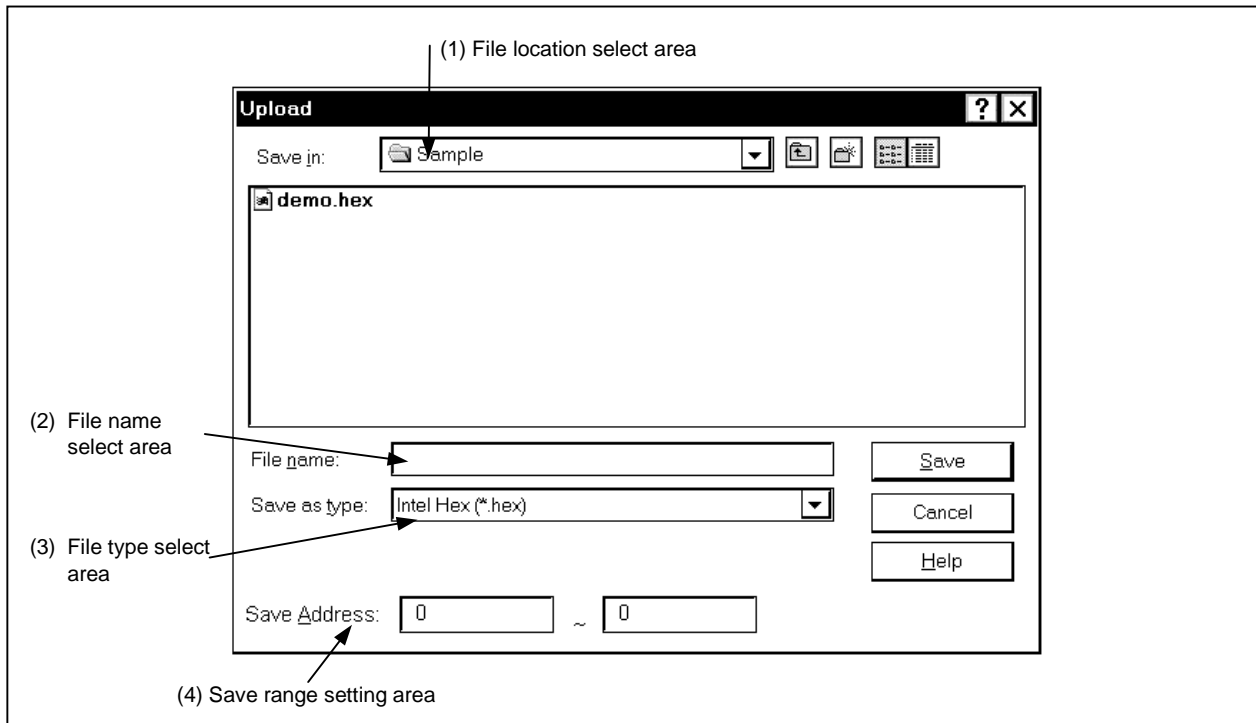
### Opening method

The dialog box can be opened in the following methods.

- In the main window  
Select [File] → [Upload...] from the menu bar.  
Press the [GRPH] + [F], and [U] keys in that order.

### Window

**Figure 6-26. Upload Dialog Box**



<b>Function</b>
-----------------

The Upload dialog box consists of the following areas.

- File location select area
- File name select area
- File type select area
- Save range setting area

The function of each area is explained below.

**(1) File location select area**

Select the drive or directory.

**(2) File name select area**

Specify the file name. The file name can be input from the keyboard or it can be selected from the list in the top column of the area.

**(3) File type select area**

The contents displayed for "Save as type" are as shown below.

**Table 6-11. Types of Files That Can Be Uploaded**

Contents (File Extension)	Format
Intel Hex (*.HEX)	Intel extended hex format
Motorola Hex (*.HEX)	Motorola hex format
Tektro Hex (*.HEX)	Extended Tektronics hex format
Binary Data (*.BIN)	Binary data format
All (*.*)	All files

These are the default file extensions; file extensions other than these can also be used.

The data format is decided by the file extension in which it is saved. However, if "All (\*.\*)" is selected, the file is saved in the default Intel extended hex format.

**(4) Save range setting area**

Save Address:	<input type="text"/>	~	<input type="text"/>
---------------	----------------------	---	----------------------

Specify the address range that is to be saved. It is possible to specify the address by symbols or by expressions. The specification method is the same as in the case of the Address Move dialog box. The default radix when numbers are input is hexadecimal.

**Function buttons** Save

After saving the specified address range in the selected file, the dialog box is closed. The font in the Source window is changed to the specified font face and font size and the dialog box is closed.

 Cancel

Closes this dialog box without saving the file.

 Help

Opens the help window.

 Restore

Returns the input data to its original form.

<b>Browse Dialog Box</b>	<b>Modal</b>
--------------------------	--------------

### General

Selects the file set in the Debugger Option dialog box or Source Search dialog box.

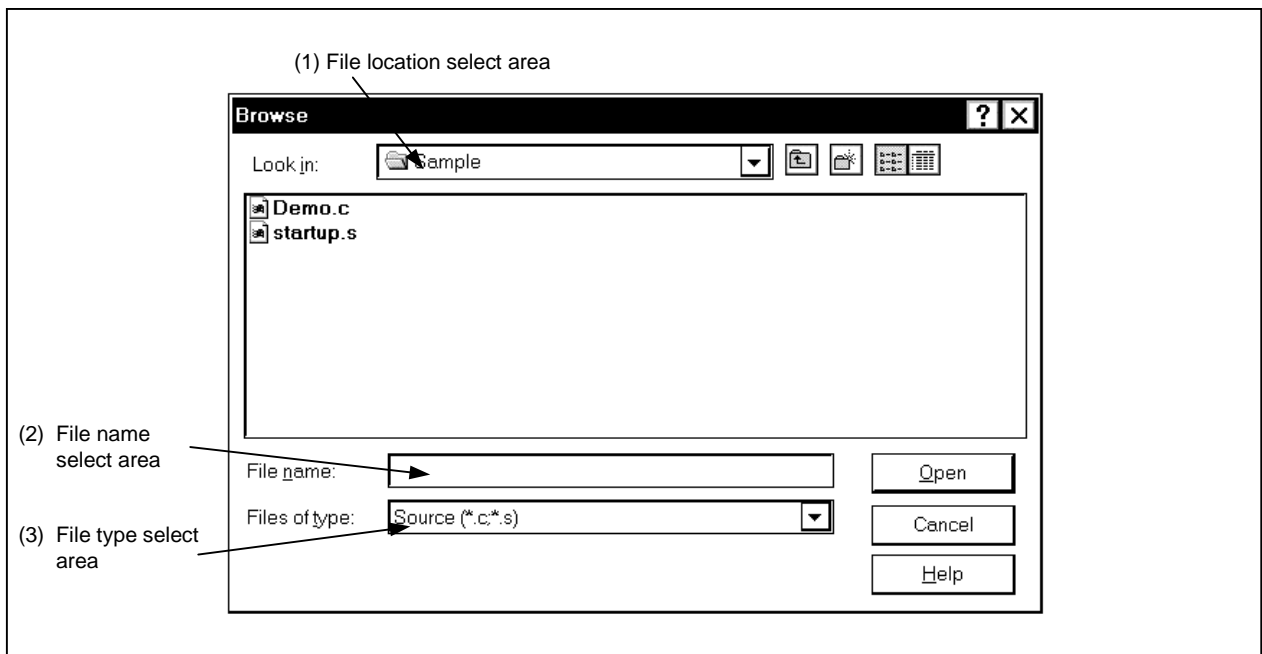
### Opening method

This dialog box can be opened in either of the following methods.

- Click the **Browse...** button in the Debugger Option dialog box.
- Click the **Browse...** button in the Source Search dialog box.
- Click the **Browse...** button in the Source Move dialog box.

### Window

**Figure 6-27. Browse Dialog Box**



<b>Function</b>
-----------------

The Browse dialog box consists of the following areas.

- File location select area
- File name select area
- File type select area

The function of each area is explained below.

**(1) File location select area**

Select the drive or directory.

**(2) File name select area**

Specify the file name. The file name can be input from the keyboard or it can be selected from the list in the top column of the area. Up to 257 characters can be specified.

**(3) File type select area**

Displays the types of files in the list.

The contents displayed in the “Files of type” are as shown below.

**Table 6-12. Types of Files That Can Be Displayed in Browse Dialog Box**

Contents (File Extension)	Format
Source (*.C;, *.S)	Source file (The file extension is specified in the Debugger Option dialog box)
Text (*.TXT)	Text file
All (*.*)	All files

**Remark** The source (c, s) file extension can be changed in the Debugger Option dialog box.

<b>Function buttons</b>
-------------------------

Open
------

Sets the selected file and closes the dialog box.

Cancel
--------

Closes this dialog box without setting the file.

Help
------

Opens the help window.



<b>Source Move Dialog Box</b>	<b>Modal</b>
-------------------------------	--------------

### General

Specifies the file displayed in the Source window and specifies the display start position.

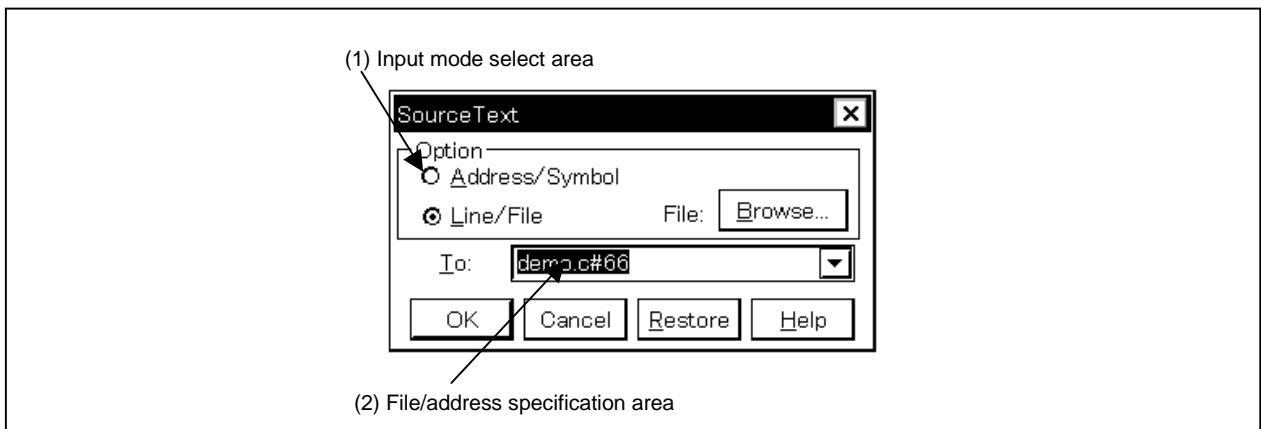
### Opening method

This dialog box can be opened in the following methods.

- When the Source window is active  
Select [View] → [Move...] from the menu bar.  
Press the [GRPH] + [V], and [V] keys in that order.  
Press shortcut keys [CTRL] + [J].

### Window

**Figure 6-28. Source Move Dialog Box**



### Function

This dialog box is displayed when [View] → [Move...] is selected while the Source window is opening and the file's display start position can be changed (a new window does not open).

The Source Move dialog box consists of the following areas.

- Input mode select area
- File/address specification area

The function of each area is explained below.

**(1) Input mode select area**

This selects the input mode in cases where the movement destination is specified.


- Address/Symbol: Specifies the movement destination by an address (or a symbol).
- Line/File: Specifies the movement destination by a line No. (or file name).

**(2) File/address specification area**

This area specifies the display file name or display address. With the default, the string selected in the accessed window, or when there is no selected string, the first character string in the input history, is displayed, but as necessary, it can be changed by inputting from the keyboard.

The following two specification methods are used.

- Display address specification
- Display file name specification

By clicking the  button and displaying the input history, the contents input previously can be reused. Up to 16 items of input history can be retained.

**(a) Display address specification**

Specifies the Address/Symbol in (1).

This specifies the address where the display starts. It is possible to specify the address by symbols or by expressions.

The specification method is the same as in the Address Move dialog box. When “Address/Symbol” is selected, the default radix when inputting numerical values becomes hexadecimal.

The source text is displayed so that source line corresponding to the specified address value can be seen.

**(b) Display file name specification**

Specifies the Line/File in (1).

This specifies the file name which is displayed in the Source window.

For the file, it is possible to specify the file name only and to specify the absolute path or the relative path.

If the file name only, or the relative path, is specified, search is made for the file in each of the following directories in the order shown. The first file to be found as a result of the search is displayed.

- The directory which is the source path specified in the Debugger Option dialog box.

The line number can also be specified other than the file name. The file's line number is specified as follows.

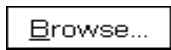
- <<Path Name>File Name #> Line No.

The file name and line number are specified with the “#” used to divide them. The line number is specified by a decimal number. The file name can be omitted. When “Line/File” is selected, the default radix when inputting numerical values is decimal.

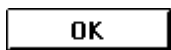
The specified line number is made the header line, and the specified file is displayed. If the file name is omitted, the file is displayed from the specified line of the currently displayed file.

When the line number is omitted, the file is displayed from the header line.

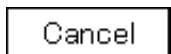
#### Function buttons

A rectangular button with a thin black border containing the text "Browse...".

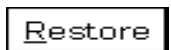
Opens the Browse dialog box.

A rectangular button with a thin black border containing the text "OK".

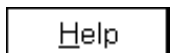
Displays the source text from the specified position.

A rectangular button with a thin black border containing the text "Cancel".

Closes this dialog box.

A rectangular button with a thin black border containing the text "Restore".

Returns the input data to original value.

A rectangular button with a thin black border containing the text "Help".

Opens the help window.

<b>Address Move Dialog Box</b>	<b>Modal</b>
--------------------------------	--------------

### General

Specifies a display start address in the Memory window or Assemble window.

### Opening method

This dialog box can be opened in the following methods.

#### (1) In Assemble window

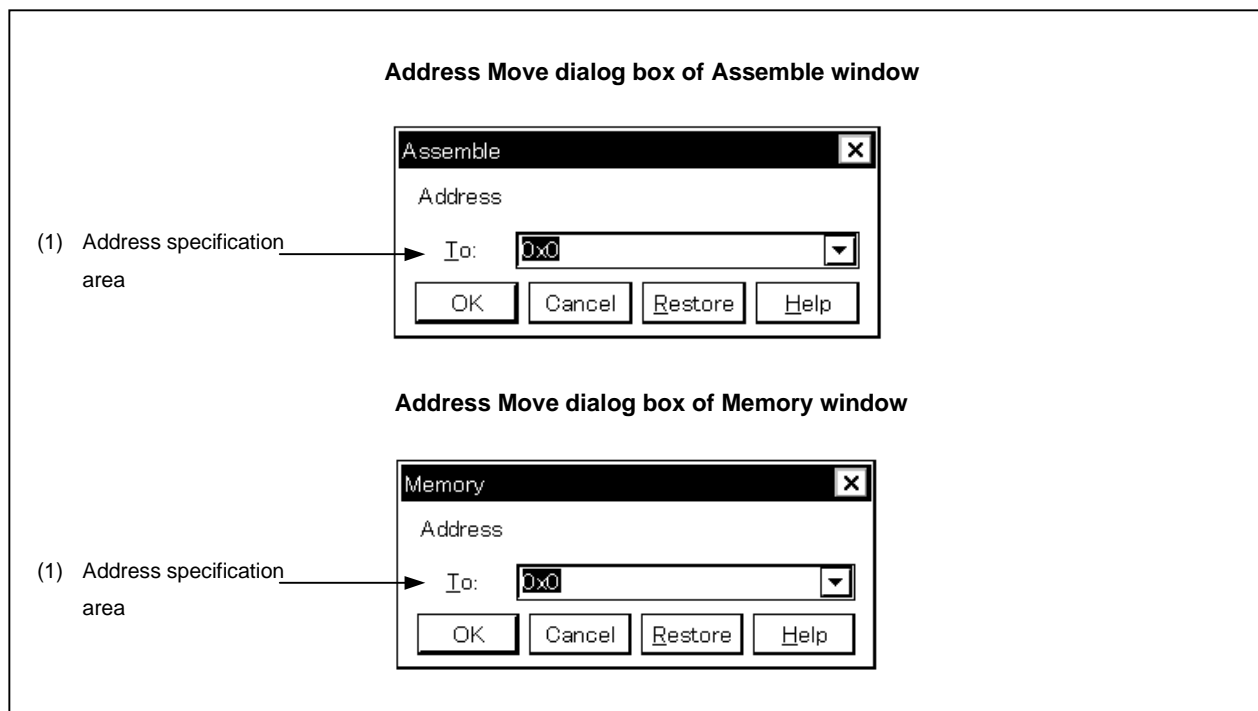
- In the main window  
Select [View] → [Move...] from the menu bar.  
Press the [GRPH] + [V], and [V] keys in that order.  
Press shortcut keys [CTRL] + [J].

#### (2) In Memory window

- In the main window  
Select [View] → [Move...] from the menu bar.  
Press the [GRPH] + [V], and [V] keys in that order.  
Press shortcut keys [CTRL] + [J].

### Window

Figure 6-29. Address Move Dialog Box

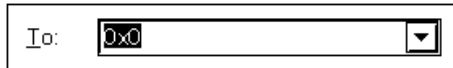


**Function**

The Address Move dialog box consists of the following area.


- Address specification area

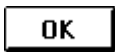
The function of this area is explained below.

**(1) Address specification area**

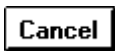
This area specifies a display start address. The default address is the string selected in the called window, or the value of the current PC. This value can be changed by keyboard input as necessary. An address can also be specified by a symbol or expression.

The specification method is the same as in the case of the Symbol To Address dialog box, but the default radix when inputting numerical values is hexadecimal.

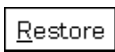
By clicking the  button and displaying the input history, the contents input previously can be reused. Up to 16 items of input history can be retained.

**Function buttons**

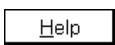
Displays the Memory window or Assemble window from the specified address.



Closes this dialog box.



Returns the input data to original value.



Opens the help window.

<b>Trace Move Dialog Box</b>	<b>Modal</b>
------------------------------	--------------

### General

Specifies the Trace View window display's display start position.  
This dialog box is valid only for CPU cores with the TRCU incorporated.

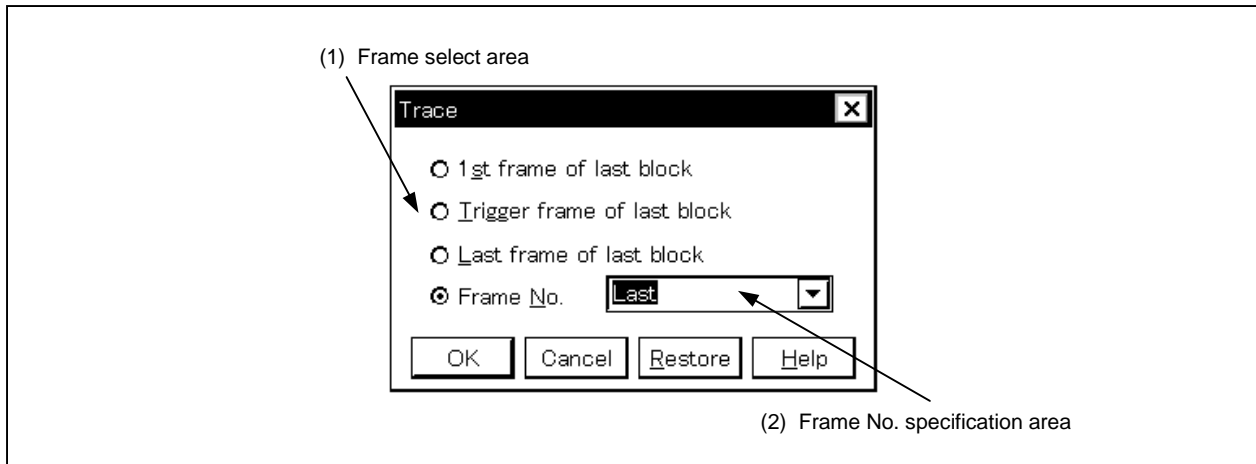
### Opening Method

This dialog box can be opened in the following methods.

- When the Trace View window is active,  
Select [View] → [Move...] from the menu bar.  
Press the [GRPH] + [V], and [V] keys in that order.  
Press shortcut keys [CTRL] + [J].

### Window

**Figure 6-30. Trace Move Dialog Box**



### Function

This dialog box is displayed by selecting [View] → [Move...] when the Trace View window is opening, and it can be used to change the trace results display starting position.

The Trace Move dialog box consists of the following areas.

- Frame select area
- Frame No. specification area

The function of each area is explained below.

**(1) Frame select area**

<input type="radio"/>	1 <sup>st</sup> frame of last block
<input type="radio"/>	Trigger frame of last block
<input type="radio"/>	Last frame of last block
<input checked="" type="radio"/>	Frame No. <input type="text" value="Last"/>

This selects the frame which is the move destination.

1st frame of last block: Move to the first frame in the trace data.

Trigger frame of last block: Move to the latest trigger frame in the trace data.

Last frame of last block: Move to the last frame in the trace data.

Frame No.: Move to the frame with the specified No.

When 0 is specified, move to the first frame in the trace data.

When this item is checked, shift the focus to the area specified by the frame No.

**(2) Frame No. specification area**


If “Frame No.” is selected in the frame select area, this is the area which specifies the frame No. In the default, the string selected in the accessed window, or “Last” is displayed, but as necessary, it can be changed by inputting from the keyboard.

The default radix when inputting numerical values is decimal.

When a “+” mark is input in front of a numerical value, the area moves forward the specified number of frames (below the display) from the frame at the cursor position. Conversely, when a “-” mark is input in front of the numerical value, the area moves the specified number of frames backward (above the display).

The frame No. specification can also be specified in the following format. It is also possible to specify it in abbreviated form. Input characters are not case sensitive.


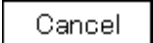


The contents are shown below.

**Table 6-13. Format of Frame No. Specification**

Specification	Abbreviated Form	Content
First	S	Move to the header frame of the trace data.
Trigger	T	Move to the trigger frame of the trace data.
Last	L	Move to the last frame of the trace data.

By clicking the  button and displaying the input history, the contents input previously can be reused. Up to 16 items of input history can be retained.

**Function buttons**

	Performs trace display from the specified position.
	Closes the Trace Move dialog box.
	Returns the input data to the original form.
	Opens the help window.



<b>Symbol To Address Dialog Box</b>	<b>Modeless</b>
-------------------------------------	-----------------

### General

This dialog box displays the address of a specified variable or function, and the value of a symbol.

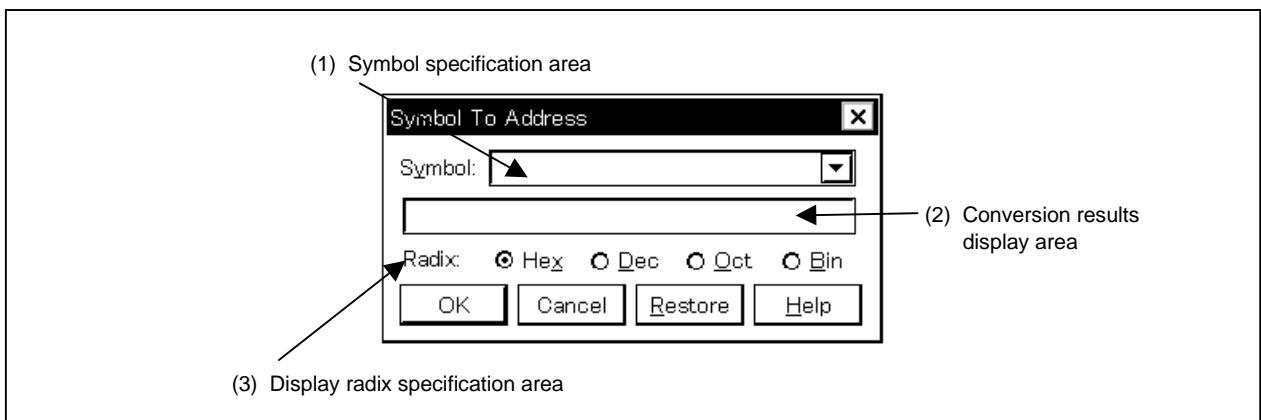
### Opening method

This dialog box can be opened in the following methods.

- Select [V]iew → [S]ymbol... from the menu bar.
- Press the [GRPH] + [V], and [Y] keys in that order.

### Window

**Figure 6-31. Symbol To Address Dialog Box**



### Function

The Symbol To Address dialog box consists of the following areas.

- Symbol specification area
- Conversion results display area
- Display radix specification area

The function of each area is explained below.

#### (1) Symbol specification area

Symbol:

This area specifies the variable, function name, symbol name, or line number whose address is to be converted. Other than this, an I/O port name, peripheral I/O register name, register name, PSW flag name or an expression can be specified.

The I/O port name and peripheral I/O register name are displayed by address values, the register name is displayed by the register contents and the PSW flag name is displayed by the flag value. Also, bit symbols are converted to "Address.bit". Furthermore, an expression cannot be written which includes a bit symbol. Specification methods are shown below.

**Table 6-14. Format of Symbol Specification**

Conversion Object	Specification Method
Variable	var file#var (If a static variable with a file name attached is specified.) func#var (If a static variable with a function name attached is specified.) file#func#var (If a static function with a file name and function name attached is specified.)
Function	func file#func (If a static function with a file name attached is specified.)
Label	label file#label (If a local label with a file name attached is specified.)
Bit symbol	bitsym file#bitsym (If a local bit symbol with a file name attached is specified.)
Line number of source file	file#no
I/O Port	portname
I/O Register	I/O regname
Register	regname
PSW Flag	pswname

**Remark** var: Variable name, func: Function, file: File name, label: Label name, bitsym: Bit symbol name, portname: I/O port name, I/O regname: Peripheral I/O register name, regname: register, pswname: PSW flag name, no: Source file line number.

A "#" (sharp) is used as the separator between the file name and variable, function name and line number.

If the specified symbol could not be found inside the scope, all symbols are searched for (static variable, static function, local label, and local bit symbol).

In the default, the symbol name has priority. Furthermore, when desiring to change the priority order temporarily, adding a "\$" (Dollar) sign in front of the symbol causes the register name to have priority.

If you are reading multiple load modules, when specifying the load module names, use the separator "\$" between the load module name and the file name, variables, function name and symbol name.

In the default, the string selected in the accessed window is displayed, but as necessary, it is possible to change it by inputting from the keyboard. The radix of the default when inputting numerical values is decimal. When changing the contents of the symbol specification area, clicking the  button displays the conversion results in the conversion results display area.

By clicking the  button and displaying the input history, the contents input previously can be reused. Up to 16 items of input history can be retained.

**(2) Conversion results display area**

0xffe048

The variables specified in the symbol specification area, the function address, symbol values, line number addresses and expression values, etc. are displayed in this area.

**(3) Display radix specification area**

Radix:  Hex  Dec  Oct  Bin

This specifies the display radix of the conversion results display area.

Hex: Displayed in hexadecimal (default).

Dec: Displayed in decimal.

Oct: Displayed in octal.

Bin: Displayed in binary.

**Function buttons**



OK

When the contents of the symbol specification area have been changed, the symbols are converted. After conversion, when you are not changing the contents of the symbol specification area, the dialog box closes.

Cancel

Closes the Symbol To Address dialog box.

Restore

Returns the input data to the original values. If the  button has already been clicked, the data return to the state they were in just after the  button was clicked.

Help

Opens the help window.

<b>Source Window</b>	<b>MDI child window</b>
----------------------	-------------------------

### General


This window displays the source file or text file.

Also, display of disassembly of a program mixed with a source file, and execution of on-line assembly on a mixed display can be done.

The Source window has two states: active state and static state.

### Opening method

This window can be opened in the following methods.

- Automatically opened following the download of a load module file when a corresponding source file exists.
- In the main window
  - Select [Browse] → [SourceText] from the menu bar.
  - Press the [GRPH] + [B], and [S] keys in that order.
  - Select [Jump] → [SourceText] from the menu bar.
  - Press the [GRPH] + [J], and [S] keys in that order.
  - Press shortcut keys [CTRL] + [U].
- Click the  button on the toolbar.
- In the View File Load dialog box, select source file or text file, then click the  button.

### Window

Figure 6-32. Source Window (1/2)

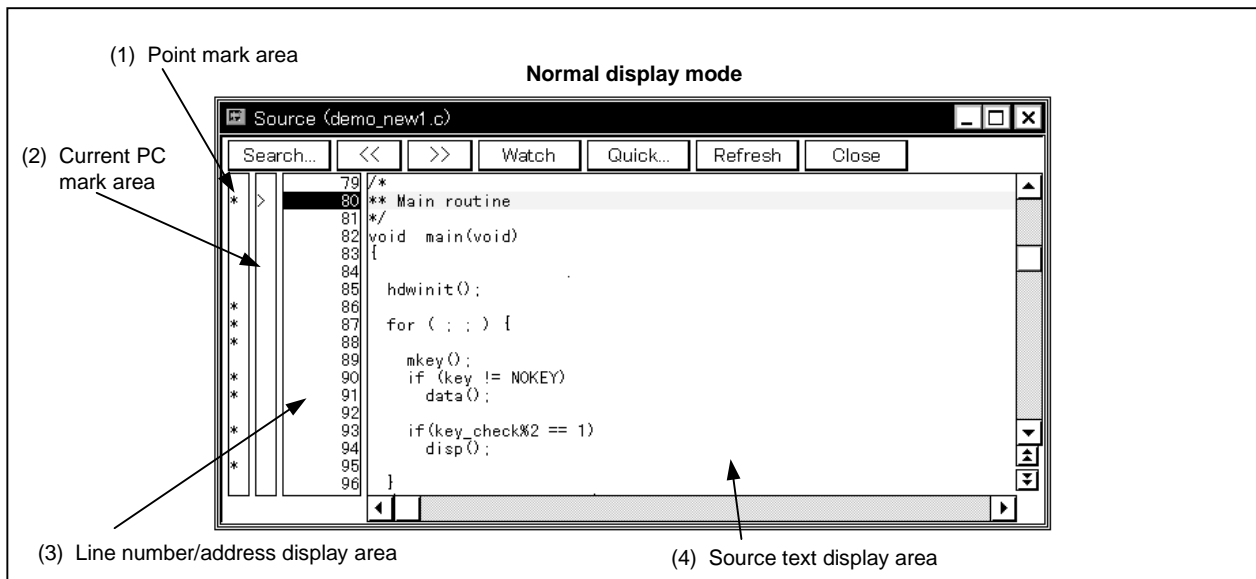
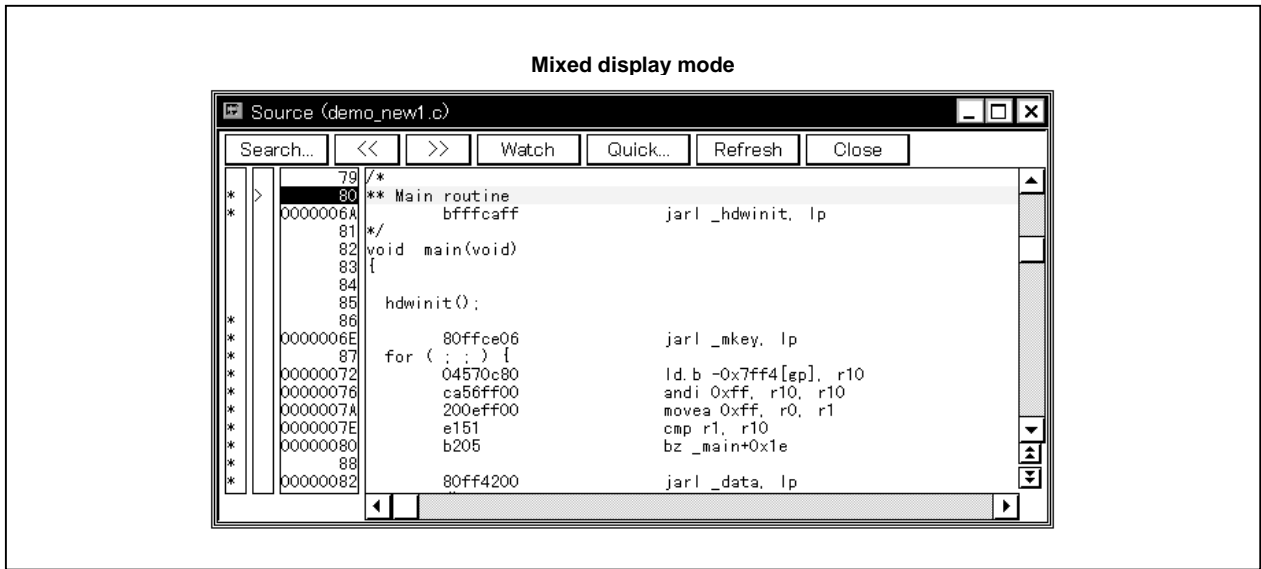


Figure 6-32. Source Window (2/2)



There are two modes in the Source window, normal display mode and mixed display mode. An explanation of each of these modes is given below.

Please refer to **(4) Source text display area** for further details.

#### (1) Normal display mode

In the normal display mode, other than display of source files, general text files can be displayed.

#### (2) Mixed display mode

In the mixed display mode, disassembly of programs combined with source files is displayed and on-line assembly is carried out. In on-line assembly, the cursor is moved to the mnemonic.

On-line assembly results are reflected in the memory window.

The display content in the mixed display mode can be saved as a display file.

Also, in the Source window which displays the source file that read the symbol information, there are two states, the active state and the static state.

Source windows in the active state can be opened one at a time only, and multiple windows in the static state can be opened at a time. Switching between states is accomplished in the [W]indow menu. Source files in which symbol information has not been read are read as text format files and their contents are not changed.

These two states are described below.

#### (1) Active state

The Source window in the active state is synchronized with the current PC value and the Source window's display is updated continuously so that it is always displaying the current PC line.

Also, if the Source window in the active state is synchronized not only with the jump function's jump destination but also with the Trace View window<sup>Note</sup>, the display in the Source window is updated in synch with the Trace View window.

The Source window first opened is in the active state.

**Note** Valid only for CPU cores with the TRCU incorporated.

**(2) Static state**

Source window in the static state is not synchronized with the current PC value and do not change the display position, but their contents are updated.

Also, a Source window in the static state does not become a jump destination and is not synchronized with the Trace View window.

If an active Source window is already open, a Source window is open in the static state.

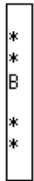
<b>Function</b>
-----------------

This window displays source files or text files. It also displays disassembled display of programs mixed with source files and can also perform on-line assembly in a mixed display. The tab size and display font can be specified in the Debugger Option dialog box.

The Source window consists of the following areas.

- Point mark area
- Current PC mark area
- Line number/address display area
- Source text display area

The function of each area is explained below.

**(1) Point mark area**

This area sets or deletes breakpoints, and displays the set status of each event condition and whether the program code exists or not.

**(a) Breakpoint setting/deleting function**

By clicking this area with the mouse, a breakpoint can be set or deleted.

The necessary mouse operations are as follows depending on the line where the mouse is clicked.

**Table 6-15. Breakpoint Setting/Deleting Function**

State of Line where Mouse is Clicked	Setting of On Mouse Click in Extended Option Dialog Box	Operation
Where blank or other marks than "B" mark are displayed (E, L, T mark etc.)	Soft Break	Sets software breakpoint
	Hard Break	Sets hardware breakpoint
Where "B" mark (blue) is displayed (software breakpoint)	Soft Break	Deletes software breakpoint
	Hard Break	Sets hardware breakpoint
Where "B" mark (red) is displayed (hardware breakpoint)	Soft Break	Sets software breakpoint
	Hard Break	Deletes hardware breakpoint
Where "B" mark (black) is displayed	Soft Break	If "B" mark is in software break, deletes breakpoint.
	Hard Break	If "B" mark is in hardware break, deletes breakpoint.

Setting and deletion of breakpoints can be performed for lines which have an "\*" displayed in the point mark area. As for lines without an "\*" displayed, setting or deletion can be done on either the line above or below the line where an "\*" is displayed.

If a breakpoint is set, the "B" mark is displayed in the point mark area. However, if some mark is already displayed, the "A" mark is displayed. The "A" mark shows that multiple events are set.

If a breakpoint is deleted, the "B" mark in the point mark area is deleted. However, if another event still remains, the mark of that event is displayed.

The "B" mark is displayed in the following colors in accordance with the type of breakpoint and its state.

**Table 6-16. Breakpoint Display Color**

Display Color	Explanation
Blue	Shows that the set software break is valid.
Red	Shows that the set hardware breakpoint is valid.
Black	Shows that the set breakpoint is invalid. Hardware breakpoints can be validated in the Event Manager or in the Break dialog box, and software breakpoints in the Software Break Manager.

If a breakpoint is set in the point mark area, it becomes valid at the time when it is set.

**(b) Event condition display function**

The set status of each event is displayed. If an execution event or access fetch event is set to the corresponding source line, a mark corresponding to the type of event is displayed.

**Table 6-17. Meaning of Event Display Mark**

Mark	Meaning
E	Set position of event condition
L	Final stage set position of event link
B	Set position of break event <sup>Note</sup>
T	Set position of trace event
A	Set position of two or more events

If a range address is specified as an event's address condition, the lower order address of the range becomes the object. Also, the address condition's mask specification is not reflected.

**Note** Valid only for CPU cores with the TRCU incorporated.

**(c) Program code display function**

Lines where an "\*" is displayed in the point mark area show that program code corresponding to the line number exists.

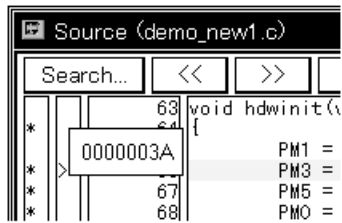
**Table 6-18. Program Codes**

Symbol	Explanation
With "*"	Shows that program code corresponding to the source file line No. exists. This is displayed only in cases where the load module is downloaded and symbol information has been read.
Without "*"	Shows that program code corresponding to the source file line No. does not exist. Also, file other than a source file, such as a text file is displayed, all the lines have no "*" displayed.

**(2) Current PC mark area**

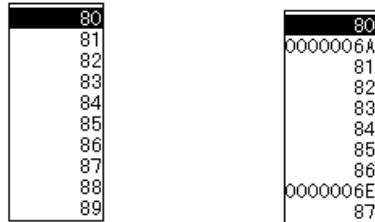
This area displays the mark ">" indicating the value of the current PC (PC register value). By clicking the mouse at this mark, the PC register value is displayed on the pop-up window.





The program is executed up to the specified line by double-clicking on the current PC mark area.

### (3) Line number/address display area



Normal display mode

Mixed display mode

Displays the line number of the source file or text file.

On lines where program code corresponding to the source file line number exists, the line number is displayed in red characters and on lines where no program code exists, the line number characters are displayed in black.


Also, the disassemble display's address is displayed in the mixed display mode. Address letters are displayed in gray.



The position of the line number/address display area can be selected as desired by clicking the mouse. A selected line number or address is displayed in reverse highlight.

The following functions are included in line number or address selection.

#### (a) Drag-and-drop function

The selected reverse highlighted line number or address can be dragged and dropped to another window or area.

If the line number or address position selected in the line number/address display area is dragged using the left button of the mouse, the mouse cursor's shape changes from an arrow to a .

If the cursor is moved into a window or area where it is possible to drop it, the shape of the mouse cursor changes from  to .

Operation is executed toward the address determined from the dropped address or line number in the window where they have been dropped.

Operation after drag-and-drop differs depending on the window or area where the items were dropped.

Table 6-19. Details of Drag and Drop Function (Line/Address)

Window or Area Which Is the Drop Object	Operation After Drop
<ul style="list-style-type: none"> <li>Event Manager</li> <li>Event manager area in each of the setting dialog boxes below <ul style="list-style-type: none"> <li>Event dialog box</li> <li>Event Link dialog box</li> <li>Break dialog box</li> <li>Trace dialog box<sup>Note</sup></li> </ul> </li> </ul>	<p>With the dropped line number or address as the address condition, the execution event conditions are generated automatically. The event condition names are generated automatically, as Evt00001, Evt00002, ... The path count is not specified. Address conditions are set in the form of symbol name + offset value for the symbol which is the closest.</p>
<ul style="list-style-type: none"> <li>Link condition setting area and disable condition setting area in the Event Link dialog box</li> <li>Break condition setting area in the Break dialog box</li> <li>Section trace start condition area, section trace end condition area, and delay trigger condition setting area in the Trace dialog box<sup>Note</sup></li> </ul>	<p>With the dropped line number or address as the address condition, the execution event conditions are generated automatically. Furthermore, automatically generated event conditions are set in the conditions setting area where they were dropped. The event condition names are generated automatically, as Evt00001, Evt00002, ... The path count is not specified. Address conditions are set in the form of symbol name + offset value for the symbol which is the closest.</p>
<ul style="list-style-type: none"> <li>Address condition setting area and data condition setting area in the Event dialog box</li> </ul>	<p>The line numbers or the address texts are set in the area where they were dropped. Address conditions are set in the form of symbol name + offset value for the symbol which is the closest.</p>

**Note** Valid only for CPU cores with the TRCU incorporated.

#### (b) Window link function

This function is a function which expresses the combination relationship between the Trace View window and Source window by line number or address. The Trace View window and the line number or address which is the link object are displayed in reverse highlight.

This function is valid only for CPU cores with the TRCU incorporated.

For details, refer to the window link function under **Trace View Window**.

#### (4) Source text display area

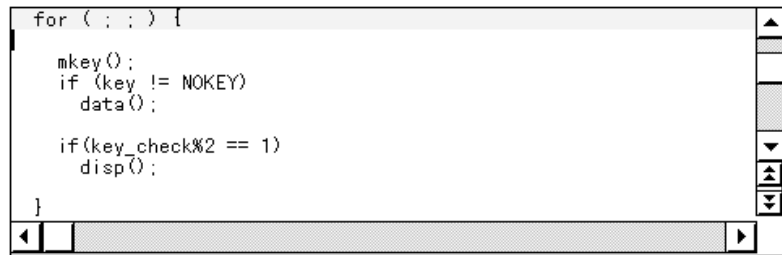
The source text display area performs display and disassembly display of source files and text files and on-line assembly.

The display contents of the source text display area can be scrolled using the vertical and horizontal scroll bars.

By selecting [View] → [Mix] on the menu bar, you can switch between the normal display mode and the mixed display mode.

**Example 1.** Normal display mode

The normal display mode is the mode in which the contents of source files or text files are viewed.



```

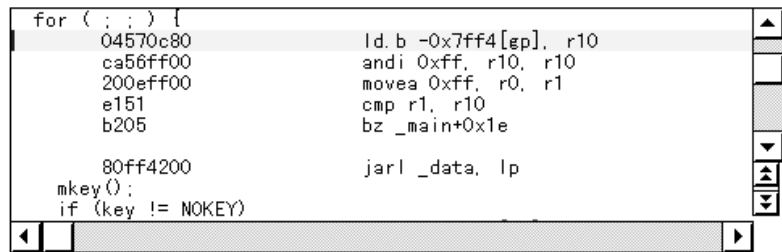
for ( ; ; ) {
    mkey();
    if (key != NOKEY)
        data();

    if(key_check%2 == 1)
        disp();
}

```

**2.** Mixed display mode

The mixed display mode is the mode where disassembly display in combination with source file display is performed.



for ( ; ; ) {	
04570c80	ld.b -0x7ff4[gp], r10
ca56ff00	andi 0xff, r10, r10
200eff00	movea 0xff, r0, r1
e151	cmp r1, r10
b205	bz _main+0x1e
80ff4200	jarl _data, lp
mkey();	
if (key != NOKEY)	

- If a line of the displayed source file corresponds to program code, the disassemble line is displayed following that source line. In the disassemble line, the address label, code data and disassembled mnemonic are displayed.
- The mnemonic's display starting position is adjusted according to the tab size setting value.
- The mixed display mode is valid only in cases where a load module is downloaded and symbol information is read into it, and a source file corresponding to the downloaded load module is displayed.
- Even if [View] → [Mix] on the menu bar is checked and mixed display mode is selected, in cases where a source file other than one corresponding to a load module is displayed, the display mode is the normal display mode.
- The disassemble display in the mnemonic column displayed in the mixed display mode can be changed. In the case of a change, move the cursor to the mnemonic column of the disassemble display line and change it. The overwrite mode and the insert mode can be switched during editing using the **INS** key.

If the cursor is moved to a different line after making changes in the mnemonic column, the change contents are checked. If the change contents are incorrect, the code data in the line where the change was made becomes an “\*”.

Even if moved to a different line, the change contents are not written to memory. A different line can be changed with the change contents not written to memory.

The changed contents of the mnemonic column are written to memory using the **Enter** key. The change contents are also checked if the **Enter** key is pressed. If the change contents are incorrect, the code data in the incorrect line become an “\*”. If there is even one incorrect line, the change contents are not written to memory. In such a case, please correct the change contents. If you are discarding the change contents, press the **ESC** key. If the change contents are not incorrect, press the **Enter** key. After the change contents are written to memory, the cursor moves to the next line's mnemonic column. Then the next line can be changed.

The current PC line and the breakpoint setting line are displayed with different display colors and in bold characters.

**Table 6-20. Current PC Line and Breakpoint Setting Line**

Meaning of Line	Display Color	Explanation
Current PC line	Yellow	This is the source line or the disassemble display line corresponding to the current PC value. It is the line where a “>” is displayed in the current PC mark area. When there is a mixed display, the display color of the disassemble display line only is changed. The source line is displayed with the normal display color.
Breakpoint setting line	Red	This is a source line or a disassemble display line where a valid breakpoint is set. It is the line where a red or blue “B” is displayed in the point mark area. When there is a mixed display, the display color of the disassemble display line only is changed. The source line is displayed with the normal display color.

The source text display area has a number of functions, including program execution with the line where the cursor is positioned as the object and setting of the breakpoint.

- Start function
- Come function
- Breakpoint setting function
- Program counter setting function
- Jump function

With these functions, if there is a cursor located on the source text line, the top address of the program code corresponding to the source line becomes the object and if the cursor is located on the disassemble display line, that address becomes the object. Also, if no program code exists on the source line, the top address of either the line above or the line below that line, where program code exists, becomes the object.

Furthermore, these functions cannot be executed in the following cases. The corresponding menu becomes gray in color and it becomes impossible to select it.

- In the case where a file other than a source file is being displayed.
- In the case where a user program is being executed.

Each function is explained below.

**(a) Start function**

Executes the user program from the line where the cursor is located.

This function is executed by performing the following operations in the sequential order.

- <1> Move the cursor to the line where execution is to start.
- <2> Perform one of the following operations in the main window.
  - Select [Run] → [Start From Here] from the menu bar.
  - Press the [GRPH] + [R], and [A] keys in that order.
  - Press the shortcut keys [SHIFT] + [f.6].

**(b) Come function**

Executes the user program up to the line where a cursor is placed.

While the user program is executed by this function, however, the currently set break event is not generated.

This function is executed by performing the following operations in the sequential order.

- <1> Move the cursor to the line at which a break is to occur.
- <2> Perform one of the following operations in the main window.
  - Select [Run] → [Come Here] from the menu bar.
  - Press the [GRPH] + [R], and [M] keys in that order.
  - Press the shortcut key [f.6].

**(c) Breakpoint setting function**

Sets a breakpoint to the line where a cursor is placed.

An execution event is used as the breakpoint to be set.

This function is executed by performing the following operations in the sequential order.

- <1> Move the cursor to the line on which a breakpoint is to be set.
- <2> Perform one of the following operations in the main window.
  - Select [Run] → [Break Point] or [Software Break Point] from the menu bar.
  - Press the [GRPH] + [R], and [B] keys or the [GRPH] + [R], and [F] keys in that order.
  - Press shortcut key [f.9] or [f.11].

**(d) Program counter setting function**

Sets the address of the line where a cursor is placed to the program counter (PC).

This function is executed by performing the following operations in the sequential order.

- <1> Move the cursor to the line where you desire to set the program counter (PC).
- <2> Perform one of the following operations in the main window.
  - Select [Run] → [Change PC] from the menu bar.
  - Press the [GRPH] + [R], and [H] keys in that order.
  - Press shortcut keys [CTRL] + [f.9].

**(e) Jump function**

Jumps to the Assemble window or Memory window by using the address of the line where a cursor is placed as a jump pointer. The jump destination window displays the jump pointer at the beginning. This function is executed by performing the following operations in the sequential order.

- <1> Move the cursor to the line which is the jump origin.
- <2> Perform one of the following operations in the main window.
  - If the jump destination is the Assemble window
    - Select [Jump] → [Assemble] from the menu bar.
    - Press the **GRPH** + **J**, and **A** keys in that order.
    - Press shortcut keys **CTRL** + **D**.
  - If the jump destination is the Memory window
    - Select [Jump] → [Memory...] from the menu bar.
    - Press the **GRPH** + **J**, and **E** keys in that order.
    - Press shortcut keys **CTRL** + **M**.

The text displayed in the source text display area can be selected by the following methods.

- From the desired position in the displayed text, drag the text to the desired position within the range of one line. Select the text from the drag start position to the end position.
  - Double-click on the character string. The words in the double-clicked position are selected.
- In the source text display area, there are a number of functions which can be performed with the selected text as the object, such as watch and quick watch.

**(f) Watch function**

Adds the selected variables, etc. to the Watch window and displays their contents. This function can be run by performing the following operations in order.

- <1> Select the text.
- <2> Perform one of the following operations in the main window.
  - Select [View] → [View Watch] from the menu bar.
  - Press the **GRPH** + **V**, and **I** keys in that order.
  - Click the **Watch** button in the Source window.

If the assembler symbol was selected, add it to the Watch window in accordance with the settings in the Debugger Option dialog box.

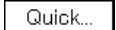
Apart from the above method, this function can also be run by performing the following operation.

- <1> Select the text.
- <2> Perform one of the following operations in the main window.
  - Select [View] → [Add Watch...] from the menu bar.
  - Press the **GRPH** + **V**, and **W** keys in that order.

By performing the above operation, the Add Watch dialog box is opened and the selected text is added to the Watch window. Furthermore, if the Watch window has not yet been opened, it will be opened.

**(g) Quick watch function**

This displays the contents of the selected variables, etc. temporarily in the Quick Watch dialog box. This function is run by performing the following operation in order.

- <1> Select the text.
- <2> Perform one of the following operations in the main window.
  - Select [View] → [Quick Watch...] from the menu bar.
  - Press the [GRPH] + [V], and [Q] keys in that order.
  - Press the shortcut keys [CTRL] + [W].
  - Click the  button in the Source window.

The Quick Watch dialog box will open and the contents will be displayed.

**(h) Balloon watch function**




This displays the contents of the selected variables, etc. temporarily in the selected place. This function is run by performing the following operation in order.

- <1> Select the text.
- <2> Place the mouse cursor on the selected text.
  - The tool tips view frame will be displayed and the contents will be displayed.

**Example** Display of contents

**(i) Drag-and-drop function**

Through a drag-and-drop operation, the selected text can be dragged to another window or area and dropped. The operation order is shown below.

- <1> Drag the selected text using the left button of the mouse. The mouse cursor's shape will change from an arrow to a .
- <2> If the mouse cursor is moved into a window or area where it is possible to drop it, the shape of the cursor changes from  to .

Operation after drag-and-drop differs depending on the window or area where the items were dropped.

Table 6-21. Details of Drag and Drop Function (Character String)

Window or Area Which Is the Drop Object	Operation After Drop												
<p>Event Manager</p> <p>Event manager area in each of the setting dialog boxes below</p> <ul style="list-style-type: none"> <li>• Event dialog box</li> <li>• Event Link dialog box</li> <li>• Break dialog box</li> <li>• Trace dialog box<sup>Note</sup></li> </ul>	<p>If it is possible to convert the dropped text to an address value as a symbol, with the converted address value as the address condition, the Access status (access all status) or the Execute status event conditions are generated automatically.</p> <p>The event condition names are generated automatically, as Evt00001, Evt00002, ... The data conditions and path count are not specified.</p> <p>The dropped text is set in the address conditions. The relationship between the generated event conditions and the symbols is as follows.</p> <table border="1"> <thead> <tr> <th>Symbol</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>Variable</td> <td>R/W</td> </tr> <tr> <td>Function</td> <td>Execute</td> </tr> <tr> <td>Symbols in the data section</td> <td>R/W</td> </tr> <tr> <td>Symbols in the code section</td> <td>Execute</td> </tr> <tr> <td>Other</td> <td>R/W</td> </tr> </tbody> </table>	Symbol	Status	Variable	R/W	Function	Execute	Symbols in the data section	R/W	Symbols in the code section	Execute	Other	R/W
Symbol	Status												
Variable	R/W												
Function	Execute												
Symbols in the data section	R/W												
Symbols in the code section	Execute												
Other	R/W												
<ul style="list-style-type: none"> <li>• Link condition setting area and disable condition setting area in the Event Link dialog box</li> <li>• Break condition setting area in the Break dialog box</li> <li>• Section trace start condition area, section trace end condition area, and delay trigger condition setting area in the Trace dialog box<sup>Note</sup></li> </ul>	<p>If it is possible to convert the dropped text to an address value as a symbol, with the converted address value as the address condition, the Access status (access all status) or the Execute status event conditions are generated automatically. Furthermore, the automatically generated event conditions are set in each dropped conditions setting area.</p> <p>The event condition names are generated automatically, as Evt00001, Evt00002, ... The data conditions and path count are not specified.</p> <p>The dropped text is set in the address conditions. The relationship between the generated event conditions and the symbols is as follows.</p> <table border="1"> <thead> <tr> <th>Symbol</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>Variable</td> <td>R/W</td> </tr> <tr> <td>Function</td> <td>Execute</td> </tr> <tr> <td>Symbols in the data section</td> <td>R/W</td> </tr> <tr> <td>Symbols in the code section</td> <td>Execute</td> </tr> <tr> <td>Other</td> <td>R/W</td> </tr> </tbody> </table>	Symbol	Status	Variable	R/W	Function	Execute	Symbols in the data section	R/W	Symbols in the code section	Execute	Other	R/W
Symbol	Status												
Variable	R/W												
Function	Execute												
Symbols in the data section	R/W												
Symbols in the code section	Execute												
Other	R/W												
<ul style="list-style-type: none"> <li>• Address condition setting area and data condition setting area in the Event dialog box</li> </ul>	Text is set in the dropped area.												
Watch window	If the dropped text can be recognized as a symbol, the symbol contents are displayed.												

**Note** Valid only for CPU cores with the TRCU incorporated.



**Function buttons****Search...**

Opens the Source Search dialog box and searches for source text character strings. If text is selected in the source text display area, the selected text is set as the search object and the Source Search dialog box is opened. If text has not been selected, the Source Search dialog box is opened with the search object in the blank state. The search method is specified in the Source Search dialog box. The search results are displayed as selected in the Source window. Performs the same operation as [V]iew → [S]earch on the menu bar.

&lt;&lt;

Searches for the text which matches the search conditions set in the Source Search dialog box in the backward direction from the cursor position (toward the top of the display). It changes to a **Stop** button during a search.

&gt;&gt;

Searches for the text which matches the search conditions set in the Source Search dialog box in the forward direction from the cursor position (toward the bottom of the display). It changes to a **Stop** button during a search.

**Stop**

Interrupts a search. During a search, the << button or the >> button changes to a **Stop** button.

**Watch**

Adds variables and other items selected in the source text display area to the Watch window. If the Watch window is currently not open, this button opens it. If text is not selected in the source text display area, the Watch window is opened only. This button operates the same as [V]iew → [V]iew Watch].

**Quick...**

Displays the selected variables and other contents of the source text display area temporarily in the Quick Watch dialog box. Opens the Quick Watch dialog box. If text is not selected in the source text display area, the Quick Watch dialog box is opened only. This button operates the same as [V]iew → [Q]uick Watch].

**Refresh**

Updates the contents of the window to the latest data.

**Close**

Closes this window.

**Caution**

If program code is described in an include file and if multiple files are included, the 1 to 1 relationship between line numbers and addresses disappears. In such include files, the function which uses the relationship between the line number and the address does not function normally.

<b>Source Search Dialog Box</b>	<b>Modeless</b>
---------------------------------	-----------------

### General

This dialog box searches file contents.

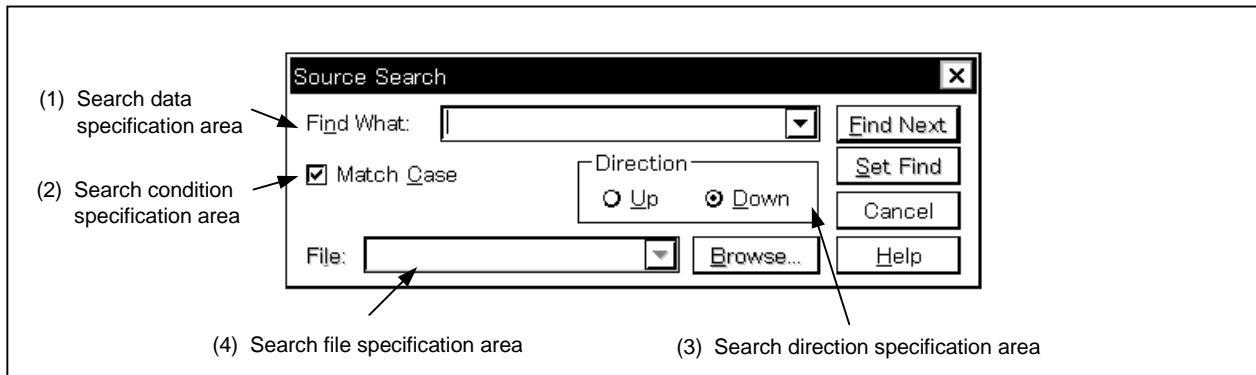
### Opening method

This dialog box can be opened in the following methods.

- When the Source window is active  
Select [View] → [Search...] from the menu bar.  
Press the [GRPH] + [V], and [S] keys in that order.  
Press the shortcut keys [CTRL] + [G].  
Click the [Search...] button.

### Window

**Figure 6-33. Source Search Dialog Box**



### Function

The Source Search dialog box consists of the following areas.

- Search data specification area
- Search condition specification area
- Search direction specification area
- Search file specification area


The function of each area is explained below.

**(1) Search data specification area**

 A rectangular box containing the text "Find What:" followed by a text input field and a small downward-pointing arrow button on the right side.

This area specifies the data to be searched.

Although the character string selected in the window that has called this dialog box is displayed as the default assumption, the character string can be changed by keyboard input as necessary.

By clicking the  button and displaying the input history, the contents input previously can be reused. Up to 16 items of input history can be retained.

**(2) Search condition specification area**

 A rectangular box containing a checked checkbox followed by the text "Match Case".

This area specifies whether the uppercase and lowercase characters of the searched data are distinguished by using the check box before search. As the default, uppercase and lowercase characters are distinguished.

- |                                     |                   |                         |
|-------------------------------------|-------------------|-------------------------|
| <input type="checkbox"/>            | <b>Match Case</b> | Does not distinguish    |
| <input checked="" type="checkbox"/> | <b>Match Case</b> | Distinguishes (default) |

**(3) Search direction specification area**

 A rectangular box with the title "Direction" at the top. Below the title are two radio buttons: one labeled "Up" and one labeled "Down". The "Down" radio button is selected.

This area specifies the direction of search. Two directions, upward and downward, can be selected.

- Up: Upward search. Searches the data from the current cursor position toward the beginning (upward on display).
- Down: Downward search. Searches the data from the current cursor position toward the end (downward on display).

**(4) Search file specification area**

 A rectangular box containing the text "File:" followed by a text input field and a small downward-pointing arrow button on the right side.

This is the area which specifies the search file. If the specification is omitted, a search is made within the accessed window.

**Function buttons****Find Next**

Searches the specified data under given conditions.

If a matching character string is found as a result of the search, it is displayed in reverse video.

To continue searching data, click this button again.

**Set Find**

Sets the specified conditions as search conditions and closes the dialog box.

**Stop**

Stops searching the data.

The **Cancel** button changes to the **Stop** button while data is being searched.**Cancel**

Closes the Source Search dialog box.

This button changes to the **Stop** button while data is being searched.**Help**

Opens the help window.

**Browse...**

Opens the Browse dialog box.


<b>Assemble Window</b>	<b>MDI child window</b>
------------------------	-------------------------

### General

This window displays a disassemble text. It can also execute on-line assemble.

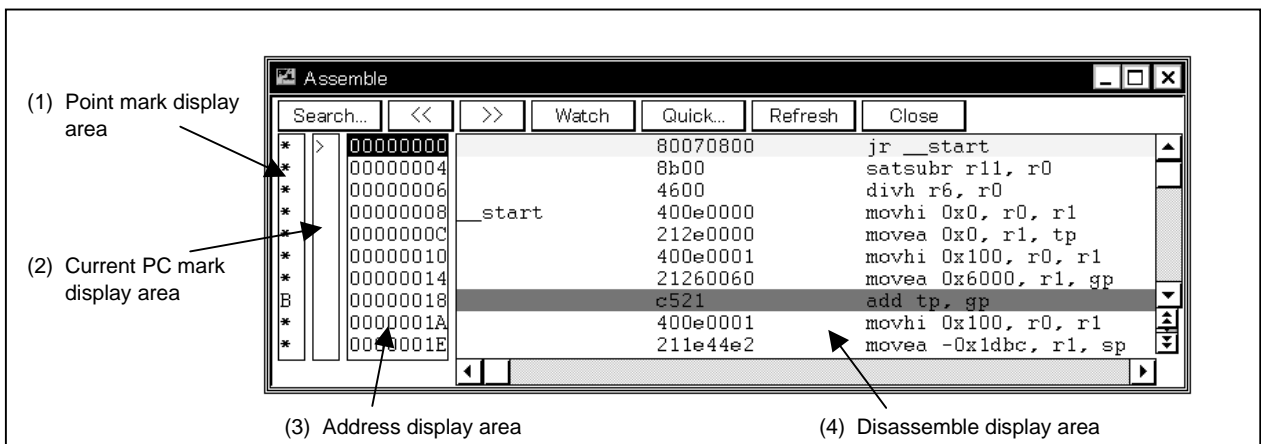
### Opening method

This window can be opened in the following methods.

- In the main window
  - Select [Browse] → [Assemble...] from the menu bar.
  - Press the [GRPH] + [B], and [A] keys in that order.
  - Select [Jump] → [Assemble...] from the menu bar.
  - Press the [GRPH] + [J], and [A] keys in that order.
  - Press the shortcut keys [CTRL] + [D].
- Click the  button on the toolbar.

### Window

Figure 6-34. Assemble Window



<b>Function</b>
-----------------

Performs disassemble display and on-line assembly.

On-line assembly is performed by moving the cursor to the mnemonic.

The on-line assembly results are also reflected in the Memory window.

The disassemble window has two states, the active state and the static state.

Only one disassemble window in the active state can be opened at a time, but multiple disassemble windows which are in the static state can be opened at the same time.

These two states are described below.

**(1) Active state**

The first Assemble window that is opened is in the active state.

The Assemble window's display is updated when it is in the active state so that the current PC line is displayed at all times in synch with the current PC value.

Also, when in the active state, the Assemble window becomes the jump destination for the jump function and, if it is synchronized with the Trace View window<sup>Note</sup>, it updates the Assemble window's display in synch with the Trace View window.

**Note** Valid only for CPU cores with the TRCU incorporated.

**(2) Static state**

An Assemble window in the static state is not synchronized with the current PC value and it displays a constant address.

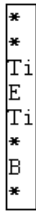
Also, an Assemble window in the static state cannot be made a jump destination, nor can it be synchronized with the Trace View window.

If an Assemble window has already been opened in the active state, then all subsequent windows are opened in the static state.

The Assemble window consists of the following areas.

- Point mark display area
- Current PC mark display area
- Address display area
- Disassemble display area

The function of each area is explained below.

**(1) Point mark display area**

This area sets or deletes breakpoints, and displays the setting status of each event.

**(a) Breakpoint setting/deleting function**

By clicking this area with the mouse, breakpoints can be set or deleted. The operation when clicking the mouse is as follows depending on the line where the mouse is clicked and the setting of On Mouse Click in the Extended Option dialog box. For details, refer to **Table 6-15 Breakpoint Setting/Deleting Function**.

If a breakpoint is set, the “B” mark is displayed in the point mark area. However, if there is a mark already displayed, the “A” mark is displayed.

The “A” mark shows that multiple events are set.

If a breakpoint is deleted, the “B” mark in the point mark area is deleted and the area is left blank or a mark other than the “B” mark is displayed.

However, if another event still remains, the mark of that event is displayed.

For details of the “B” mark, refer to **Table 6-16 Breakpoint Display Color**.

If a breakpoint is set in the point mark area, it becomes valid at the time when it is set.

**(b) Event condition display function**

The set status of each event is displayed. If the execution event, or the fetch condition of an access event is set, a mark indicating the type of each event condition is displayed in the corresponding assemble line. For details, refer to **Table 6-17 Meaning of Event Display Mark**.

If a range address is specified as an event’s address condition, the lower order address of the range becomes the object. Also, the address condition’s mask specification is not reflected.

**(2) Current PC mark display area**

This area displays mark “>” indicating the value of the current PC (PC register value). The program is executed up to the specified line by double-clicking on the current PC mark area.


**(3) Address display area**



This area displays the disassemble start address. When this field is clicked, an address is displayed in reverse video and selected.

In addition, this area has the following functions.

**(a) Drag-and-drop function**

A selected line number or address which is reverse highlighted can be dragged to another window or area and dropped.

If the mouse cursor is positioned on the position of the selected line number or address in the line number/address display area and the left button is pressed and held down, then when the mouse is dragged, the arrow cursor form changes to a .

If the mouse cursor is moved into a window or area where it is possible to drop it, the shape of the cursor changes form  to .

Operation is performed with respect to the dropped address in the window where it is dropped.

Operations after an address is dropped depend on the respective window or area where it is dropped.

For details, refer to **Table 6-19 Details of Drag and Drop Function (Line/Address)**.

**(b) Window link function**

This function is a function which shows the combination relationship between the Trace View window and the disassemble display by an address.

The Trace View window and the address which is the link object is displayed in reverse highlight.

This function is valid only for CPU cores with the TRCU incorporated.

For details, refer to the window link function of the **Trace View Window**.



**(4) Disassemble display area**

<code>_main</code>	<code>952d</code>	<code>br _main+0x52</code>
	<code>bfffcaff</code>	<code>jarl _hdwinit, lp</code>
	<code>80ffce06</code>	<code>jarl _mkey, lp</code>
	<code>04570c80</code>	<code>ld.b -0x7ff4[gp], r10</code>
	<code>ca56ff00</code>	<code>andi 0xff, r10, r10</code>
	<code>200eff00</code>	<code>movea 0xff, r0, r1</code>
	<code>e151</code>	<code>cmp r1, r10</code>
	<code>b205</code>	<code>bz _main+0x1e</code>

The address label, code data and disassembled mnemonic are displayed in the disassemble display area.

The disassemble display in the mnemonic column can be changed.

If you are changing it, move the cursor to the mnemonic column to make the change. During editing, the **INS** key can be used to switch between the overwrite mode and the insert mode.

After making the desired changes in the mnemonic column, if the cursor is moved to a different line, the contents of the change are checked. If the change contents are incorrect, the code data in the changed line becomes an “\*”.

Even if it is moved to a different line, the change contents are not written to memory. You can change another line without the changes being written to memory.

The changed contents in the mnemonic column are written to memory by pressing the **Enter** key. The change contents are also checked when the **Enter** key is pressed. If the change contents are incorrect, the code data in the incorrect line become an “\*”. If even one line is incorrect, the change contents are not written to memory. In such a case, please correct the change contents. If the change contents are being discarded, press the **ESC** key. If the change contents are not incorrect, if the **Enter** key is pressed, after the change contents are written to memory, the cursor moves to the next line’s mnemonic column. In this way changes can continue to be made in the next line.

If, as a result of the change, the number of bytes in the instructions after the change is fewer than the number of bytes in the instruction before the change, the remaining bytes are replaced by the “NOP” instruction.

Also, if, as a result of the change, the number of bytes in the instructions after the change is greater than the number of bytes in the instruction before the change, the next instruction is overwritten. In this case also, the remaining bytes are replaced by the “NOP” instruction. Even in cases where the change runs into the source line, it is overwritten in the same way.

As for the current PC line and the breakpoint setting line, the display color is changed and they are displayed in bold.

For details, refer to **Table 6-20 Current PC Line and Breakpoint Setting Line**.

The disassemble display area includes a number of functions such as program execution with the line where the cursor is located as the object, or setting of the breakpoint, etc.

- Start function
- Come function
- Breakpoint setting function
- Program counter setting function
- Jump function

Furthermore, in the following case, these functions cannot be executed. The corresponding menu is displayed in gray and it becomes impossible to select them.

- If a user program is being executed.

Each function is explained below.

**(a) Start function**

Executes the user program from the line where the cursor is located.

This function is executed by performing the following operations in the sequential order.

- <1> Move the cursor to the line where execution is to start.
- <2> Perform one of the following operations in the main window.
  - Select [Run] → [Start From Here] from the menu bar.
  - Press the [GRPH] + [R], and [A] keys in that order.
  - Press the shortcut keys [SHIFT] + [f.6].

**(b) Come function**

Executes the user program up to the line where a cursor is placed.

While the user program is being executed in this mode, however, the break event currently set is not generated.

This function is executed by performing the following operations in the sequential order.

- <1> Move the cursor to the line where execution is to start.
- <2> Perform one of the following operations in the main window.
  - Select [Run] → [Come Here] from the menu bar.
  - Press the [GRPH] + [R], and [M] keys in that order.
  - Press the shortcut key [f.6].

**(c) Breakpoint setting function**

Sets a breakpoint to the line where a cursor is placed.

An execution event is used as the breakpoint to be set.

This function is executed by performing the following operations in the sequential order.

- <1> Move the cursor to the line to which a breakpoint is to be set.
- <2> Perform one of the following operations in the main window.
  - Select [Run] → [Break Point] or [Software Break Point] from the menu bar.
  - Press the [GRPH] + [R], and [B] keys or the [GRPH] + [R], and [F] keys in that order.
  - Press shortcut key [f.9] or [f.11].

**(d) Program counter setting function**

Sets the address of the line where a cursor is placed to the PC (program counter).

This function is executed by performing the following operations in order.

- <1> Move the cursor to the line where you desire to set the program counter (PC).
- <2> Perform one of the following operations in the main window.
  - Select [Run] → [Change PC] from the menu bar.
  - Press the [GRPH] + [R], and [H] keys in that order.
  - Press shortcut keys [CTRL] + [f.9].

**(e) Jump function**

Jumps to the Source window or Memory window by using the address of the line where a cursor is placed as a jump pointer.

The jump destination window displays the jump pointer at the beginning.

This function is executed by performing the following operations in order.

- <1> Move the cursor to the line which is the jump origin.
- <2> Perform one of the following operations in the main window.
  - If the jump destination is the Source window
    - Select [Jump] → [SourceText] from the menu bar.
    - Press the [GRPH] + [J], and [S] keys in that order.
    - Press shortcut keys [CTRL] + [U].
  - If the jump destination is the Memory window
    - Select [Jump] → [Memory] from the menu bar.
    - Press the [GRPH] + [J], and [E] keys in that order.
    - Press shortcut keys [CTRL] + [M].

The text displayed in the disassemble display area can be selected by the following method.

- From the position where the desired text starts in the display, drag the mouse to the position within the range of 1 line where the desired text ends. The text from the starting position to the end position is the text that is selected.
- Double-click on the character string. The words in the double-clicked position are selected.

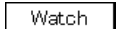
In the disassemble display area, a number of functions, such as watch and quick watch with the selected text as the object, can be performed.

**(f) Watch function**

This adds the selected data to the Watch window and displays its contents.

Furthermore, if an assembler symbol is selected, it is added to the Watch window in accordance with the settings in the Debugger Option dialog box.

This function is executed by performing the following operations in order.

- <1> Select the text.
- <2> Perform one of the following operations in the main window.
  - Select [View] → [View Watch] from the menu bar.
  - Press the [GRPH] + [V], and [I] keys in that order.
  - Select [View] → [Add Watch...] from the menu bar.
  - Press the [GRPH] + [V], and [W] keys in that order.
  - Click the  button in the Assemble window.

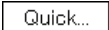
If any of these operations is carried out, the Add Watch dialog box opens and adding is performed.

If the Watch window is not currently open, the Watch window is opened.

**(g) Quick watch function**

This temporarily displays the contents of the selected data in the Quick Watch dialog box.

This function is executed by performing the following operations in order.

- <1> Select the text.
- <2> Perform one of the following operations in the main window.
  - Select [View] → [Quick Watch...] from the menu bar.
  - Press the [GRPH] + [V], and [Q] keys in that order.
  - Press the shortcut keys [CTRL] + [W].
  - Click the  button in the Assemble window.

The Quick Watch dialog box opens and its contents are displayed.

**(h) Balloon watch function**

This temporarily displays the contents of the selected data in the selected place.




This function is executed by performing the following operations in order.

- <1> Select the text.
- <2> Position the mouse cursor on top of the selected text.
  - A tool tip type frame is displayed and the contents are displayed.

**Example** Display of contents

**(i) Drag-and-drop function**

Through a drag-and-drop operation, the selected text can be dragged to another window or area and dropped. The operation order is shown below.

- <1> Drag the selected text using the left button of the mouse. The mouse cursor's shape will change from an arrow to a .
- <2> If the mouse cursor is moved into a window or area where it is possible to drop it, the shape of the cursor changes from  to .

Operation after drag-and-drop differs depending on the window or area where the items were dropped. For details, refer to **Table 6-21 Details of Drag and Drop Function (Character String)**.

## Function buttons

Search...

Opens the Assemble Search dialog box and searches the character string of a mnemonic. If text has been selected in the disassemble display area set the selected text as the search object and open the Assemble Search dialog box.

If text has not been selected, the Assemble Search dialog box will open with the search object in the blank state.

Specify the search method in the Assemble Search dialog box. The search results are selectively displayed in the Assemble window.

Operation is the same as when [View] → [Search...] is selected on the menu bar.

<<

Searches for the contents which match with the search conditions set in the Assemble Search dialog box before the cursor position (toward the top of the display).

During the search, it changes to a **Stop** button.

>>

Searches for the contents which match with the search conditions set in the Assemble Search dialog box after the cursor position (toward the bottom of the display).

During the search, it changes to a **Stop** button.

Stop

Interrupts the search.

During a search, the << button or the >> button changes to the **Stop** button.

Watch

Adds the symbol, etc. selected in the disassemble display area to the Watch window.

If the Watch window has not been opened, it opens the Watch window.

If text has not been selected in the disassemble display area, the Watch window opens only.

Operation is the same as when [View] → [View Watch] is selected on the menu bar.

Quick...

Temporarily displays the contents of symbols, etc. selected in the disassemble text display area in the Quick Watch dialog box.

It opens the Quick Watch dialog box.

If text has not been selected in the disassemble text display area, the Quick Watch dialog box opens only.

Operation is the same as when [View] → [Quick Watch] is selected on the menu bar.

Refresh

Updates the window contents to the latest data.

Close

Closes the Assemble window.

<b>Assemble Search Dialog Box</b>	<b>Modeless</b>
-----------------------------------	-----------------

### General

Searches the contents of the Assemble window.

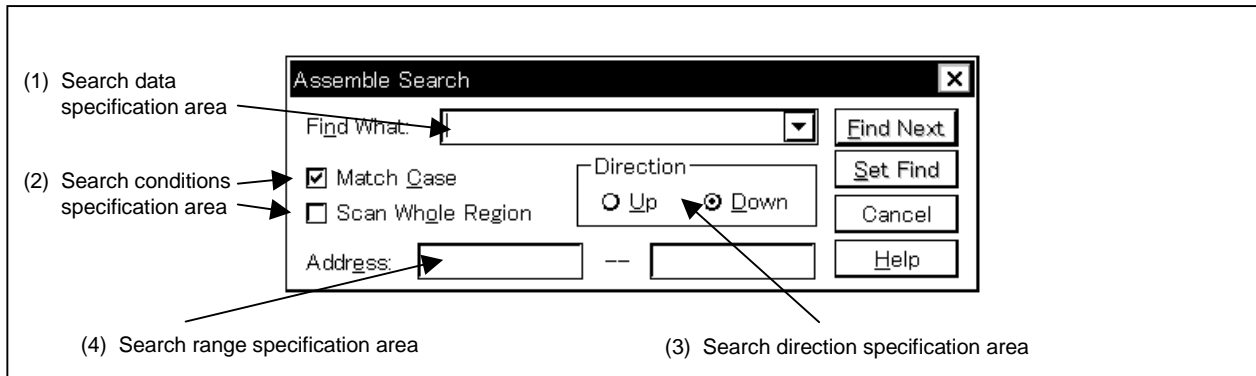
### Opening Method

This dialog box can be opened by one of the following methods when the Assemble window is the current window.

- In the main window  
Select [View] → [Search...] from the menu bar.  
Press the [GRPH] + [V], and [S] keys in that order.  
Press the shortcut keys [CTRL] + [G].
- In the Assemble window  
Click the [Search...] button.

### Window

**Figure 6-35. Assemble Search Dialog Box**



### Function

This initiates a search for the disassemble contents. Continuous null characters in which are included input character strings and disassemble character strings are compared as a null character.

The Assemble Search dialog box consists of the following areas.


- Search data specification area
- Search conditions specification area
- Search direction specification area
- Search range specification area

The function of each area is explained below.

**(1) Search data specification area**

This specifies the data that are to be searched for.

In the default, the string selected in the accessed window is displayed, but as necessary, it can be changed by inputting from the keyboard.

By clicking the  button and displaying the input history, the contents input previously can be reused. Up to 16 items of input history can be retained.

**(2) Search conditions specification area**

(a)  Match Case

When searching, specification is made by the check box whether or not any distinction will be made in the search data between uppercase and lowercase letters.

In the default setting, the data is case sensitive.

- Match Case      Not distinguished.  
 Match Case      Distinguished (default)

(b)  Scan Whole Region

When searching, this check box specifies whether to search the whole range or not. In the default, only the remaining portion of the range is searched.

- Scan Whole Region      Only the remaining portion of the range is searched (default).  
 Scan Whole Region      The whole range is searched.

**(3) Search direction specification area**

Direction  
 Up     Down

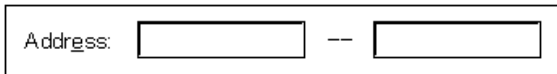
This specifies the search direction.

There are two types of search, search up, and search down.

Up:      This is a search before the current cursor position. Search is made up to the current cursor position (toward the top of the screen).

Down:    This is a search after the current cursor position. Search is beyond the current cursor position (toward the bottom of the screen).

#### (4) Search range specification area



Address:  --

This is the area where the addresses being searched are specified. A symbol or expression can also be specified for the address specification.

The specification method is the same as in the case of the Address Move dialog box. The default radix when inputting numerical values is hexadecimal.

#### Function buttons

**Find Next**

Searches for the specified search data in accordance with the conditions. As the search results, the matching character string is displayed in reverse highlight. If you are searching continuously, press this button again.

**Set Find**

Sets the specified conditions as the search conditions and closes the dialog box.

**Stop**

Interrupts the search for data.

During the data search, the **Cancel** button changes to the **Stop** button.

**Cancel**

Closes the Assemble Search dialog box.

During a data search, it changes to the **Stop** button.

**Help**

Opens the help window.




<b>Memory Window</b>	<b>MDI child window</b>
----------------------	-------------------------

**General**

This window displays and changes the memory contents.

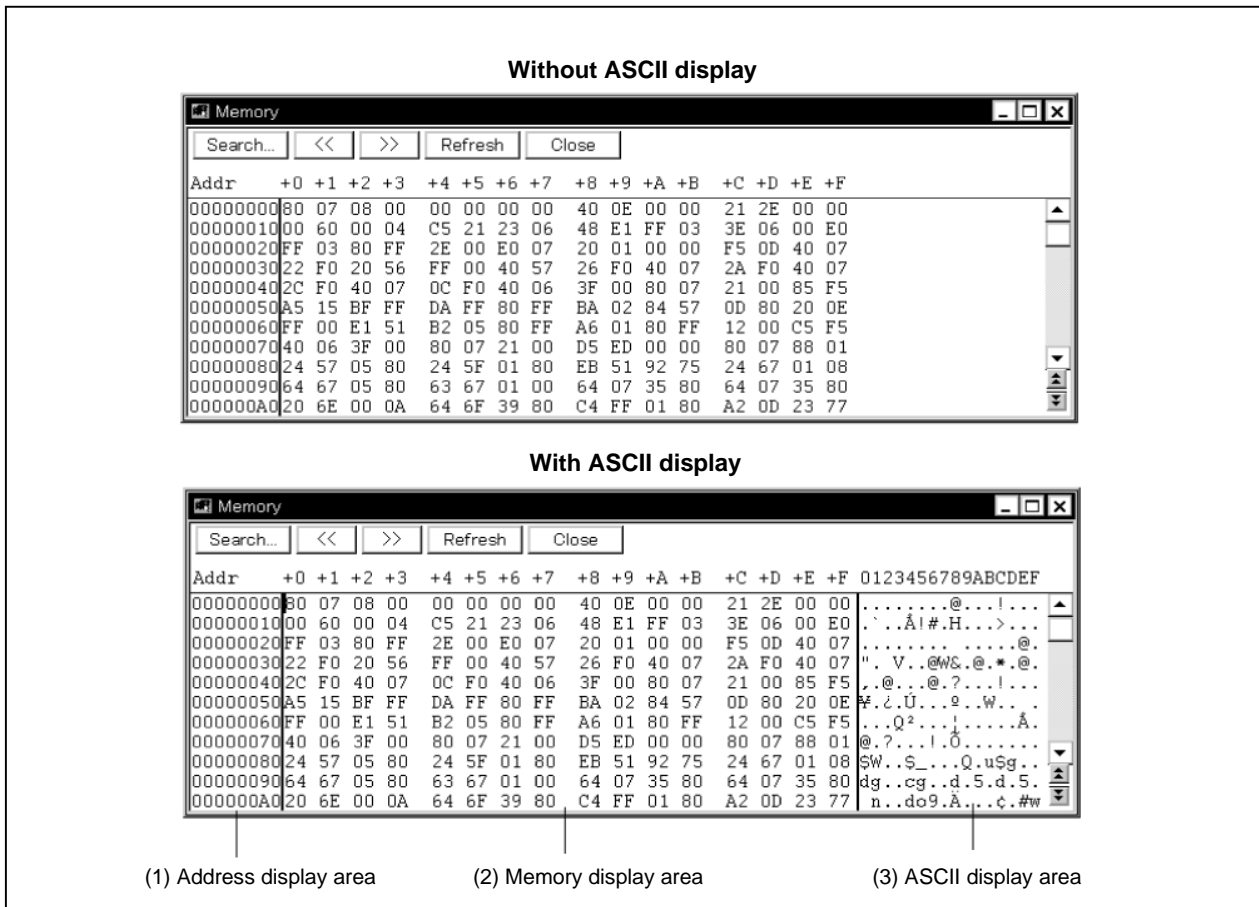
**Opening method**

This window can be opened in the following methods.

- In the main window  
 Select [Browse] → [Memory] from the menu bar.  
 Press the [GRPH] + [B], and [M] keys in that order.  
 Select [Jump] → [Memory] from the menu bar.  
 Press the [GRPH] + [J], and [E] keys in that order.  
 Click the  button on the toolbar.

**Window**

Figure 6-36. Memory Window





<b>Function</b>
-----------------

The memory contents are displayed and changed.

When the Memory window opens, the display start position is as follows.

- First time                                      Starts from the top address of the real-time RAM sampling area when the real-time RAM sampling function is valid.  
When the function is invalid, the display starts from the top address of the internal RAM area.
- Second and subsequent times            Starts from the place where the last active window closed. This applies to both active and static windows (if the active window has never been closed, the start position is the same as “first time” above).

A vertical scroll bar is always displayed so that the memory contents can be scrolled.

You can also use the  button or the  button to move the display one page up or one page down.

Also, even while user program is executed, any 1 KB of internal RAM area (the display range is set in the Extended Option dialog box) can be displayed in real time.

The Memory window has two states, the active state and the static state.

There can be only one Memory window in the active state at a time, but multiple windows can be opened if they are in the static state.

Each Memory window state is explained below.

#### (1) Active state

The Memory window in the active state is the jump destination of the jump function and the memory window's display is updated in synchronization with the Trace View window if it is synchronized with the Trace View window<sup>Note</sup>. The Memory window first opened is in the active state.

**Note** Valid only for CPU cores with the TRCU incorporated.

#### (2) Static state

Memory window in the static state is not the jump destination and do not synchronize with the Trace View window. Except for those points, it operates in the same way as in the active state.

If an active Memory window is already open, a Memory window is open in the static state.

The Memory window consists of the following areas.

- Address display area
- Memory display area
- ASCII display area

The function of each area is explained below.

**(1) Address display area**

```

Addr
00000000
00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
000000A0

```

This area displays memory addresses.

**(2) Memory display area**

+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
80	07	08	00	00	00	00	00	40	0E	00	00	21	2E	00	00
40	0E	00	01	21	26	00	60	C5	21	40	0E	00	01	21	1E
44	E2	40	0E	00	01	21	F6	00	E0	80	FF	3E	00	E0	07
20	01	00	00	D5	15	40	07	22	F0	20	56	FF	00	40	57
26	F0	40	07	2A	F0	40	07	20	F0	40	07	00	F0	40	07
24	F0	40	07	36	F0	23	FF	01	00	44	1A	7F	00	5C	1A
63	FF	01	00	95	ED	00	00	95	2D	BF	FF	CA	FF	80	FF
CE	06	04	57	0C	80	CA	56	FF	00	20	0E	FF	00	E1	51
B2	05	80	FF	42	00	24	5F	15	80	58	1A	63	37	01	00
63	3F	05	00	02	32	0B	38	80	FF	00	08	23	3F	05	00
06	58	23	37	01	00	48	1A	61	5A	BA	05	80	FF	F8	03

This is where the memory contents are displayed and changed. The contents to be changed are displayed in red characters, then by pressing the **Enter** key, they are actually written to the target. The contents prior to the change (when in the red character state) can be erased using the **ESC** key.

Up to 256 bytes of data can be changed at one time.

This area also has the following two functions in addition to displaying the memory contents making changes.

**(a) Jump function**

Jumps to the Source window or Assemble window by using the address at the cursor position as a jump pointer. The jump destination window displays the jump pointer at the beginning.

This function is executed by performing the following operations in the sequential order.

- <1> Move the cursor to the line which is the jump origin.
- <2> Perform one of the following operations in the main window.
  - If the jump destination is the Source window
    - Select [Jump] → [SourceText] from the menu bar.
    - Press the **GRPH** + **J**, and **S** keys in that order.
    - Press shortcut keys **CTRL** + **U**.
  - If the jump destination is the Assemble window
    - Select [Jump] → [Assemble] from the menu bar.
    - Press the **GRPH** + **J**, and **A** keys in that order.
    - Press shortcut keys **CTRL** + **D**.

**(b) Window link function**

Displays the combination relationship between the Trace View window and the Memory window as memory contents. The relevant address is displayed in reverse highlight.

This function is valid only for CPU cores with the TRCU incorporated.

For details, refer to the window link function under the **Trace View Window**.

**(3) ASCII display area**

```

0123456789ABCDEF
.....@...!...
. ^ . Å ! # . H . . > . . .
.....@.
" . V . . @ W & . @ . * . @ .
. . @ . . . @ . ? . . . ! . . .
. ¢ . Û . . . e . . W . . .
. . . Q 2 . . . ! . . . . Å .
@ . ? . . . ! . Ö . . . . .
$W . . $ _ . . . Q . u $ g . .
dg . . cg . . d . 5 . d . 5 .
n . . do9 . Å . . . ç . #w

```

This area displays the memory contents in ASCII. The memory contents can be converted into ASCII characters for display. The conversion method is the same as the memory display area.

The display can be turned on/off by selecting [View] → [Ascii] on the menu bar (display is on as default assumption).

This area is also a jump pointer for the jump function, just like the **(2) Memory display area**.

**Function buttons****Search...**

Opens the Memory Search dialog box and searches for a character string in the displayed memory contents or memory contents. At this time, the selected data (memory value) is displayed in the Memory Search dialog box as the subject to search. If the dialog box is opened without data specified, specify data by key input. The result of the search is displayed in reverse video in the Memory window.

&lt;&lt;

Searches for the memory contents which match the search conditions set in the Memory Search dialog box in the backward direction from the cursor position (toward the top of the display). It changes to a **Stop** button during a search.

&gt;&gt;

Searches for the memory contents which match the search conditions set in the Memory Search dialog box in the forward direction from the cursor position (toward the bottom of the display). It changes to a **Stop** button during a search.

**Stop**

Interrupts a search.

During a search, the **>>** button or the **<<** button changes to a **Stop** button.

**Refresh**

Updates the window's contents with the latest data.

**Close**

Closes this window.

<b>Memory Search Dialog Box</b>	<b>Modeless</b>
---------------------------------	-----------------

### General

This window searches the memory contents.

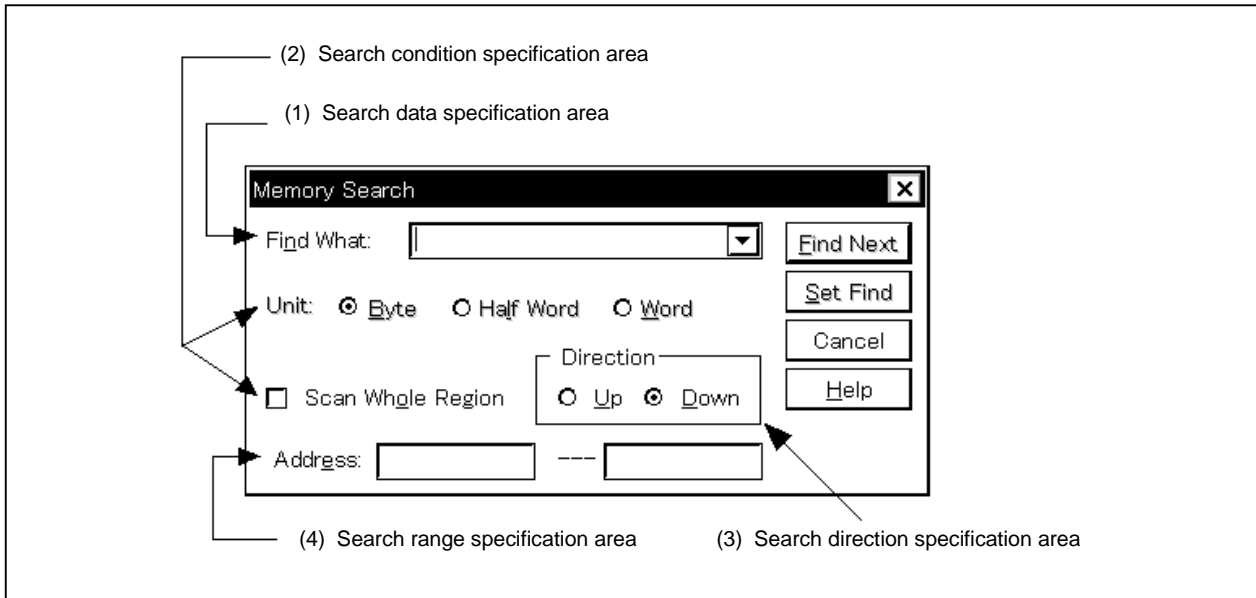
### Opening method

This window can be opened in the following methods.

- Set the Memory window as the current window and in the main window;  
Select [View] → [Search...] from the menu bar.  
Press the [GRPH] + [V], and [S] keys in that order.  
Press the shortcut keys [CTRL] + [G].
- In the Memory window  
Click the [Search...] button.

### Window

Figure 6-37. Memory Search Dialog Box



<b>Function</b>
-----------------

This searches the contents of memory for the Memory window where the cursor is positioned.

If the cursor is in the memory display area, the specified data is regarded as a binary data string and the memory display area is searched.

If the cursor is in the ASCII display area, the specified data is regarded as an ASCII character string and the ASCII display area is searched.

Unmapped areas, peripheral I/O register areas and I/O protected areas are not searched.

The Memory Search dialog box consists of the following areas.

- Search data specification area
- Search condition specification area
- Search direction specification area
- Search range specification area

The function of each area is explained below.

### (1) Search data specification area

Find What: <input type="text"/>
---------------------------------

This area specifies the data to be searched.

Although the character string selected in the window that has called this dialog box is displayed as the default assumption, the character string can be changed by keyboard input as necessary (input cannot be done using a symbol or expression.)

When searching the memory display area, a maximum of 16 items of data can be specified at one time. Specify each item of data using a space as a delimiter.

When searching the ASCII display area, a maximum of 256 characters of data can be specified. Spaces in the data are treated as null characters.

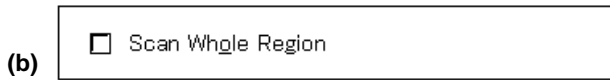
By clicking the  button and displaying the input history, the contents input previously can be reused. Up to 16 items of input history can be retained.

### (2) Search condition specification area

(a) Unit: <input checked="" type="radio"/> <u>B</u> yte <input type="radio"/> Half Word <input type="radio"/> <u>W</u> ord
--

This area specifies what bits of data each specified search data is to be regarded as by using the check box before search. As the default, the data is searched as 8-bit data.

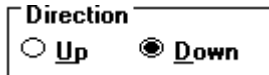
- |   |                                |
|---|--------------------------------|
| <input checked="" type="radio"/> <u>B</u> yte | Search as 8-bit data (default) |
| <input type="radio"/> Half Word               | Search as 16-bit data          |
| <input type="radio"/> <u>W</u> ord            | Search as 32-bit data          |



This is a check box which specifies whether to search the whole specified region or not when conducting search. With the default, only the portion remaining in the range is searched.

- Scan Whole Region Searches the remaining portion of the range (default).  
 Scan Whole Region Searches the whole range.

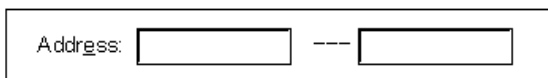
### (3) Search direction specification area



This area specifies the direction of search. Two directions, upward and downward, can be selected.

- Up: Upward search. Searches the data from the current cursor position toward the beginning (upward on display).
- Down: Downward search. Searches the data from the current cursor position toward the end (downward on display).

### (4) Search range specification area



This is the area where the addresses being searched are specified. A symbol or expression can also be specified for the address specification.

The specification method is the same as in the case of the Address Move dialog box. The default radix when inputting numerical values is hexadecimal.

### Function buttons

- Find Next** Searches for the specified search data in accordance with the conditions. As the search results, the matching character string is displayed in reverse highlight. If you are searching continuously, press this button again.
- Set Find** Sets the specified conditions as the search conditions and closes the dialog box.
- Stop** Interrupts the search for data.  
 During the data search, the **Cancel** button changes to the **Stop** button.
- Cancel** Closes the Memory Search dialog box.  
 During a data search, it changes to the **Stop** button.
- Help** Opens the help window.



<b>Memory Fill Dialog Box</b>	<b>Modal</b>
-------------------------------	--------------

### General

This dialog box initializes the memory contents with a specified code.

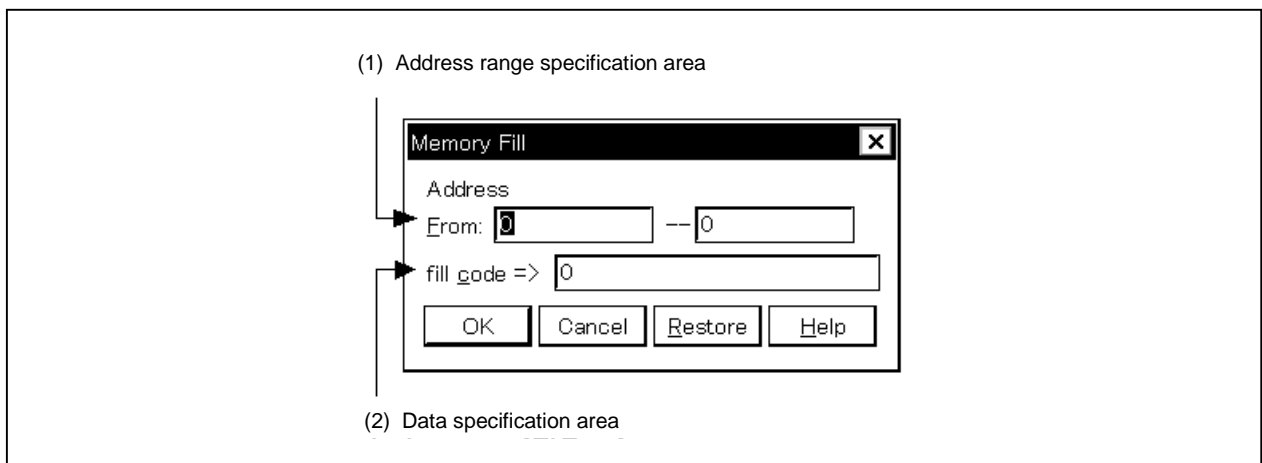
### Opening method

This dialog box can be opened in the following methods.

- In the main window  
Select [Edit] → [Memory] → [Fill...] from the menu bar.  
Press the [GRPH] + [E], [M], and [F] keys in that order.

### Window

**Figure 6-38. Memory Fill Dialog Box**



### Function

The Memory Fill dialog box consists of the following areas.

- Address range specification area
- Data specification area

The function of each area is explained below.

**(1) Address range specification area**

Address	
From: <input type="text" value="0"/>	-- <input type="text" value="0"/>

This area specifies the address range of the memory contents to be initialized.

A symbol or expression can also be specified for the address specification. The default radix when inputting numerical values is hexadecimal.

Specify the first address for initialization and then the end address, in that order.

**(2) Data specification area**

fill code => <input type="text" value="0"/>
---

This area specifies the data to be initialized.

Binary data strings (byte data strings) with up to 16 items of data can be specified. Specify each item of data using a space as a delimiter.

The default radix is hexadecimal and the radix can be specified for each item of data.

**Function buttons**

OK

Initializes the memory contents.

Cancel

Closes this dialog box.

This button changes to the  button during memory initialization.

Stop

Stops initializing the memory.

The  button changes to the  button during memory initialization.

Restore

Returns the input data to the original value.

Help

Opens the help window.

<b>Memory Copy Dialog Box</b>	<b>Modal</b>
-------------------------------	--------------

### General

This dialog box copies the memory contents.

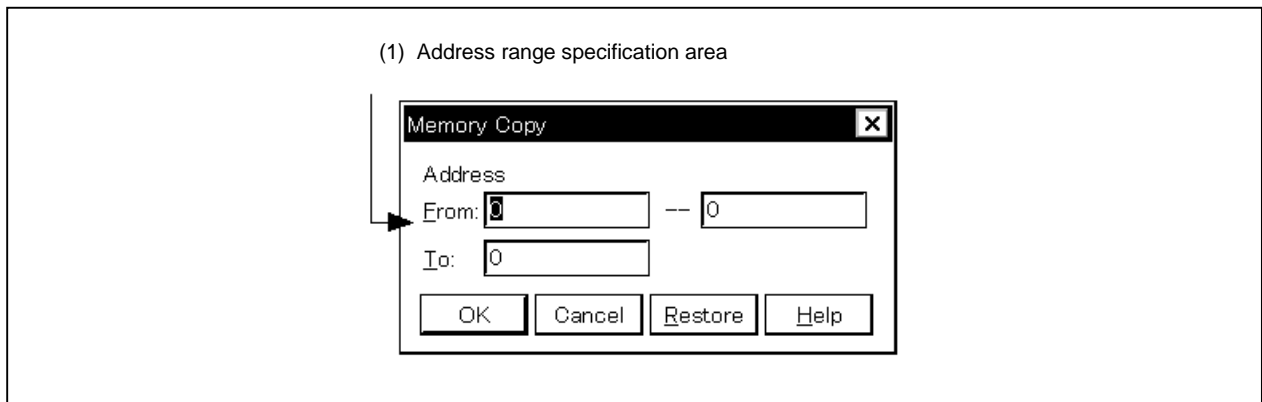
### Opening method

This dialog box can be opened in the following methods.

- In the main window  
Select [Edit] → [Memory] → [Copy...] from the menu bar.  
Press the **GRPH** + **E**, **M**, and **C** keys in that order.

### Window

**Figure 6-39. Memory Copy Dialog Box**



### Function

The Memory Copy dialog box consists of the following area.

- Address range specification area

The function of this area is explained below.

#### (1) Address range specification area

Address

From:  --

To:

This area specifies the copy source addresses and copy destination address.

A symbol or expression can also be specified for the address specification. The default radix when inputting numerical values is hexadecimal.

**From:** Specifies the address range of the copy source. Specify the copy source start address and copy source end address, in that order.

**To:** Specifies the start address of the copy destination.

#### Function buttons

OK

Copies the memory contents.

Cancel

Closes this dialog box.

This button changes to the Stop button while copying the memory.

Stop

Stops copying the memory data.

The Cancel button changes to the Stop button while copying the memory.

Restore

Returns the input data to the original value.

Help

Opens the help window.

<b>Memory Compare Dialog Box</b>	<b>Modal</b>
----------------------------------	--------------

### General

This dialog box compares the memory contents.

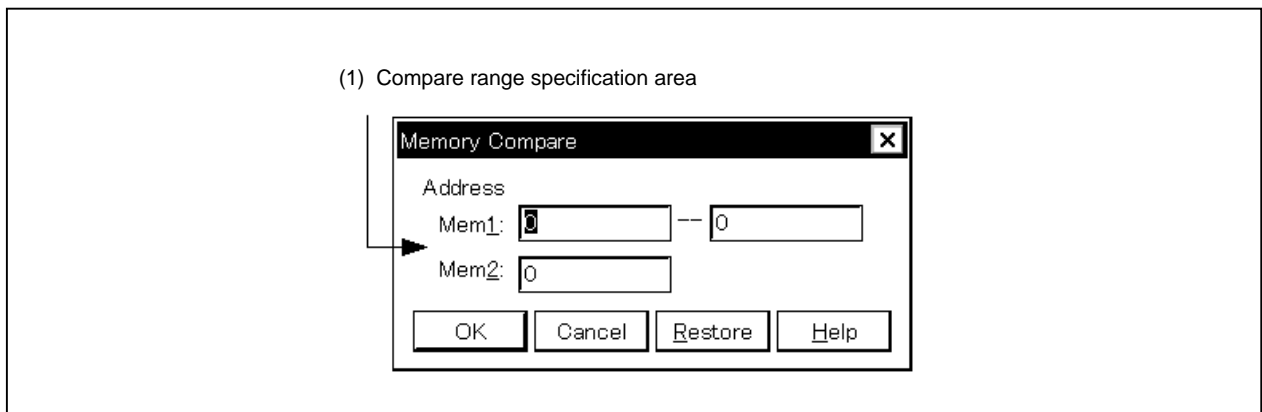
### Opening method

This dialog box can be opened in the following methods.

- In the main window  
Select [Edit] → [Memory] → [Compare...] from the menu bar.  
Press the [GRPH] + [E], [M], and [P] keys in that order.

### Window

**Figure 6-40. Memory Compare Dialog Box**



### Function

The Memory Compare dialog box consists of the following area.

- Compare range specification area

The function of this area is explained below.

#### (1) Compare range specification area

Address

Mem1:  --

Mem2:

This area specifies the compare source addresses and compare destination addresses.

A symbol or expression can also be specified for the address specification. The default radix when inputting numerical values is hexadecimal.

Mem1: Specifies the address range of the compare source. Specify the compare source start address and compare source end address, in that order.

Mem2: Specifies the start address of the compare destination.

#### Function buttons

OK

Compares the memory contents.

If no difference is found as a result of comparison, a confirmation dialog box is displayed. If a difference is found, a Memory Compare Result dialog box is opened. If no difference is found as a result of comparison, the confirmation dialog box is displayed.

By clicking the OK button in this confirmation dialog box, the Memory Compare dialog box is closed.

Cancel

Closes this dialog box.

It changes to Stop button during comparison.

Stop

Stops comparing the memory contents.

The Cancel button changes to the Stop button while comparing the memory contents.

Restore

Returns the input data to the original value.

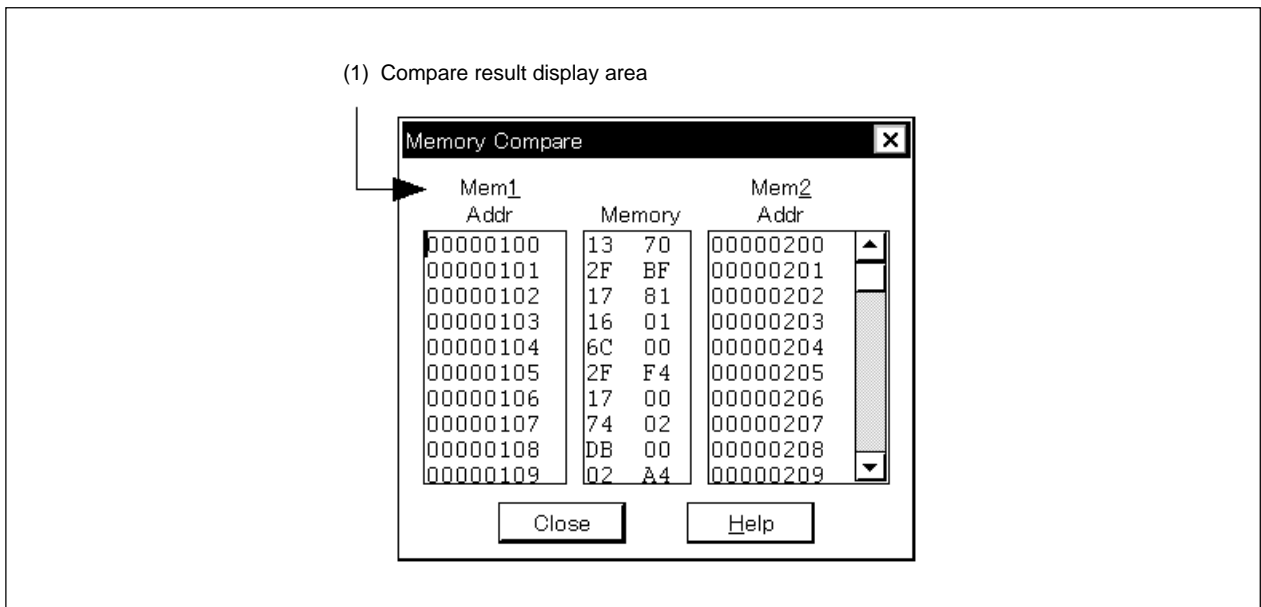
Help

Opens the help window.

**Memory Compare Result Dialog Box****Modeless****General**

This dialog box displays the result of comparing memory contents.

This dialog box is opened if a discrepancy is found as a result of comparing the memory contents in the Memory Compare Dialog box. If no discrepancy is found, this dialog box cannot be opened, but a confirmation dialog box is opened.

**Window****Figure 6-41. Memory Compare Result Dialog Box****Function**

The Memory Compare Result dialog box consists of the following area.

- Compare result display area

The function of this area is explained below.

**(1) Compare result display area**

Mem1 Addr	Memory	Mem2 Addr
00000100	13 70	00000200
00000101	2F BF	00000201
00000102	17 81	00000202
00000103	16 01	00000203
00000104	6C 00	00000204
00000105	2F F4	00000205
00000106	17 00	00000206
00000107	74 02	00000207
00000108	DB 00	00000208
00000109	02 A4	00000209

This area displays the result of comparing memory contents. Only this displays if a discrepancy is found as a result of comparing.

**Mem1 Addr:** Displays the compare source address at which a discrepancy is found as a result of comparing.

**Memory:** Displays the data which that a discrepancy is found as a result of comparing. The compare source data is displayed on the left, and the compare destination data is displayed on the right.

**Mem2 Addr:** Displays the compare destination address at which a discrepancy is found as a result of comparing.

**Function buttons**

Close

Closes this dialog box.

Help

Opens the help window.



<b>DMM Dialog Box</b>	<b>Modeless</b>
-----------------------	-----------------

### General

This dialog box sets the address and data with which to perform DMM (Dynamic Memory Modification). The contents of the memory can be overwritten in real time during user program execution.

### Opening method

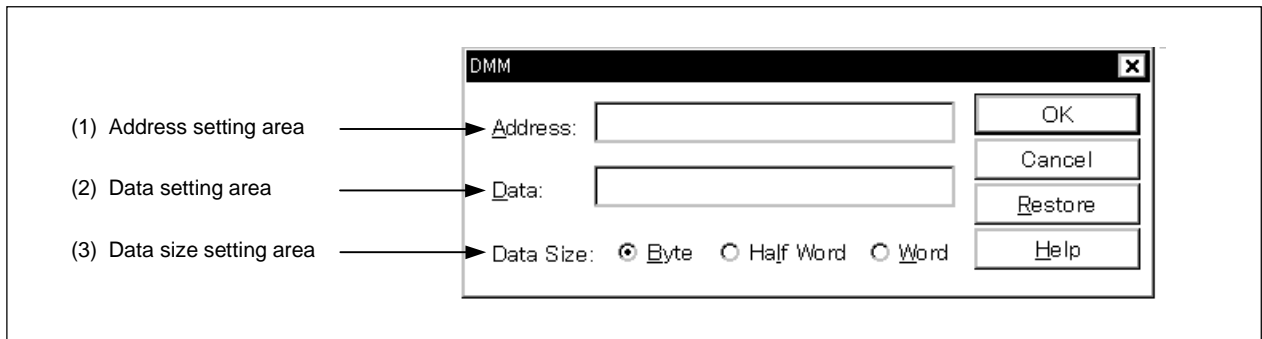
This dialog box can be opened by either of the following methods.

During user program execution

- In the main window  
Select [Edit] → [Memory] → [DMM...].  
Press the **[GRPH] + [E], [M], and [M]** keys in that order.

### Window

**Figure 6-42. DMM Dialog Box**



### Function

This dialog box overwrites the contents of the memory in real time during user program execution.

Note, however, that because this function is realized by software emulation, the user program execution is temporarily stopped and then reexecuted.

The DMM dialog box consists of the following areas.

- Address setting area
- Data setting area
- Data size setting area

The function of each area is explained below.

**(1) Address setting area**

Address:

This area specifies the address to which data is written.

It is possible to specify an address by a symbol or by expression.

The specification method is the same as in the Address Move dialog box. The default radix when inputting numerical values is hexadecimal.

**(2) Data setting area**

Data:

This area specifies data to be written to the memory address specified in the address setting area.

Only one string of binary data can be specified at the same time. The data size is specified in the data size setting area. The default radix is hexadecimal.

**(3) Data size setting area**

Data Size:  Byte  Half Word  Word

This area specifies the size of the data to be written.

Byte: Writes the data specified in the data setting area as 8-bit data.

Half Word: Writes the data specified in the data setting area as 16-bit data.

Word: Writes the data specified in the data setting area as 32-bit data.

**Function buttons**

OK

Writes the data.

Cancel

Closes the DMM dialog box.

Restore

Returns the input data to its original value.

Help

Opens the help window.

**Cautions**

Be aware of the following points when using the DMM function.

- The DMM (Dynamic Memory Modification) function is performed after stopping the user program execution temporarily.
- When data is written to the address where a software break is set, the program may not operate correctly.
- When another break trigger is generated immediately before writing and the program is stopped, that generated break trigger becomes invalid because the program execution restarts regardless of the break trigger.
- The Verification check is always invalid.
- It is impossible to write data to the area mapped to the ROM emulation memory.



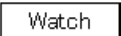
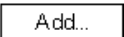
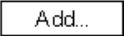
<b>Watch Window</b>	<b>MDI child window</b>
---------------------	-------------------------

**General**

This window displays the specified data and changes it.

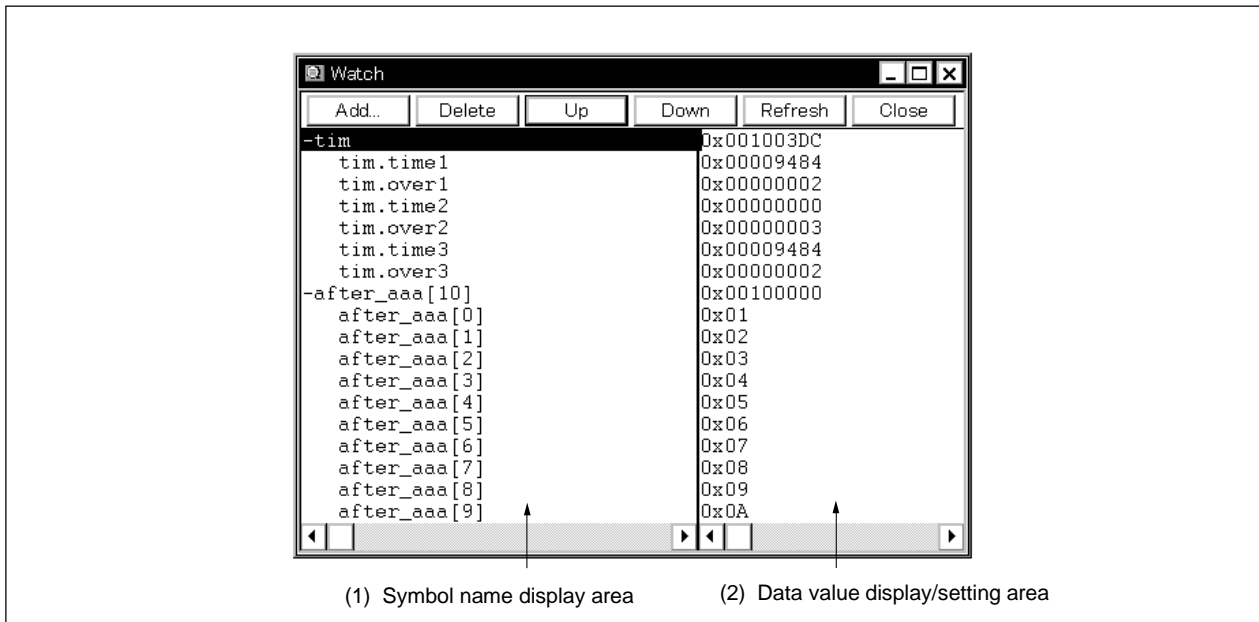
**Opening method**

This window can be opened by either of the following methods.

- In the main window  
Select [B]rowse → [W]atch from the menu bar.  
Press the [GRPH] + [B], and [W] keys in that order.  
Click the  button on the toolbar.
- In the Source window  
After selecting data, select [V]iew → [V]iew Watch from the menu bar.  
Press the [GRPH] + [V], and [I] keys in that order.  
Click  button.
- In the Assemble window  
After selecting data, select [V]iew → [V]iew Watch from the menu bar.  
Press the [GRPH] + [V], and [I] keys in that order.  
Click  button.
- In the Quick Watch dialog box  
Click  button.
- In the Add Watch dialog box  
Click  button.

## Window

Figure 6-43. Watch Window



## Function

Displays and changes data.

Adding of the data display is done by selecting a variable name or symbol name in the Source window or the Assemble window and clicking  button. Adding is also done by dragging and dropping the selected part directly to the Watch window with the mouse.

Adding can also be done by specifying a variable name or symbol name in the Quick Watch dialog box or the Add Watch dialog box and clicking .

To delete a data display, click a variable name or symbol name (it is also possible to select multiple items using the  key and the  key), then after reverse highlighting it, click  button. Alternatively, make a selection using the arrow keys, then after reverse highlighting your selection delete it by pressing the  key.

Data value update results and rewrites are reflected in the Memory window.

Also, wide area data (such as global variables or public symbols) allocated to real-time RAM monitor area<sup>Note</sup> can be displayed in real-time even while user program is executed, just as in the Memory window.

The boundary line between the symbol name display area and data value display/setting area can be moved to the left or right using the mouse.

To move the boundary, drag the mouse cursor at the point when it changes from "↖" to "↔".

**Note** Sets in the Extended Option dialog box.

The Watch window consists of the following areas.

- Symbol name display area
- Data value display/setting area

Each function is explained below.

### (1) Symbol name display area

```

-tim
tim.time1
tim.over1
tim.time2
tim.over2
tim.time3
tim.over3
-after_aaa[10]
after_aaa[0]
after_aaa[1]
after_aaa[2]
after_aaa[3]
after_aaa[4]
after_aaa[5]
after_aaa[6]
after_aaa[7]
after_aaa[8]
after_aaa[9]

```

This area displays variable names, symbol names, and the tag names of their type or structure/union. Whether the latter is displayed or not can be selected in the Debugger Option dialog box. The default is no display.

If the variable name is prefixed with “&” when registered in the Add Watch dialog box, the address to which that variable is allocated can be displayed.

A “+” is displayed in front of arrays, pointer type variables, and structures/unions. These variables are expanded and displayed as shown below by double-clicking them;

- For an array, all the elements of the variable are displayed in accordance with the type of array variable.
- For pointer type variables, the data indicated by the pointer is displayed.
- For structures/unions, all the members of the structure/union are displayed in accordance with the type of member variable.

However, if a structure/union is defined within a structure/union, the members up to the tag name or variable name in the internal structure/union are displayed.

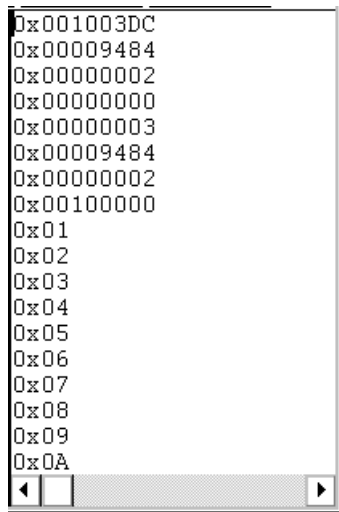
For variables which have been expanded and displayed, the “+” display is changed to a “-” display. If they are double-clicked in this state, the expansion display is deleted.

Note that variables with a “+” that are displayed inside a structure/union are expanded and displayed in the same way by double-clicking them.

To expand and display, or to delete an expansion display, select the variable with the arrow keys, then press the **Enter** key.

Data which has become invalid due to the changing of the scope or optimization compilation are displayed in gray.

Data other than global data (global variables, public symbols, etc.) allocated within the real-time RAM monitor range is displayed in gray.

**(2) Data value display/setting area**

This area is where data values are displayed and changed.

When the variable is a structure/union or an "&" is appended, the address is displayed.

When the variable is an enumeration type, the member name is displayed.

When the variable is a floating point, it is displayed as follows.

```

[+|-] inf
[+|-] nan
[+|-] integer portion e [+|-] exponent
[+|-] integer portion. Decimal portion [e [+|-] exponent]

```

The radix and size of data values can be selected on an individual variable basis using the View menu.

Note, however, that if the size in which C variables, assembler bit symbols, and registers (etc.) are displayed is fixed, the display size cannot be selected.

If [View] → [Adaptive], [View] → [Proper] has been selected, data is displayed using the radix and size specified in the Debugger Option dialog box.

**Size**

Byte	Displayed in 8 bits (default)
Halfword	Displayed in 16 bits
Word	Displayed in 32 bits

**Radix (xxxx are numerals)**

Hex	Hexadecimal (0xxxxx)
Dec	Decimal (xxxx)
Oct	Octal (0xxxx)
Bin	Binary (0bxxxx)
String	Character string

Values are updated when execution is stopped. If you desire to save values, select [File] → [Save As...] in the main window, then save the displayed file. Also, if the acquisition of data values has failed, this area will be blank.

A question mark “?” is displayed for data which has been invalid due to the changing of the scope or optimized compilation.

In addition, a “?” is displayed for data except wide area data (such as global variables or public symbols) allocated to real-time RAM monitor area.

The contents of changes are written to the target by pressing the **Enter** key. The contents prior to the change can be erased using the **ESC** key.

#### Function buttons

<b>Add..</b>	Opens the Add Watch dialog box. In the Add Watch dialog box, if the data is specified and <b>Add..</b> button is clicked, the specified data is added to the Watch window.
<b>Delete</b>	Deletes the specified data from the watch window. The <b>DEL</b> key also performs this operation.
<b>Up</b>	Moves the selected data one line up.
<b>Down</b>	Moves the selected data one line down.
<b>Refresh</b>	Updates the contents of the window with the latest data.
<b>Close</b>	Closes the Watch window.



<b>Quick Watch Dialog Box</b>	<b>Modal</b>
-------------------------------	--------------

### General

This is where the specified data is temporarily displayed and changed.

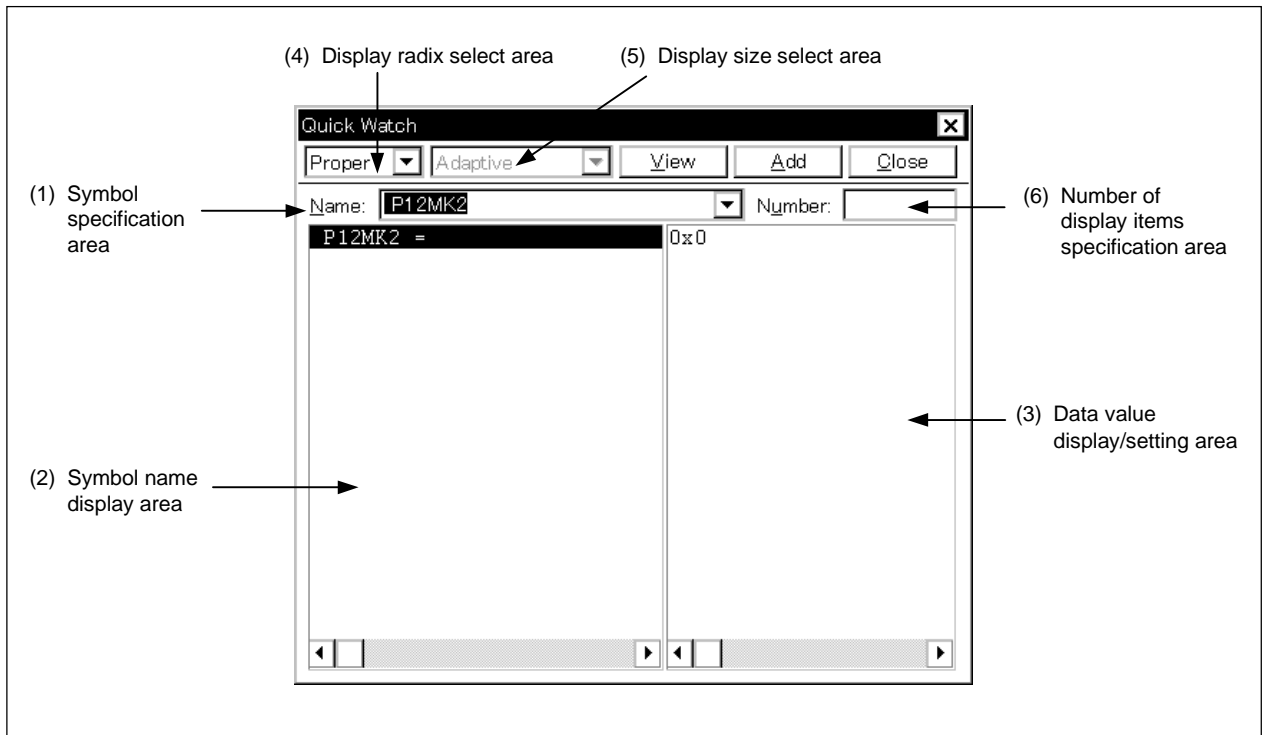
### Opening method

This dialog box can be opened by any of the following methods.

- In the main window  
Select [View] → [Quick Watch...] from the menu bar.  
Press the **GRPH** + **V**, and **Q** keys in that order.
- In the Source window  
Click the **Quick...** button in the Source window.  
Press the shortcut keys **CTRL** + **W**.
- In the Assemble window  
Click the **Quick...** button in the Assemble window.  
Press the shortcut keys **CTRL** + **W**.



### Window

**Figure 6-44. Quick Watch Dialog Box**



## Functions

In this dialog box the specified data is temporarily displayed and changed.

The boundary between the symbol name display area and the data value display/setting area can be moved to the left or right using the mouse. To move the boundary, drag the mouse cursor at the point when it changes from “” to “”.

The Quick Watch dialog box consists of the following areas.

- Symbol specification area
- Symbol name display area
- Data value display/setting area
- Display radix select area
- Display size select area
- Number of display items specification area


Each function is explained below.


### (1) Symbol specification area

Name:  

This is where the data to be displayed is specified.

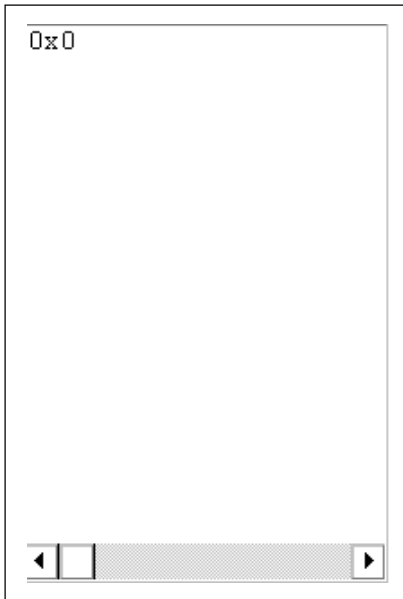
In the default setting, the string selected in the accessed window is displayed, but as necessary, it is also possible to change it by inputs from the keyboard. However, if the selected string does not exist, this area becomes blank.

By clicking the  button and displaying the input history, the contents input previously can be reused. Up to 16 items of input history can be retained.

When the contents of the symbol specification area have been changed, clicking  button causes the data specified in the symbol specification area to be displayed in the symbol name display area and data value display/setting area.

**(2) Symbol name display area**

In this area the variable names or symbol names are displayed.  
However, data which has becomes invalid due to the changes of the scope or optimization compilation are not displayed.

**(3) Data value display/setting area**

In this area the data values are displayed.  
The display contents and the way to change the contents are the same as in the Watch window.

**(4) Display radix select area**


The radix to be displayed is selected from the drop-down list.

The radix of the data name selected in the symbol name display area is changed.

Proper:	In the case of a variable, the value specified for each variable is displayed. In the case of a symbol, it is displayed in the radix set in the Debugger Option dialog box.
Hex:	Displayed in hexadecimal (0xXXXX).
Dec:	Displayed in decimal (XXX).
Oct:	Displayed in octal (OXXX).
Bin:	Displayed in binary (0bXXX).
String:	Displayed as a character string.

**(5) Display size select area**


The size to be displayed is selected from the drop-down list.

The size of the data selected in the symbol name display area is changed.

Adaptive:	In the case of a variable, the value specified for each variable is displayed. In the case of a symbol, the size set in the Debugger Option dialog box is displayed.
Byte:	Displayed in 8 bits.
Half Word:	Displayed in 16 bits.
Word:	Displayed in 32 bits.

If the display size of variables, or assembler bit, symbols or registers, etc. in the C language is fixed, the display size cannot be selected.

**(6) Number of display items specification area**


In this area the number of data to be displayed is specified.

Specify blank or a number 1 to 256. The default is blank.

If blank, the data is displayed as a simple variable. If a number 1 or greater is specified, it is displayed as an array variable. In this case, a "+" is displayed in front of the data. By double-clicking these data, all the elements of the data are expanded and displayed in accordance with the data type. "+" of the expanded and displayed data changes to "-". Double-clicking in this state cancels the expanded display.

If the display number of variables, assembler bit symbol, or registers, etc. in the C language is fixed, the display number cannot be specified.

**Function buttons****V**iew

Displays the data specified in the symbol specification area in the symbol name display area and the data value display/setting area.

**A**dd

Adds the data specified in the symbol specification area to the Watch window.

**C**lose

Closes this dialog box.

Change contents that have not actually been written to the target are erased.

<b>Add Watch Dialog Box</b>	<b>Modal</b>
-----------------------------	--------------

### General

This is where data to be displayed in the Watch window is registered.

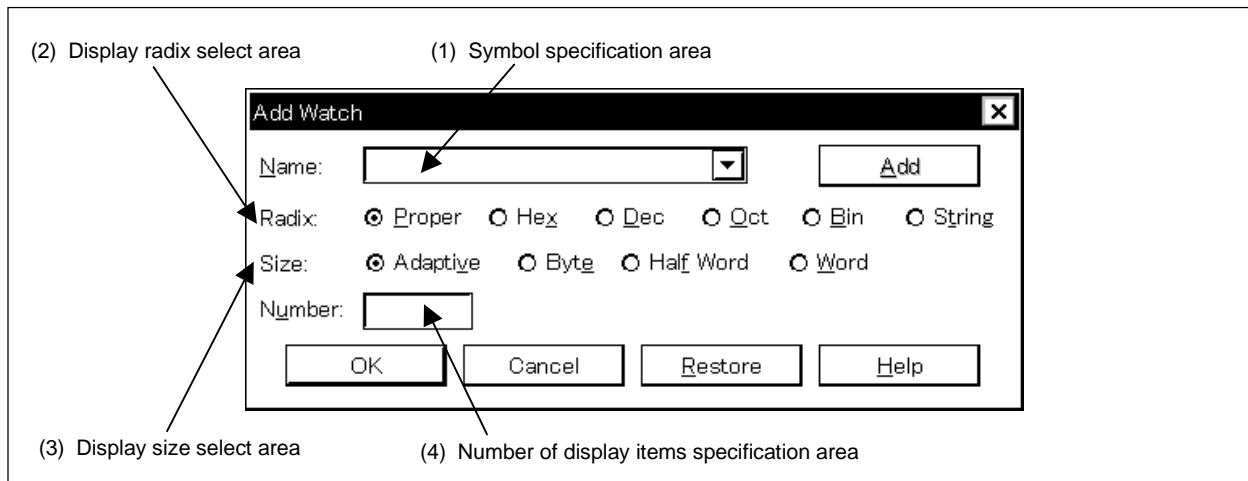
### Opening method

This dialog box can be opened by either of the following methods.

- In the main window  
Select [View] → [Add Watch...] from the menu bar.  
Press the [GRPH] + [V], and [W] keys in that order.
- In the Watch window  
Click the  button in the Watch window.

### Window

**Figure 6-45. Add Watch Dialog Box**



### Functions

The Add Watch dialog box consists of the following areas.


- Symbol specification area
- Display radix select area
- Display size select area
- Number of display items specification area

The function of each area is explained below.

**(1) Symbol specification area**

Specify the data to be viewed in the Watch window.

In the default setting, the string selected in the accessed window is displayed, but as necessary, it is also possible to change it by inputs from the keyboard. However, if the selected string does not exist, this area becomes blank.

By clicking the  button and displaying the input history, the contents input previously can be reused. Up to 16 items of input history can be retained.

The expressions that can be used for inputting data are as shown below.

- C language variable name (Variable expression: Variable name)
  - Variable expression [decimal constant values]: Array elements
  - Variable expression. Member name: Structure/union actual member
  - Variable expression → Member name: Member of structure/union that indicates a pointer
  - \*Variable expression: Value of pointer variable
  - & variable: Address which a variable was arranged
- Register name
- IOR name, IOR bit name
- Label and immediate address
- Register name.bit
- Label name.bit.bit, immediate address.bit
- Bit symbol

**(2) Display radix select area**

This area is where the display radix is selected.

The display radix can be selected from among the following (XXX is a number).

- Proper: In the case of a variable, the value specified for each variable is displayed.  
In the case of a symbol, it is displayed in the radix set in the Debugger Option dialog box.
- Hex: Displayed in hexadecimal (0xXXX).
- Dec: Displayed in decimal (XXX).
- Oct: Displayed in octal (0XXX).
- Bin: Displayed in binary (0bXXX).
- String: Displayed as a character string.

**(3) Display size select area**

Size:	<input checked="" type="radio"/> Adaptive	<input type="radio"/> Byte	<input type="radio"/> Half Word	<input type="radio"/> Word
-------	---	----------------------------	---------------------------------	----------------------------

This area is where the display size is selected.

The display size can be selected from among the following.

**Adaptive:** In the case of a variable, the value specified for each variable is displayed.  
In the case of a symbol, the size set in the Debugger Option dialog box is displayed.

**Byte:** Displayed in 8 bits.

**Half Word:** Displayed in 16 bits.

**Word:** Displayed in 32 bits.

If the display size of variables, or assembler bit symbols or registers, etc. in the C language is fixed, the specified size is displayed in the Watch window regardless of the specified display size.

**(4) Number of display items specification area**

Number:	<input type="text"/>
---------	----------------------

In this area the number of data to be displayed is specified.

Specify blank or a number 1 or greater (1 to 256). The default is blank.

If blank, the data is displayed as a simple variable. If a number 1 or greater is specified, it is displayed as an array variable. In this case, a "+" is displayed in front of the data. By double-clicking these data, all the elements of the data are expanded and displayed in accordance with the data type. "+" of the expanded and displayed data changes to "-". Double-clicking in this state releases the expanded display.

If the display number of various assembler bit symbol, or registers, etc. in the C language is fixed, the display number cannot be specified.

**Function buttons**

Add

Adds the specified data to the Watch window. Does not close the Add Watch dialog box.

OK

After adding the specified data to the Watch window, it closes the Add Watch dialog box.

Restore

Returns the input data to its source.

Cancel

Closes the Add Watch dialog box.

Help

Opens the help window.




<b>Register Window</b>	<b>MDI child window</b>
------------------------	-------------------------

### General

This window displays and changes the contents of the registers (program registers and system registers).

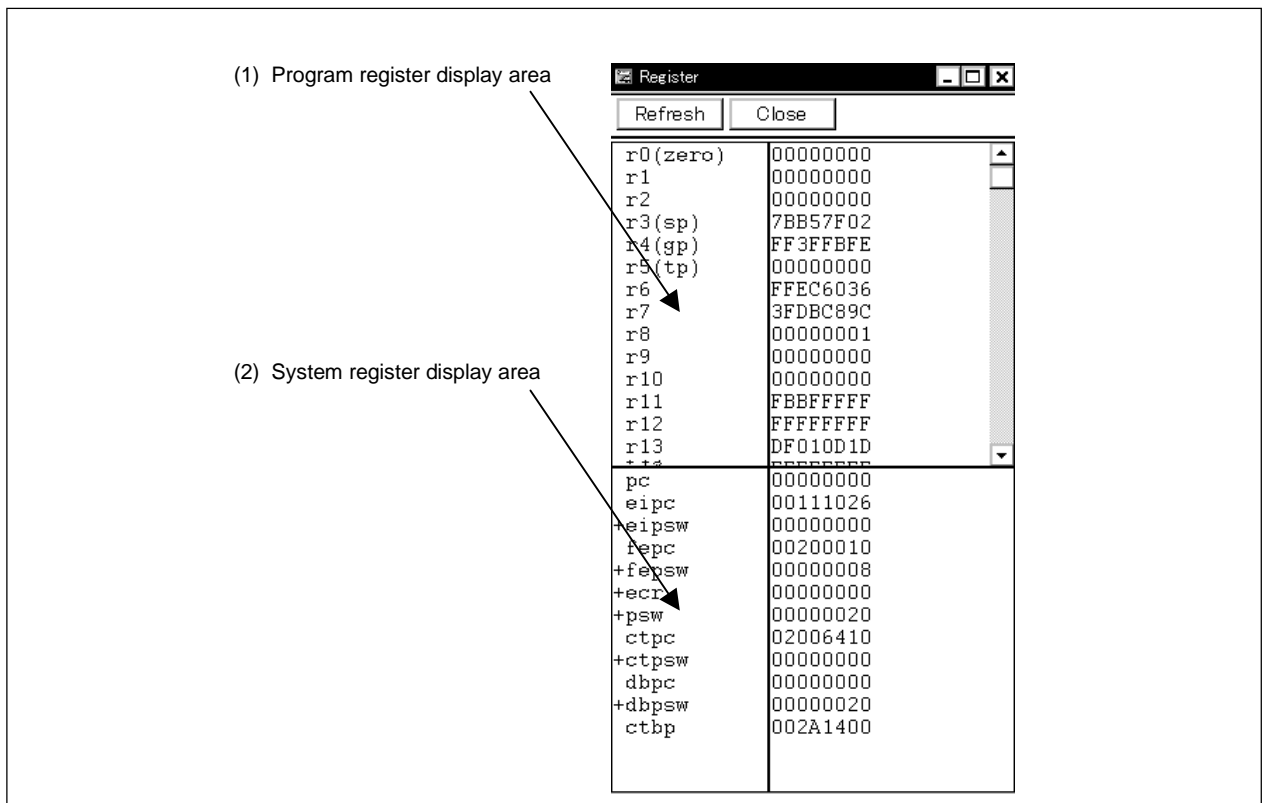
### Opening method

This window can be opened in the following methods.

- In the main window
  - Select [B]rowse] → [R]egister] from the menu bar.
  - Press the [GRPH] + [B], and [R] keys in that order.
  - Click the  button on the toolbar.

### Window

**Figure 6-46. Register Window**





### Function

This window displays and changes the contents of registers (program registers and system registers).

The register to be displayed can be selected in the Register Select dialog box.

The Register window consists of the following areas.

- Program register display area
- System register display area

The boundary line between the program register display area and system register display area can be moved up or down using the mouse. To move the boundary, drag the mouse cursor at the point when it changes from “” to “”.

The function of each area is explained below.

### (1) Program register display area

r0(zero)	00000000	▲
r1	00000000	
r2	00000000	
r3(sp)	7BB57F02	
r4(gp)	FF3FFBFE	
r5(tp)	00000000	
r6	FFEC6036	
r7	3FDBC89C	
r8	00000001	
r9	00000000	
r10	00000000	
r11	FBBFFFFFF	
r12	FFFFFFFF	
r13	DF010D1D	
+ r14	FFFFFFFF	▼

This area displays or changes the contents of the program registers. The contents can be written to the target memory by pressing the **Enter** key.

The contents prior to the change can be erased using the **ESC** key.

In addition, this area has a jump function that jumps to the Source window, Assemble window, or Memory window by using a selected program register value as a jump pointer. The jump destination window displays the jump pointer at the beginning.

This function is executed by performing the following operations in the sequential order.

- <1> Select the displayed register value.
- <2> Perform either of the following operations in the main window.
  - If the jump destination window is the Source window.
    - Select **[Jump]** → **[Source]** from the menu bar.
    - Press **GRPH** + **J**, and **S** keys in that order.
    - Press the shortcut keys **CTRL** + **U**.
  - If the jump destination window is the Assemble window.
    - Select **[Jump]** → **[Assemble]** from the menu bar.
    - Press **GRPH** + **J**, and **A** keys in that order.
    - Press the shortcut keys **CTRL** + **D**.
  - If the jump destination window is the Memory window.
    - Select **[Jump]** → **[Memory]** from the menu bar.
    - Press **GRPH** + **J**, and **E** keys in that order.
    - Press the shortcut keys **CTRL** + **M**.

**(2) System register display area**

pc	00000000
eipc	00111026
+eipsw	00000000
fepc	00200010
+fepsw	00000008
+ecr	00000000
+psw	00000020
ctpc	02006410
+ctpsw	00000000
dbpc	00000000
+dbpsw	00000020
ctbp	002A1400

This area displays or changes the contents of the system registers. The contents can be written to the target memory by pressing the **Enter** key.

The contents prior to the change can be erased using the **ESC** key.

The flag names and flag values of the register to which “+” is prefixed are displayed by double-clicking the register name. “+” then changes to “-”. Switching can also be done by selecting a register name using the arrow keys then pressing the **Enter** key.

In addition, this area has a jump function as a jump pointer. Operation of the jump function is the same as in the case of the program register.

**Function buttons****Refresh**

Updates the contents of the window to the latest data.

**Close**

Closes this window.

<b>Register Select Dialog Box</b>	<b>Modal</b>
-----------------------------------	--------------

### General

This dialog box selects the registers that are not displayed in the Register Window.

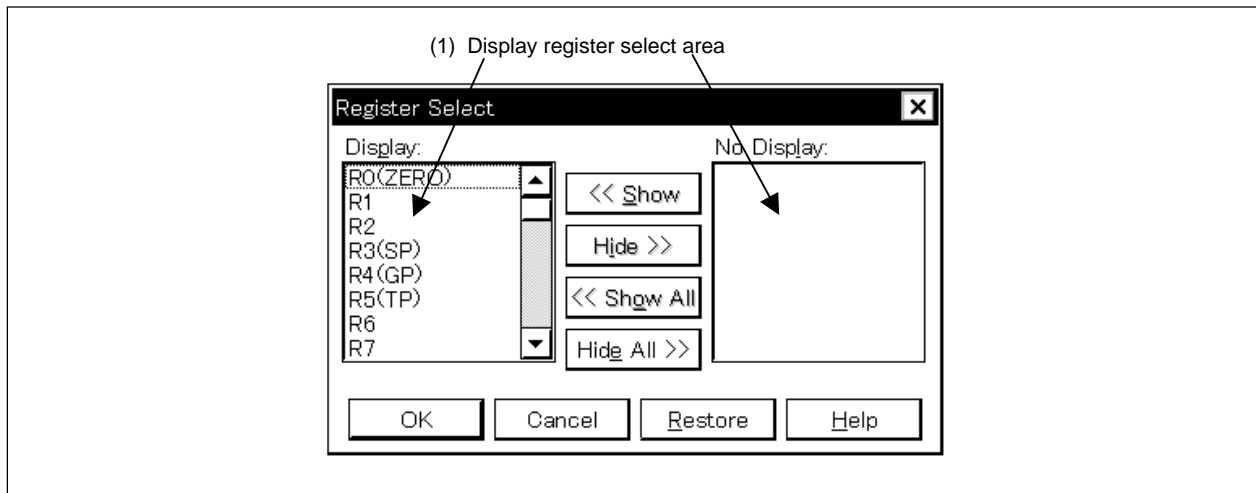
### Opening method

This dialog box can be opened in the following methods when the current window is the Register Window.

- In the main window  
Select [View] → [Select...] from the menu bar.  
Press the [GRPH] + [V], and [E] keys in that order.

### Window

**Figure 6-47. Register Select Dialog Box**

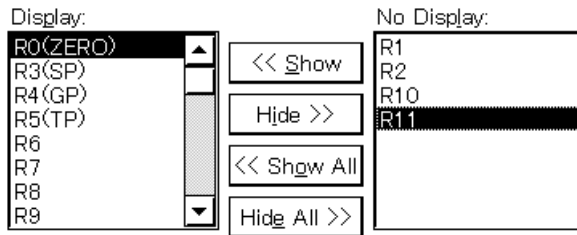


### Function

The Register Select dialog box consists of the following area.

- Display register select area

The function of this area is explained below.

**(1) Display register select area**

Select the registers that are to be displayed on the Register Window, and the registers that are not to be displayed.

The registers displayed in the Register Window are shown on the [Display:] list, and the registers not displayed are shown on the [No Display:] list. The registers displayed or not displayed in the Register Window can be changed by selecting a register name from these lists and by clicking the  or  button. If all the registers are to be displayed, click the  button. If all the registers are to be hide, click the  button. The function of each button is explained below.

<input type="button" value="&lt;&lt; Show"/>	Moves the register selected in the [No Display:] area to the cursor position in the [Display:] list area.
<input type="button" value="Hide &gt;&gt;"/>	Moves the register selected in the [Display:] list area to the [No Display:] area.
<input type="button" value="&lt;&lt; Show All"/>	Moves all the registers to the cursor position in the [Display:] list area.
<input type="button" value="Hide All &gt;&gt;"/>	Moves all the registers to [No Display:] area.

Two or more peripheral I/O registers and I/O ports can be selected by clicking the mouse while pressing the  or  key when selecting a register from the [Display:] list or [No Display:] list.

**Function buttons**

<input type="button" value="OK"/>	Reflects the result of selection and specification of display order in the Register window and closes the Register Select dialog box.
<input type="button" value="Cancel"/>	Cancels the change and restores the original setting and closes this dialog box.
<input type="button" value="Restore"/>	Cancels the change and restores the original setting.
<input type="button" value="Help"/>	Opens the help window.


<b>IOR Window</b>	<b>MDI child window</b>
-------------------	-------------------------

### General

This window displays and changes the contents of the peripheral I/O registers. Also, the entered I/O port registered in the Add I/O Port dialog box can be viewed and changed.

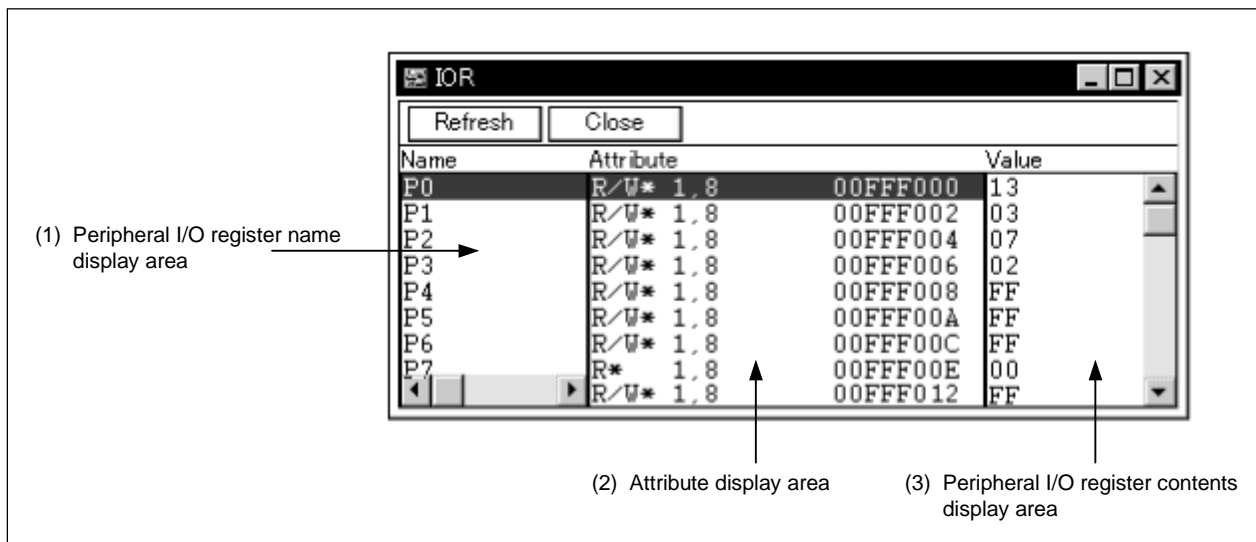
### Opening method

This window can be opened in the following methods.

- In the main window  
Select [Browse] → [I/O Register] from the menu bar.  
Press the [GRPH] + [B], and [G] keys in that order.  
Click the  button on the toolbar.

### Window

Figure 6-48. IOR Window



### Function

This window displays or changes the contents of peripheral I/O registers or I/O ports registered in the Add I/O Port dialog box.

The IOR window consists of the following areas.

- Peripheral I/O register name display area
- Attribute display area
- Peripheral I/O register contents display area

The function of each area is explained below.

**(1) Peripheral I/O register name display area**

Name
P0
P1
P2
P3
P4
P5
P6
P7

This area displays the names of the peripheral I/O registers and I/O port name.

For types corresponding to programmable I/O register for which programmable I/O area is set in the Configuration dialog box, programmable I/O register is also displayed.

The bit name and bit values of the peripheral I/O register to which “+” is prefixed are displayed by double-clicking the peripheral I/O “+” then changes to “-” register name.

Switching can also be done by selecting a peripheral I/O name using the arrow keys then pressing the  key.

If the I/O port address is an undefined value, the I/O port name is displayed in gray.

**(2) Attribute display area**

Attribute	
R/W 1,8	00FFF000
R/W 1,8	00FFF002
R/W 1,8	00FFF004
R/W 1,8	00FFF006
R/W 1,8	00FFF008
R/W 1,8	00FFF00A
R/W 1,8	00FFF00C
R 1,8	00FFF00E

This area displays read/write attributes, access types, and absolute addresses of each peripheral I/O register and I/O port.

Selection of whether to display the attribute display area can be made by selecting  →  from the menu bar.

The following read/write attributes are displayed.

R(\*): Read-only peripheral I/O register or I/O port  
 W(\*): Write-only peripheral I/O register or I/O port  
 R/W(\*): Read/write peripheral I/O register or I/O port

Peripheral I/O registers with an asterisk (\*) affixed to their attribute indicate registers that read the data copy via the emulation register when reading.

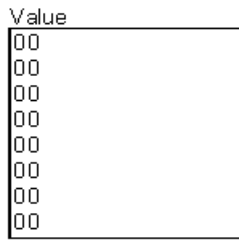
To read data directly from these registers, select  →  from the menu bar.

This function is not supported in all devices.

The following access types are displayed.

1: Peripheral I/O register that can be accessed in bit units  
 8: Peripheral I/O register or I/O port that can be accessed in byte units  
 16: Peripheral I/O register or I/O port that can be accessed in half word units  
 32: Peripheral I/O register or I/O port that can be accessed in word units

### (3) Peripheral I/O register contents display area



This area displays and changes the contents of the peripheral I/O registers or I/O ports.  
The contents are displayed as follows, depending on the attributes of the registers or I/O ports.

- Read-only peripheral I/O register or I/O port:      Displayed in black
- Write-only peripheral I/O register or I/O port:      “--” is displayed
- Read/write peripheral I/O register or I/O port:      Displayed in black
- Peripheral I/O register or I/O port whose  
value changes when read:      “\*\*\*” is displayed

The contents can be written to the target by pressing the **Enter** key.

The contents prior to the changes can be erased using the **ESC** key.

The value of a read-protected peripheral I/O register can be read by executing **[View] → [Compulsion Read]** on the menu bar.

#### Function buttons

**Refresh**

Updates the contents of the window to the latest data.

**Close**

Closes this window.



<b>IOR Select Dialog Box</b>	<b>Modal</b>
------------------------------	--------------

### General

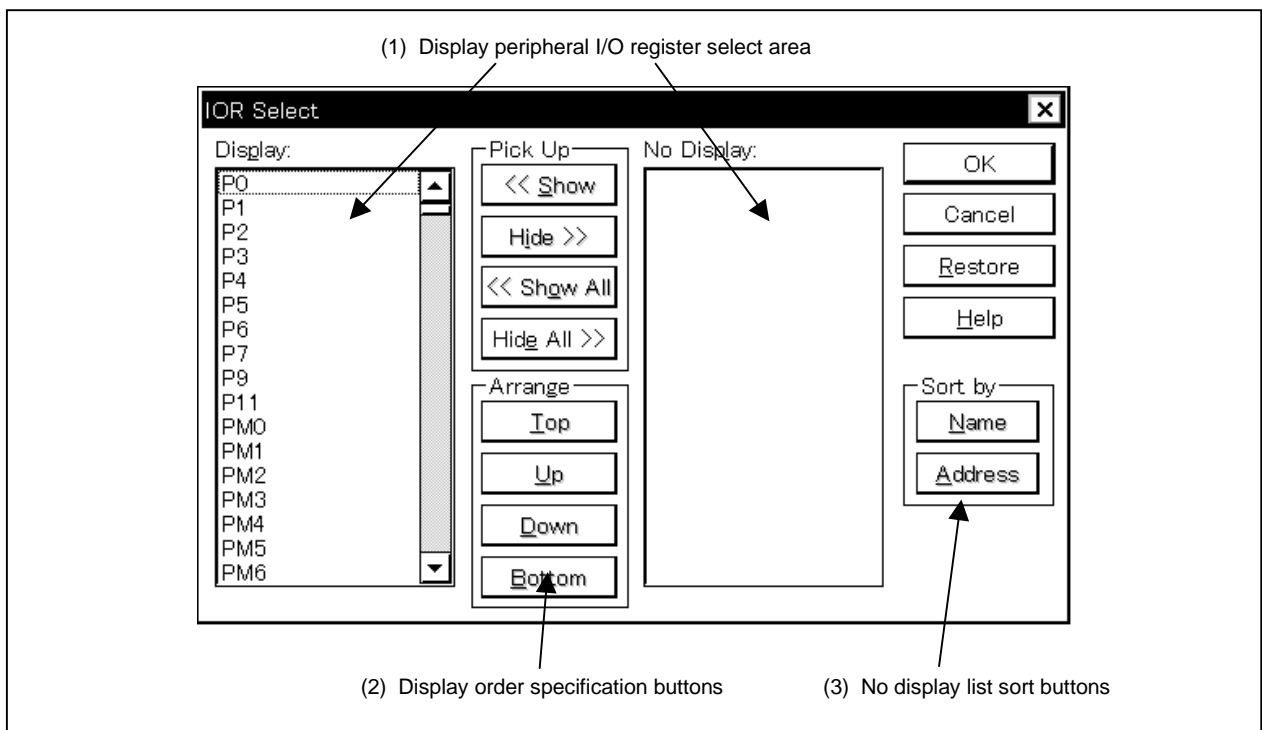
This dialog box is used to select a peripheral I/O register or I/O port that is not being displayed in the IOR window. The display order is also set here.

### Opening method

This window can be opened in the following methods when the current window is the IOR window.

- In the main window  
Select [View] → [Select...] from the menu bar.  
Press the **[GRPH]** + **[V]**, and **[E]** keys in that order.

**Figure 6-49. IOR Select Dialog Box**

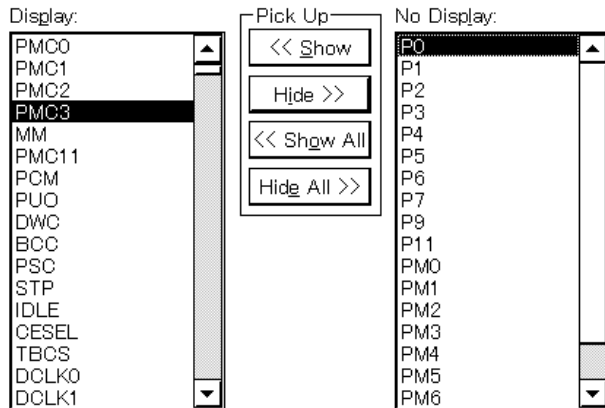


### Functions

The IOR Select dialog box consists of the following areas.

- Display peripheral I/O register select area
- Display order specification buttons
- No display list sort buttons

The function of each area is explained below.

**(1) Display peripheral I/O register select area**

Select the peripheral I/O registers and I/O ports that are to be displayed on the IOR Window, and the peripheral I/O registers and I/O ports that are not to be displayed.

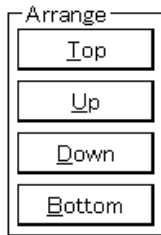
The peripheral I/O registers and I/O ports displayed in the IOR Window are shown on the [Display:] list, and the peripheral I/O registers and I/O ports not displayed are shown on the [No Display:] list.

The peripheral I/O registers and I/O ports displayed or not displayed in the IOR Window can be changed by selecting a peripheral I/O register name or I/O port name from these lists and by clicking the  or  button. If all the registers and I/O ports are to be displayed, click the  button. If all the registers and I/O ports are to be hide, click the  button.

The function of each button is explained below.

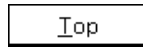
<input type="button" value="&lt;&lt; Show"/>	Moves the peripheral I/O registers and I/O ports selected in the [No Display:] area to the cursor position in the [Display:] list area.
<input type="button" value="Hide &gt;&gt;"/>	Moves the peripheral I/O registers and I/O ports selected in the [Display:] list area to the [No Display:] area.
<input type="button" value="&lt;&lt; Show All"/>	Moves all the peripheral I/O registers and I/O ports to the cursor position in the [Display:] list area.
<input type="button" value="Hide All &gt;&gt;"/>	Moves all the peripheral I/O registers and I/O ports to [No Display:] area.

Two or more peripheral I/O registers and I/O ports can be selected by clicking the mouse while holding down the  key or  key when selecting a peripheral I/O register or I/O port from the [Display:] list or [No Display:] list.

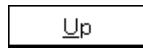
**(2) Display order specification buttons**

These set the order in which the [Display:] list is displayed. The peripheral I/O register or I/O port selected in the [Display:] list moves to the top line of the list when  button is clicked and moves to the bottom line of the list when  button is clicked. It also moves one line up when  button is clicked and one line down when  button is clicked.

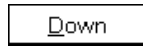
The functions of each button are as shown below.



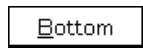
Moves the peripheral I/O register or I/O port selected in the [Display:] list to the top line of the list.



Moves the peripheral I/O register or I/O port selected in the [Display:] list one line up.



Moves the peripheral I/O register or I/O port selected in the [Display:] list one line down.

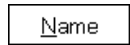


Moves the peripheral I/O register or I/O port selected in the [Display:] list to the bottom line of the list.

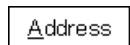
**(3) No display list sort buttons**

Select the order in which the a [No Display:] list is displayed. When  button is clicked, the items are displayed in alphabetical order, and when  button is clicked, the items are displayed in address order.

The functions of each button are as shown below.



Displays the [No Display:] list in alphabetical order.



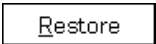
Displays the [No Display:] list in address order.

**Function buttons**OK

After the selection results and the specified display order are reflected in the IOR window, clicking this button closes the IOR Select dialog box.

Cancel

Cancels the change and closes this dialog box.

Restore

Cancels the change and restores the original states.

Help

Opens the help window.

<b>Add I/O Port Dialog Box</b>	<b>Modal</b>
--------------------------------	--------------

### General

This is where an I/O port to be added to the IOR window is registered.

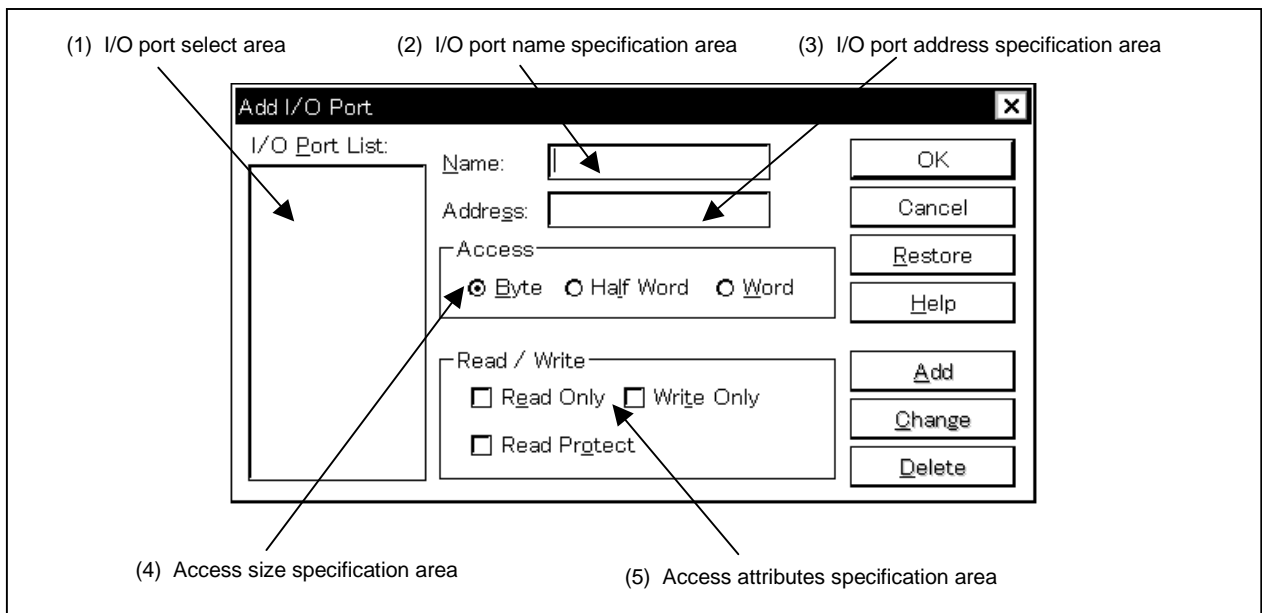
### Opening method

This dialog box can be opened in either of the following methods.

- In the main window  
Select [O]ption → [Add I/O Port] from the menu bar.  
Press [GRPH] + [O], and [P] keys in that order.

### Window

Figure 6-50. Add I/O Port Dialog Box

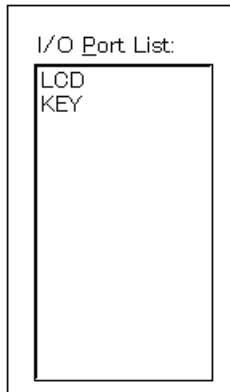


### Functions

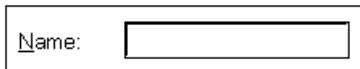
The Add I/O Port dialog box consists of the following areas.

- I/O port select area
- I/O port name specification area
- I/O port address specification area
- Access size specification area
- Access attributes specification area

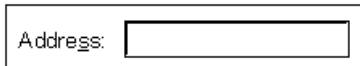
The functions of each area are explained below.

**(1) I/O port select area**

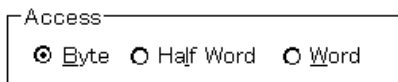
This area is where a list of the currently entered I/O ports is displayed. If you enter a new I/O port, it is added to this list. You can also select an I/O port that has already been entered and change or delete it.

**(2) I/O port name specification area**

This is where the name of the I/O port to be added is specified. An I/O port name with up to 15 characters can be specified.

**(3) I/O port address specification area**

This is where the I/O port address to be added is specified. Addresses which can be specified are those in the Target area or IOR area. Address specification can be done by specifying a symbol or expression. The specification method is the same as in the case of the Address Move dialog box. The default radix when inputting numerical values is hexadecimal.

**(4) Access size specification area**

This is where the access size of the added I/O port is specified.

- Byte      8 bits (default)
- Half Word    16 bits
- Word        32 bits

**(5) Access attributes specification area**

Read / Write	
<input type="checkbox"/> Read Only	<input type="checkbox"/> Write Only
<input type="checkbox"/> Read Protect	

This is where the access attributes of the added I/O port are specified.

- |                                       |                            |
|---------------------------------------|----------------------------|
| <input type="checkbox"/> Read Only    | Sets exclusive read only.  |
| <input type="checkbox"/> Write Only   | Sets exclusive write only. |
| <input type="checkbox"/> Read Protect | Sets read protect.         |

In the default (all unchecked state), read and write are both enabled.

**Function buttons**

- |  |   |
|--|---|
| <input type="button" value="OK"/>      | After reflecting the add results in the IOR window, this button closes the Add I/O Port dialog box. |
| <input type="button" value="Cancel"/>  | Cancels the change and closes this dialog box.  |
| <input type="button" value="Restore"/> | Cancels the change and restores the original states.  |
| <input type="button" value="Help"/>    | Opens the help window.  |
| <input type="button" value="Add"/>     | Adds the I/O port with the specified address.   |
| <input type="button" value="Change"/>  | Changes the settings for the selected I/O port.   |
| <input type="button" value="Delete"/>  | Deletes the selected I/O port.  |


<b>Local Variable Window</b>	<b>MDI child window</b>
------------------------------	-------------------------

### General

This window displays or changes the local variable in the current function.

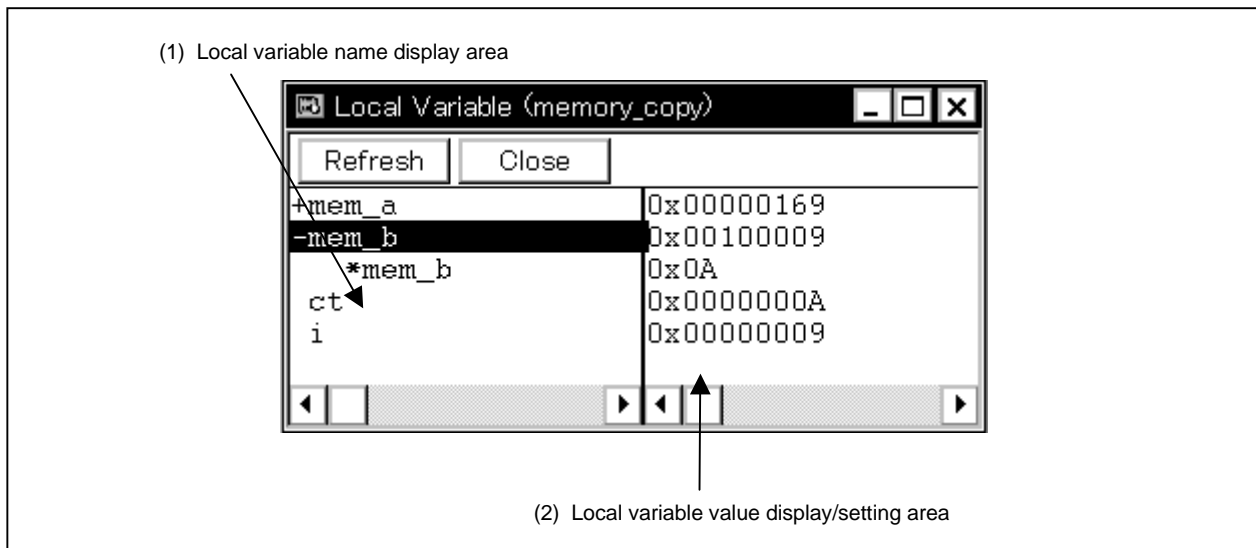
### Opening method

This window can be opened in the following methods.

- In the main window  
 Select [B]rowse] → [L]ocal Variable] from the menu bar.  
 Press the [GRPH] + [B], and [L] keys in that order.  
 Click the  button on the toolbar.

### Window

**Figure 6-51. Local Variable Window**



### Function

This window displays or changes the local variable.

This window automatically displays the local variable in the current function. Variables cannot be add or deleted.

The boundary line between the local variable name display area and local variable value display/setting area can be moved to the left or right by the mouse. The boundary can be moved by dragging when the mouse cursor changes from "↔" to "⇄".

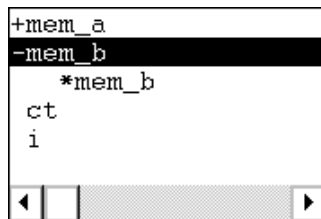


The Local Variable window consists of the following areas.

- Local variable name display area
- Local variable value display/setting area

Each function is explained below.

#### (1) Local variable name display area

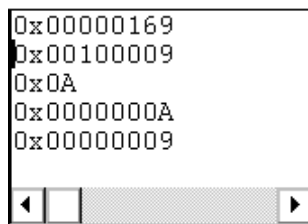


This area displays local variable names.

The type and variable name are shown. These are displayed in the same way as in the Watch window. Also, it is possible to have an expansion display like in the Watch window. However, variable names that have become invalid due to optimization compilation, etc., are not displayed.

The auto variable, internal static variable and the register variable can be displayed.

#### (2) Local variable value display/setting area



This area displays/changes local variable values.

The variable value is displayed in the same way as in the Watch window.

Change contents are written to the target by pressing the **Enter** key. The contents prior to the change can be erased using the **ESC** key.

#### Function buttons

- |         |  |
|---------|--|
| Refresh | Updates the contents of the window to the latest data. |
| Close   | Closes this window.                                    |


<b>Stack Window</b>	<b>MDI child window</b>
---------------------	-------------------------

### General

This window displays and changes the current stack contents of the user program.

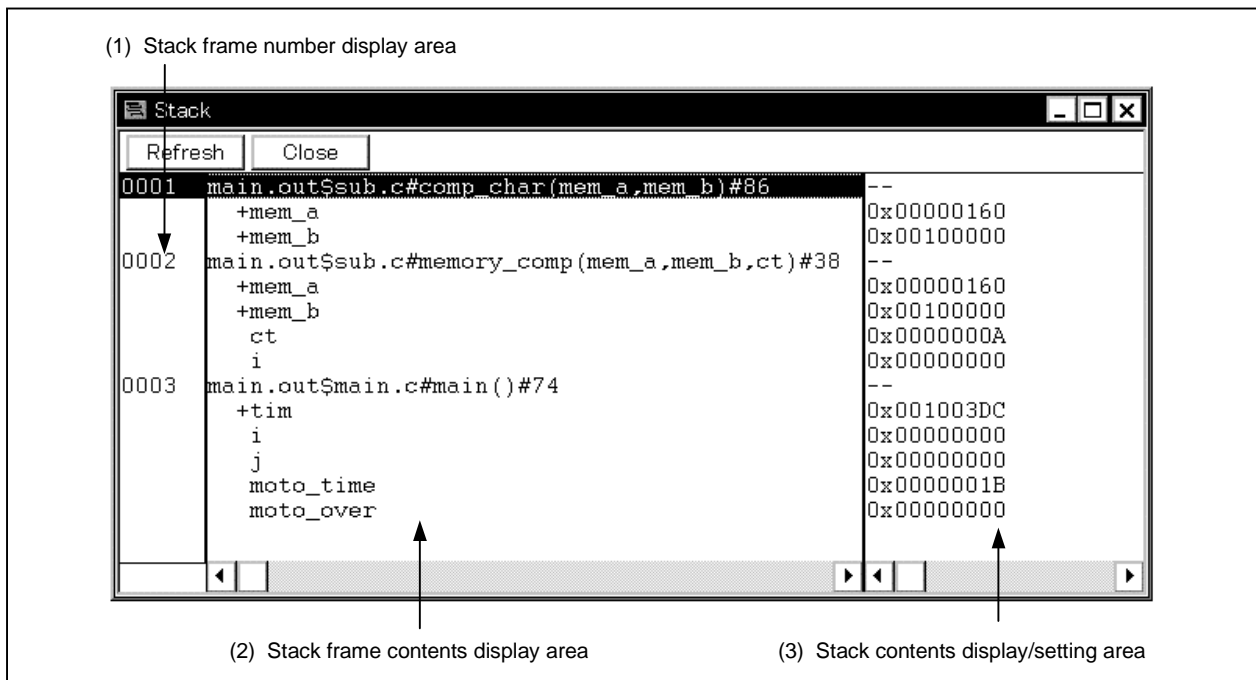
### Opening method

This window can be opened in the following methods.

- In the main window  
 Select [B]rowse] → [Stack Trace] from the menu bar.  
 Press the [GRPH] + [B], and [K] keys in that order.  
 Click the  button on the toolbar.

### Window



Figure 6-52. Stack Window



### Function

This is where the stack contents are displayed and changed.

If the full traced contents cannot be displayed, a vertical scroll bar is displayed so that the contents can be scrolled.

The boundary between the stack frame contents display area and the stack contents display/setting area can be moved to the left or right by the mouse. To move the boundary, drag the mouse cursor at the point when it changes form  to .

This window consists of the following areas.

- Stack frame number display area
- Stack frame contents display area
- Stack contents display/setting area

The function of each area is explained below.

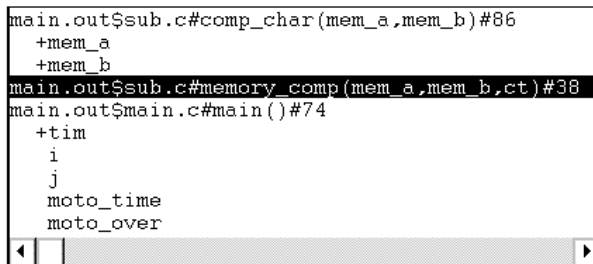
### (1) Stack frame number display area



This area displays the stack frame numbers.

The stack frame number is a natural number starting from 1, and the shallower the stack nest is, the larger this number becomes. A stack number that is 1 larger than a certain function is the calling function of that function.

### (2) Stack frame contents display area

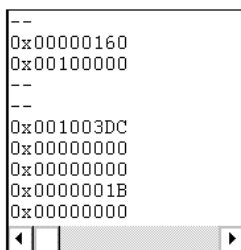


This area is where the stack frame contents are displayed.

The function or local variable names are displayed.

If the stack contents are functions, they is displayed by [Program name \$ File name # Function name (Index list) # Line no.]. The delimiter between the program name and the file name is a "\$" and the delimiter between the file name and the function name is a "#". The "#" is also used as the delimiter between the function name and the line no. If there are local variables in a function, the local variables are displayed on the following and subsequent lines.

If the contents of the stack are local variables, the type and variable name is displayed. They are displayed in the same way as in the Watch window. Note that internal static variables and register variables are not displayed.

**(3) Stack contents display/setting area**

This area displays and changes the contents of the stack. “-” or local variable values are displayed. If the stack contents are functions, a “-” is displayed and the contents cannot be changed.

If the stack contents are local variables, the variable values are displayed. Variable values are displayed in the same way as in the Watch window. It is possible to change variable values.

Changed contents are written to the target by pressing the **Enter** key. The contents prior to the change can be erased using the **ESC** key.

Other than displaying the stack contents, this area has the following functions.

**(a) Jump function**

Jumps to the Source window, Assemble window, or Memory window by using the function indicated by the stack frame number at the cursor position as a jump pointer. The jump destination window displays the jump pointer at the beginning.

This function is executed by performing the following operations in the sequential order.

- <1> Position the cursor in the stack contents display/setting area.
- <2> Perform one of the following operations on the main window.

- If the jump destination is the Source window
  - Select **[Jump]** → **[SourceText]** from the menu bar.
  - Press the **GRPH** + **J**, and **S** keys in that order.
  - Press shortcut keys **CTRL** + **U**.
- If the jump destination is the Assemble window
  - Select **[Jump]** → **[Assemble]** from the menu bar.
  - Press the **GRPH** + **J**, and **A** keys in that order.
  - Press shortcut keys **CTRL** + **D**.
- If the jump destination is the Memory window
  - Select **[Jump]** → **[Memory]** from the menu bar.
  - Press the **GRPH** + **J**, and **E** keys in that order.
  - Press the shortcut keys **CTRL** + **M**.

Details of the jump destination are as follows.

**Table 6-22. Details of Jump Function**

Type of Function	Description of Operation
In the case of the current function	<ul style="list-style-type: none"> <li>• If the jump destination is the Source window, it jumps to the current PC line.</li> </ul>
	<ul style="list-style-type: none"> <li>• If the jump destination is other than the Source window, it jumps to the current PC address.</li> </ul>
In the case of a function other than the current function	<ul style="list-style-type: none"> <li>• If the jump destination is the Source window, it jumps to the line where the function nested is called.</li> </ul>
	<ul style="list-style-type: none"> <li>• If the jump destination is other than the Source window, it jumps to the next address of the instruction that is calling the function nested.</li> </ul>

#### Function buttons

Refresh

Updates the contents of the window to the latest data.

Close

Closes this window.

#### Cautions

- (1) If the stack trace display function has a function that does not generate a stack frame, it may not be displayed correctly.
- (2) [ERROR] may be displayed during function prologue processing or epilogue processing.

<b>Trace View Window</b>	<b>MDI child window</b>
--------------------------	-------------------------


### General

This window displays the trace results.

This window is valid only for CPU cores with the TRCU incorporated.

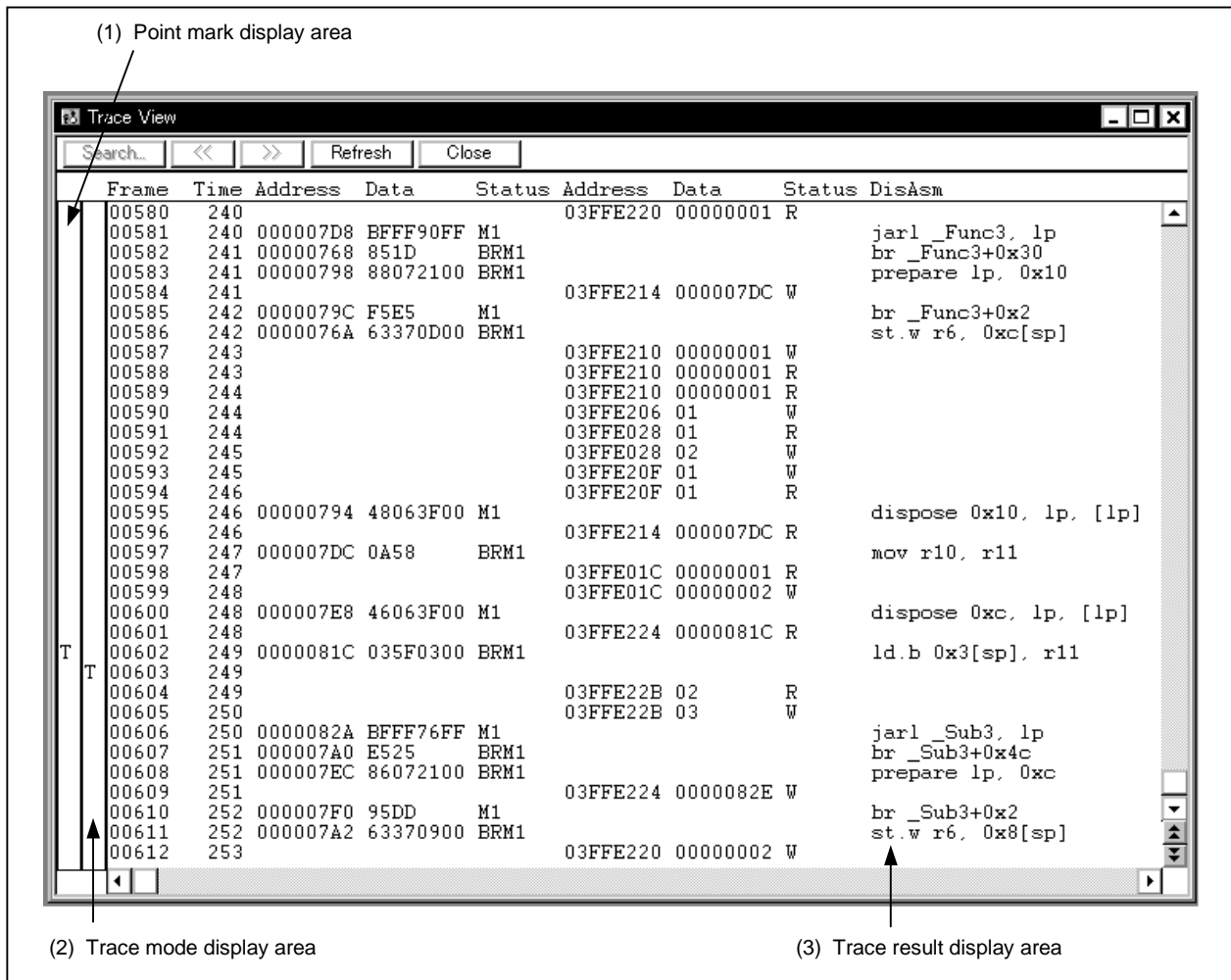
### Opening method

This window can be opened in the following methods.

- In the main window
  - Select [Browse] → [Trace] from the menu bar.
  - Press the [GRPH] + [B], and [T] keys in that order.
  - Click the  button on the toolbar.



### Window

**Figure 6-53. Trace View Window**



## Functions

This is where the trace results are displayed.

The vertical scroll bar is displayed at all times and the contents can be scrolled. It is also possible to move up one page or down one page using the  button or  button.

The Trace View window can be opened only in the active state and the window's display is updated in a break or during step execution.

The tracer has a ring structure. For that reason, if data exceeding a valid frame is written, whether to overwrite the oldest data or stop the tracer at that point can be selected by selecting [Run] → [Tracer Control] Mode]. Also, in the display, the frames are displayed in frame order with the oldest data as frame 0.

To clear the trace contents, select [Option] → [Trace] → [Clear].

A tracer is made valid by selecting [Option] → [Tracer ON].

Note that, operation with respect to a tracer during program execution is performed by selecting [Run] → [Tracer Start/Close].

The Trace View window consists of the following areas.

- Point mark display area
- Trace mode display area
- Trace result display area

The function of each area is explained below.

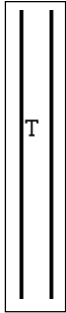
### (1) Point mark display area



This is where the setting states of various events are displayed. If an execution event or access event is set to the corresponding trace event, the mark corresponding to the type of the event is displayed.

Note that the marks displayed are not marks from tracing but are event marks set when the trace results are displayed. For details, refer to **Table 6-17 Meaning of Event Display Mark**.

(2) Trace mode display area



This is where the type of trace mode is displayed.  
The display contents are as follows.

- T: Delay trigger frame
- N: Frame in which not all the Lost Data was traced

(3) Trace result display area

Frame	Time	Address	Data	Status	Address	Data	Status	DisAsm
00750	309				03FFE207	04	R	
00751	310				03FFE20C	00000752	R	
00752	310	00000752	0A58	BRM1				mov r10, r11
00753	310				03FFE01C	00000001	R	
00754	311				03FFE00C	00000005	W	
00755	311	0000075E	42063F00	M1				dispose 0x4, lp, [lp]

(a)            (b)            (c)            (d)            (e)

This is where the trace results are displayed. If this area is selected, the jump function and the window linking function can be used.

The following contents are displayed in the trace results display area. Note that display or non-display of each area can be selected in the Trace Data Select dialog box.

- Trace frame number display
- Time tag display
- Fetch operation access display
- Data operation access display
- Mnemonic display

(a) Trace frame number display (Frame)

This is where the trace frame number is displayed.

However, complemented frames<sup>Note</sup> other than branch frames and access frames do not have frame numbers and this area is left blank.

**Note** A function to embed a dummy fetch frame in between branch frames.



**(b) Time tag display (Time)**

This area displays the number of clocks required for the target chip from the execution start of the program to the execution start of an instruction of each frame or generation of memory access cause is displayed. If Add Up Timetag is checked, an accumulate display is shown.

However, complemented frames other than branch frames and access frames do not have time tag display and this area is left blank.

The time tag display can be switched between clock count display and time display in the Trace View Select dialog box.

**(c) Fetch operation access display (Address Data Status)**

This is where the fetch result of the program is displayed. This field is displayed as follows depending on the status displayed in the status display field (Status).

**Table 6-23. Contents of Fetch Operation Access Display**

Item	Display Contents		
Status	BRM1	Program fetch display	The first byte fetch of the first instruction after branching
	M1		The first byte fetch of the instruction
	Blank		6th byte and 8th byte fetch of the instruction

If 6-byte long and 8-byte long instruction codes are displayed, the top 4 bytes are displayed in the first frame, and the remaining bytes are displayed in the second frame and third frame. Also, if 2 instructions are executed simultaneously, 1 frame's display becomes 2 lines, the instruction code in the lower address side is displayed in the first line, and the instruction code in the higher address side is displayed in the second line.

In the case of the program fetch display, the display is as follows.

**Table 6-24. Contents of Program Fetch Display**

Item	Display Contents
Address	Fetch address display
Data	Fetch data display

**(d) Data operation access display (Address Data Status)**

This is where the access result of data is displayed.

**Table 6-25. Contents of Data Operation Access Display**

Items	Display Contents	
Status	R	Data read
	W	Data write

**(e) Mnemonic display (DisAsm)**

This is where the disassemble result is displayed.

This is displayed only when the status of the fetch operation access display is BRM1 or M1.

If the instruction code which execute two instructions at a time is displayed, one instruction is displayed in the first line, and another executed at the same time is displayed in the second line.

The jump function and window linking function are detailed below.

- Jump function
- Window link function

**<1> Jump function**

The jump function is executed by performing the following operations in order.

- (i) Move the cursor to the jump source line.
- (ii) In the main window, do either of the following operation.
  - If the jump destination window is the Source window
    - Select [Jump] → [Source] from the menu bar.
    - Press the **[GRPH] + [J]**, and **[S]** keys in that order.
    - Press the shortcut keys **[CTRL] + [U]**.
  - If the jump destination window is the Assemble window
    - Select [Jump] → [Assemble] from the menu bar.
    - Press the **[GRPH] + [J]**, and **[A]** keys in that order.
    - Press the shortcut keys **[CTRL] + [D]**.
  - If the jump destination window is the Memory window
    - Select [Jump] → [Memory] from the menu bar.
    - Press the **[GRPH] + [J]**, and **[E]** keys in that order.
    - Press the shortcut keys **[CTRL] + [M]**.

The address used as the jump source changes according to the cursor position.

Caret Position	Jump Pointer
Access address display area	Access address <sup>Note</sup>
Access data display area	Access address <sup>Note</sup>
Access status display area	Access address <sup>Note</sup>
Other than above	Fetch address

**Note** If the jump destination is the Source window or the Assemble window, the fetch address is always made the jump pointer.

**<2> Window link function**

The window link function can be used by performing the following operations in order.

- (i) Make the Trace View window the current window and select the window that you would like to link with from among the items in [View] → [Window Synchronize] on the menu bar.

**Table 6-26. Window Link Function**

Items in [Window Synchronize]	Window To Be Linked
Source	Source window
Assemble	Assemble window
Memory	Memory window

- (ii) Move the cursor to the link origin line in the trace results display area in the Trace View window. If the window you would like to link with is not open in the active state, it will open in the active state.
- (iii) The trace results address selected in (ii) is used as the pointer, and the linked contents are displayed in the corresponding window selected in (i), or the cursor is positioned there.

In the case of window linking as well, the address which is the link origin changes as follows in accordance with the cursor position.

Cursor Position	Jump Pointer
Access address display area	Access address <sup>Note</sup>
Access data display area	Access address <sup>Note</sup>
Access status display area	Access address <sup>Note</sup>
Other than above	Fetch address

**Note** If the linking source is the Source window or the Assemble window, the fetch address is always made the jump pointer.

Unlike the jump function, if the cursor's position is moved in the Trace View window, the window linking function moves the reverse video in each linked window.

<b>Function buttons</b>
-------------------------

Search...
-----------

Cannot be selected.

<<
----

Cannot be selected.

>>
----

Cannot be selected.

Refresh
---------

Updates the contents of the window with the latest data.

Close
-------

Close this window.

<b>Trace Data Select Dialog Box</b>	<b>Modal</b>
-------------------------------------	--------------

### General

This dialog box selects the items to be displayed in the Trace View window.  
This dialog box is valid only for CPU cores with the TRCU incorporated.

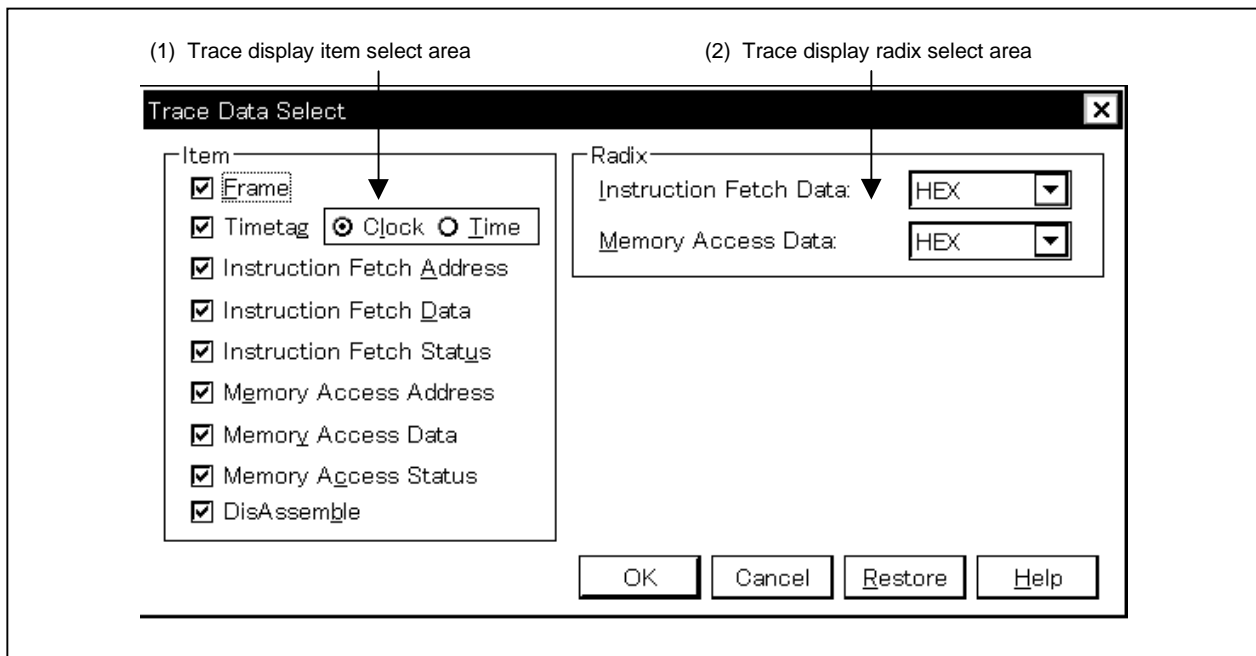
### Opening method

This dialog box can be opened in the following methods when the current window is the Trace View window.

- In the main window  
Select [View] → [Select] from the menu bar.  
Press the [GRPH] + [V], and [E] keys in that order.

### Window

**Figure 6-54. Trace Data Select Dialog Box**

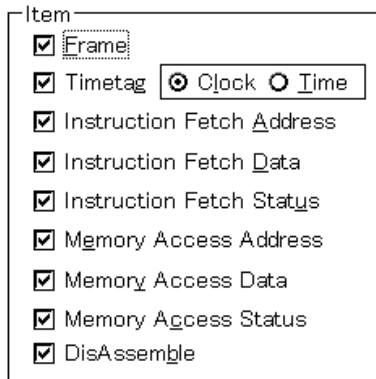


### Function

The Trace Data Select dialog box consists of the following areas.

- Trace display item select area
- Trace display radix select area

The function of each area is explained below.

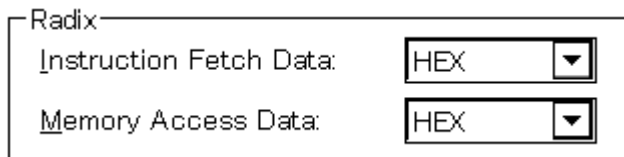
**(1) Trace display item select area**


This is where the items to be displayed in the Trace View window are selected.

The following items can be selected to be displayed or not displayed.

If there is a check mark, that area is displayed.

Frame:	Frame no. display field
Timetag:	Time tag display field
	• Clock: Clock display (default)
	• Time: Time display
Instruction Fetch Address:	Fetch address display field
Instruction Fetch Data:	Fetch data display field
Instruction Fetch Status:	Fetch status display field
Memory Access Address:	Access address display field
Memory Access Data:	Access data display field
Memory Access Status:	Access status display field
DisAssemble:	Disassemble display field

**(2) Trace display radix select area**






This is where the radix to be displayed is selected. The display radix can be selected for the following items.

Instruction Fetch Data:	Fetch data display field
Memory Access Data:	Access data display field

The display radix can be selected from the following.

Hex:	Displayed in hexadecimal
Dec:	Displayed in decimal
Oct:	Displayed in octal
Bin:	Displayed in binary

**Function buttons**

- |   |  |
|---|--|
|  | Reflects the result of selection in the Trace View window. |
|  | Closes this dialog box.                                    |
|  | Cancel the change and restore the original setting.        |
|  | Opens the help window.                                     |

<b>Trace Option Dialog Box</b>	<b>Modal</b>
--------------------------------	--------------

### General

This dialog box makes various settings related to trace such as the trace fetch mode setting and trace data selection.

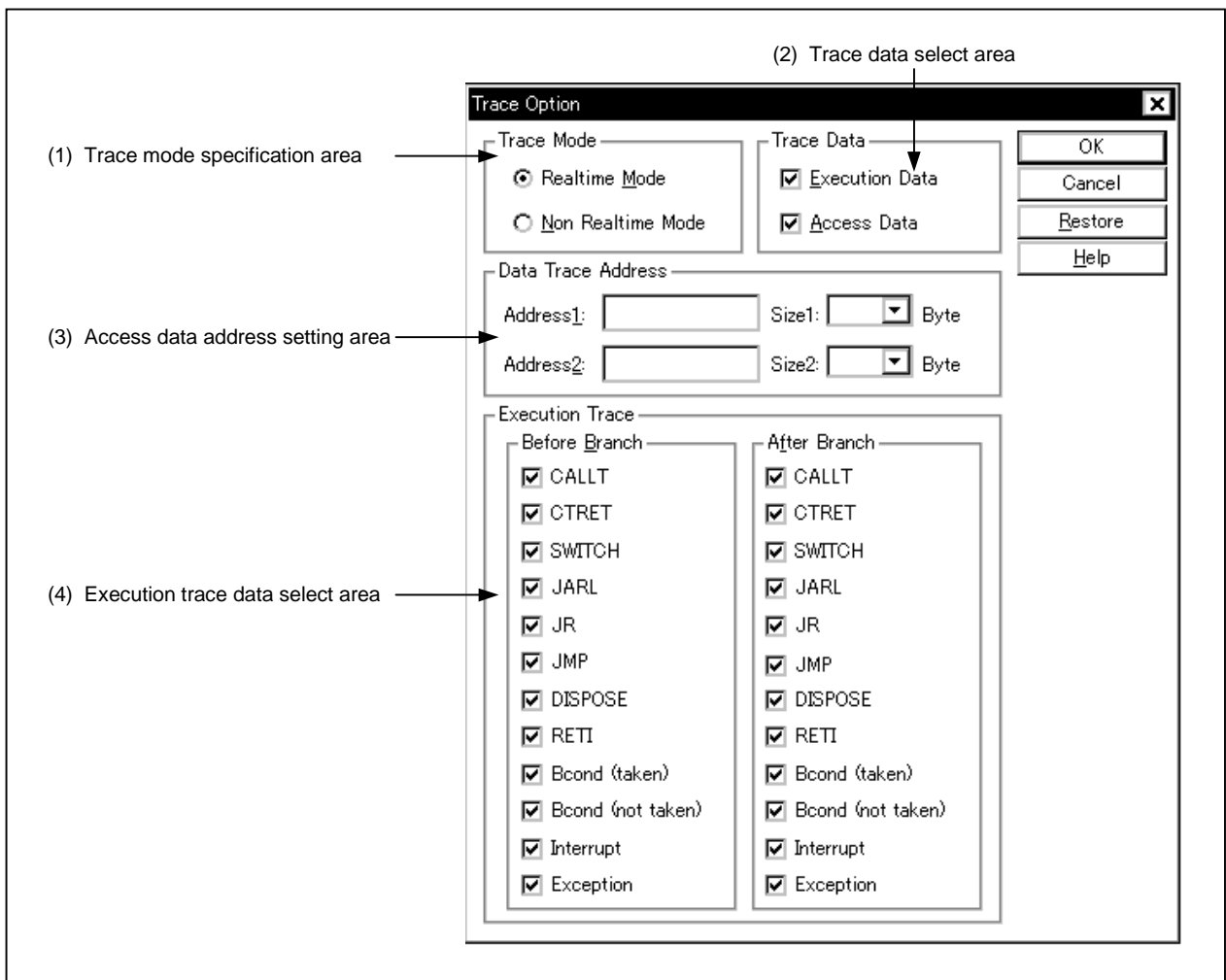
This dialog box is valid only for CPU cores that incorporate the TRCU.

### Opening method

- In the main window  
Select [Option] → [Trace] → [Trace Option...] from the menu bar.  
Press the [GRPH] + [O], [T], and [O] keys in that order.

### Window

**Figure 6-55. Trace Option Dialog Box**



<b>Function</b>
-----------------

This dialog box sets the trace mode, selects the trace data, and specifies the data trace range.  
The Trace Option dialog box consists of the following areas.

- Trace mode specification area
- Trace data select area
- Access data address setting area
- Execution trace data select area

The function of each area is explained below.

### (1) Trace mode specification area

Trace Mode
<input type="radio"/> Realtime Mode
<input checked="" type="radio"/> Non Realtime Mode

This area selects the mode in which the trace data is fetched. Select from the following two modes.

- Realtime Mode: Real-time trace mode.  
Traces the CPU execution in real time.  
Note, however, that not all the trace data may be fetched, depending on the program or trace trigger.
- Non Realtime Mode: Non real-time trace mode.  
This is the all trace mode. If some trace data is likely to be lost, the CPU execution is temporarily stopped internally, and all the trace data is properly fetched.

### (2) Trace data select area

Trace Data
<input checked="" type="checkbox"/> Execution Data
<input checked="" type="checkbox"/> Access Data

This area specifies the trace data to be fetched.

- Execution Data: Execution data.  
Traces the branch source set in the execution trace data select area.  
This makes it possible to ascertain the program execution flow.
- Access Data: Access data.  
Traces the data access to the specified address. Traces the data access cycle in the address range specified in the data trace address setting area.



**(3) Access data address setting area**

Data Trace Address

Address1:  Size1:  Byte

Address2:  Size2:  Byte

This area specifies the address where the access data trace is performed.

Inputting to this area is possible only when access data is selected in the trace data select area.

Up to 2 addresses can be set and the address and size must be specified.

Note, however, that the Address2 setting is ignored when section trace conditions are enabled (section trace conditions are set and [Run] menu → [Cond. Trace ON] is selected).

Address1, Address2: Any address can be set to the address area, but after setting, the address will be aligned with addresses set in 1 KB units (lower addresses: 000, 400, 800, C00).

Size1, Size2: The following values can be selected for the size area (unit byte).

4, 8, 16, 32, 64, 128, 256, 512, 1024

**(4) Execution trace data select area**

Execution Trace

Before Branch	After Branch
<input type="checkbox"/> CALLT	<input checked="" type="checkbox"/> CALLT
<input type="checkbox"/> CTRET	<input checked="" type="checkbox"/> CTRET
<input type="checkbox"/> SWITCH	<input type="checkbox"/> SWITCH
<input checked="" type="checkbox"/> JARL	<input checked="" type="checkbox"/> JARL
<input checked="" type="checkbox"/> JR	<input checked="" type="checkbox"/> JR
<input checked="" type="checkbox"/> JMP	<input checked="" type="checkbox"/> JMP
<input checked="" type="checkbox"/> DISPOSE	<input checked="" type="checkbox"/> DISPOSE
<input type="checkbox"/> RETI	<input type="checkbox"/> RETI
<input checked="" type="checkbox"/> Bcond (taken)	<input checked="" type="checkbox"/> Bcond (taken)
<input checked="" type="checkbox"/> Bcond (not taken)	<input checked="" type="checkbox"/> Bcond (not taken)
<input checked="" type="checkbox"/> Interrupt	<input type="checkbox"/> Interrupt
<input checked="" type="checkbox"/> Exception	<input type="checkbox"/> Exception

This area selects the trigger to start an execution trace from amongst the program execution branch triggers.

Inputting to this area is possible only when execution data is selected in the trace data select area.

The trace triggers that can be selected are divided into the following two types.

Before Branch area: If the checked trigger in this area is an instruction of the branch origin, this instruction is fetched as trace data.

After Branch area: If the checked trigger in this area is an instruction of the branch origin, this instruction is fetched as trace data.

The check items and their meanings are as follows.

The trace data to be fetched consists only of branch triggers selected in this area. If the number of fetched triggers to be fetched is small, trace data supplementation may not be performed correctly.

**Table 6-27. Trace Triggers of Execution Trace Data**

Item	Meaning
CALLT	Branch by CALLT instruction
CTRET	Branch by CTRET instruction
SWITCH	Branch by SWITCH instruction
JARL	Branch by JARL instruction
JR	Branch by JR instruction
JMP	Branch by JMP instruction
DISPOSE	Branch by DISPOSE instruction
RETI	Branch by RETI instruction
Bcond(taken)	Branch by conditional branch instruction (when condition is satisfied)
Bcond(not taken)	Branch by conditional branch instruction (when condition is not satisfied)
Interrupt	An interrupt (NMI/INT) is generated.
Exception	An exception occurs.

#### Function buttons

OK

Closes this dialog box after setting the specified trace mode, trace data, and trace address.

Cancel

Closes this dialog box without setting the specified trace mode, trace data, and trace address.

Restore

Returns the Trace Option dialog box to the original setting status before it was opened.

Help

Opens the help window.

<b>Event Manager</b>	<b>MDI child window</b>
----------------------	-------------------------

### General




This window displays, enables/disables and deletes various events.

Through manipulation in this window, the event conditions registered in the Event dialog box or Event Link dialog box can be allocated to break event or trace event<sup>Note</sup>.

**Note** The trace function is valid only for CPU cores with the TRCU incorporated.

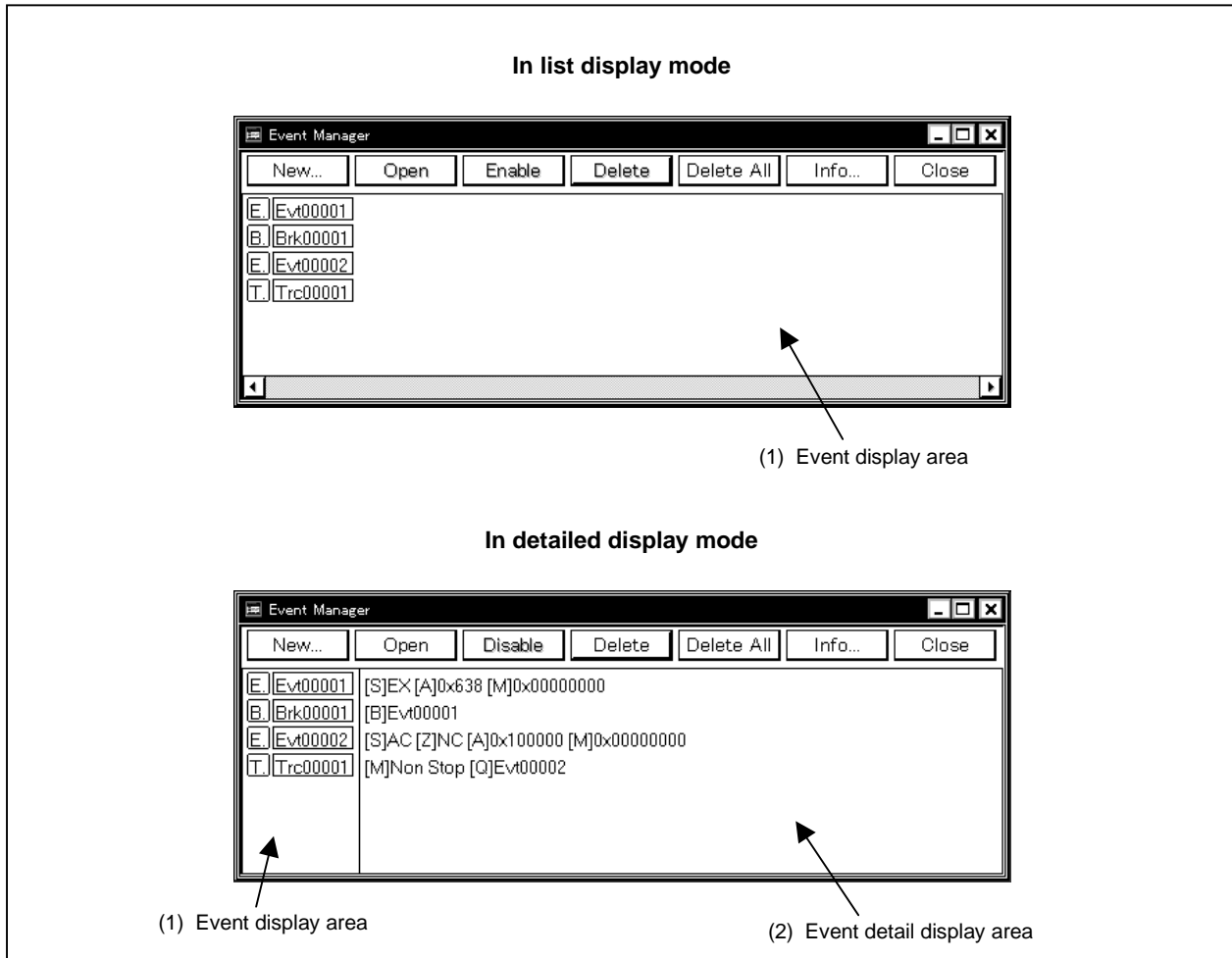
### Opening method

This window can be opened in the following methods.

- In the main window
  - Select [Event] → [Event Manager] from the menu bar.
  - Press the **[GRPH] + [N]**, and **[M]** keys in that order.
  - Click the  button from the toolbar.
- In the Source window
  - Moves the cursor to the source line in the source text display area where the event is set, or to the disassemble line in case of the mixed display mode, and carry out either of the following methods.
  - Select [View] → [Event?] from the menu bar.
  - Press the **[GRPH] + [V]**, and **[E]** keys in that order.
- In the Assemble window
  - Moves the cursor to the line in the mnemonic display/change area where the event is set, and carry out either of the following methods.
  - Select [View] → [Event?] from the menu bar.
  - Press the **[GRPH] + [V]**, and **[E]** keys in that order.
- In the Event dialog box
  - Click the  button.
  - Press the shortcut keys **[GRPH] + [G]**.
- In the Set Other dialog box
  - Click the  button.
  - Press the shortcut keys **[GRPH] + [M]**.

## Window

Figure 6-56. Event Manager



## Function

This window displays, enables/disables, and deletes various events and manages event conditions for registering or setting various event conditions (such as event link conditions, break event conditions, and trace event conditions<sup>Note</sup>).

The Event Manager consists of the following areas.

- Event display area
- Event detail display area

**Note** The trace function is valid only for CPU cores with the TRCU incorporated.

The function of each area is explained below.

**(1) Event display area**

- In list display mode

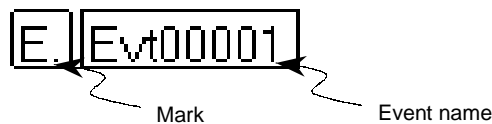
E.	Evt00001
B.	Brk00001
E.	Evt00002
T.	Trc00001

- In detailed display mode

E.	Evt00001	[S]EX [A]0x638 [M]0x00000000
B.	Brk00001	[B]Evt00001
E.	Evt00002	[S]AC [Z]NC [A]0x100000 [M]0x00000000
T.	Trc00001	[M]Non Stop [Q]Evt00002

In the event display area, icons (event icons) for various types of event conditions of the registered event, event link, break, and trace<sup>Note</sup> are displayed.

The icon of the event condition consists of a mark indicating the type of each event condition and an event name.



The switching list/detailed display mode in the event display area can be changed by the Event Info dialog box which is opened by the [View] menu or the  button in the Event Manager.

**Note** The trace function is valid only for CPU cores with the TRCU incorporated.

The marks are listed below.

**Table 6-28. Event Icon Marks**

Mark	Meaning
<input type="checkbox"/> E.	Indicates event condition
<input type="checkbox"/> L.	Indicates event link condition
<input type="checkbox"/> B.	Indicates break event
<input type="checkbox"/> T.	Indicates trace event <sup>Note</sup>

**Note** Valid only for CPU cores with the TRCU incorporated.

The color of the character displayed in a mark indicates the setting status of each event condition.

Table 6-29. Character Color of Event Icon

Character Color in Mark	Corresponding Mark	Meaning
Red (valid)	<b>E.</b> , <b>L.</b>	Indicates that the event conditions using events and event link conditions are valid.
	<b>B.</b> , <b>T.</b>	Indicates that each event condition is "enabled". Each event is generated when condition is satisfied.
Black (invalid)	<b>E.</b> , <b>L.</b>	Indicates that the event conditions using events and event link conditions are invalid.
	<b>B.</b> , <b>T.</b>	Indicates that an event is invalid. The event does not occur even if the conditions are established.
Yellow (pending)	<b>E.</b> , <b>L.</b> , <b>B.</b> , <b>T.</b>	Indicates that the symbol specified in the event cannot be recognized by the program that is currently loading, so it is held pending.

In addition to the above function, this area also has the following four functions.

#### (a) Jump function

Jumps to the Source window, Assemble window, or Memory window by using the address condition of the selected icon as a jump pointer if the selected icon is the event condition. The jump destination window displays the jump pointer at the beginning.

If the address condition is a range setting, the lower address becomes the jump pointer. If a mask is set, the point address before the mask is executed becomes the jump pointer.

In the case of a bit setting, the address of the specified bit position becomes the jump pointer.

This function is executed by performing the following operations in the sequential order.

<1> Select an icon.

<2> Perform one of the following operations in the main window.

- If the jump destination is the Source window  
 Select [Jump] → [SourceText] from the menu bar.  
 Press the **GRPH** + **J**, and **S** keys in that order.  
 Press shortcut keys **CTRL** + **U**.
- If the jump destination is the Assemble window  
 Select [Jump] → [Assemble] from the menu bar.  
 Press the **GRPH** + **J**, and **A** keys in that order.  
 Press shortcut keys **CTRL** + **D**.
- If the jump destination is the Memory window  
 Select [Jump] → [Memory] from the menu bar.  
 Press the **GRPH** + **J**, and **E** keys in that order.  
 Press shortcut keys **CTRL** + **M**.

**(b) Open function**

This function opens the setting dialog box corresponding to the selected icon's event. The contents of the selected event are displayed in each setting dialog box that is opened.

This function is executed by performing one of the following operations.

- <1> Double-click the icon.
- <2> Select the icon, then click .

**(c) Enabled state switching function**

This function switches the state of the event corresponding to the selected icon between enabled and disabled.

Except for an event and event link, it is possible to operate break and trace<sup>Note</sup> events by icons.

This function is executed by performing one of the following operations.

- <1> Click the left button of the mouse when the mouse cursor is on the icon mark.
- <2> Select an icon and click  or , or press the  +  keys.

If an event is in the enabled state, it becomes disabled (black), and if the event is in the disabled state, it becomes enabled (red).

The  button is displayed when an icon in the disabled state is selected, and the  button is displayed when an icon in the enabled state is selected.

**Note** The trace function is valid only for CPU cores with the TRCU incorporated.

**(d) Delete function**

Deletes registration/set contents of the event condition of a selected icon.

Multiple icons can be selected using the  and  keys.

To select all the icons, select  →  from the menu bar.

However, an event condition  and event link condition  can be deleted only when that event condition is not used by other event condition ( or ). If the event condition or event link condition has been already used for any other event, delete the event used.

This function is executed by performing one of the following operations.

- <1> Select an icon and perform one of the following operations in the main window.
  - Select  →  from the menu bar.
  - Press the  + , and  keys in that order.
- <2> Select an icon and perform one of the following operations in the Event Manager.
  - Click  button.
  - Press the  key.

## (2) Event detail display area

```
[S]EX [A]0x638 [M]0x00000000
[B]Evt00001
[S]AC [Z]NC [A]0x100000 [M]0x00000000
[M]Non Stop [Q]Evt00002
```

This area is displayed only when the detailed display mode is selected.

This area displays the detailed information corresponding to the icon of each event. This area displays the contents of the status condition, access size condition, data condition, and mask condition, in that order, by using the information of each key as a separator, when an event condition is specified. The following tables show the correspondence between the information of each key and the condition.

**Table 6-30. Key Information in Detailed Event Display Mode**

- In the case of event condition

Key Information	Contents
[S]	Status condition
[Z]	Access size condition
[A]	Address condition (The actual address is displayed within ( ))
[D]	Data condition
[M]	Mask condition

- In the case of event link condition

Key Information	Contents
[P1] to [P4]	Event link condition of nth stage (n = 1 to 4)
[D]	Disable condition
[P]	Pass count condition

- In the case of break condition

Key Information	Contents
[B]	Break condition

- In the case of trace conditions<sup>Note</sup>

Key Information	Description
[S]	Section trace start condition
[E]	Section trace end condition
[T]	Delay trigger condition

**Note** Valid only for CPU cores with the TRCU incorporated.



## Function buttons

<input type="button" value="New..."/>	<p>This button opens the New Event dialog box.</p> <p>In the New Event dialog box, by clicking each button, the setting dialog box for event, event link, break, and trace events can be opened.</p> <p>Each setting dialog box that is opened has the new event name set in it.</p> <p>After each setting dialog box is opened, the New Event dialog box is closed.</p> <p>If <input type="button" value="Cancel"/> is clicked or the <input type="button" value="ESC"/> key is pressed, the New Event dialog box closes and the screen returns to the Event Manager.</p>
<input type="button" value="Open"/>	<p>Opens the setting dialog box corresponding to the event condition selected in the event display area. The event condition contents selected in the event display area are displayed in the setting dialog box.</p> <p>If the event condition has not been selected, or if multiple conditions are selected, this button is displayed in gray and can no longer be clicked.</p> <p>The <input type="button" value="Enter"/> key also has the same operation.</p>
<input type="button" value="Enable"/>	<p>Sets break and trace events in the enabled state.</p> <p>The <input type="button" value="Enable"/> button is displayed in place of the <input type="button" value="Disable"/> button if one disabled event or multiple events including disabled only events are selected.</p>
<input type="button" value="Disable"/>	<p>Sets break and trace events in the disabled state.</p> <p>The <input type="button" value="Disable"/> button is displayed in place of the <input type="button" value="Enable"/> button if one enabled event is selected or multiple events including enabled events are selected.</p> <p>If an event or an event link only is selected, this button is displayed in gray and it can no longer be clicked.</p>
<input type="button" value="Delete"/>	<p>Deletes the event condition selected in the event display area.</p> <p>When deleting an event or event link, if that event is being used by a break, trace, or event link, it results in an error. The <input type="button" value="DEL"/> key also has the same operation.</p>
<input type="button" value="Delete All"/>	<p>Deletes all the events whether they are enabled or disabled.</p>
<input type="button" value="Info..."/>	<p>Opens the Event Info dialog box.</p> <p>The <input type="button" value="Sort by Name"/> button rearranges events in name order and displays them.</p> <p>The <input type="button" value="Sort by Kind"/> button rearranges events by kind and displays them.</p> <p>The <input type="button" value="Unsort"/> button displays the events in the order in which they were entered without rearranging them.</p> <p>The <input type="button" value="Detail"/> button sets the display mode to the detailed display mode.</p> <p>The <input type="button" value="Overview"/> button sets the display mode to the list display mode.</p> <p>The <input type="button" value="Cancel"/> button or the <input type="button" value="ESC"/> key closes the dialog box.</p>
<input type="button" value="Close"/>	<p>Closes the Event Manager.</p>

<b>Software Break Manager</b>	<b>MDI child window</b>
-------------------------------	-------------------------

### General

Displays, enables/disables, and deletes software breaks.

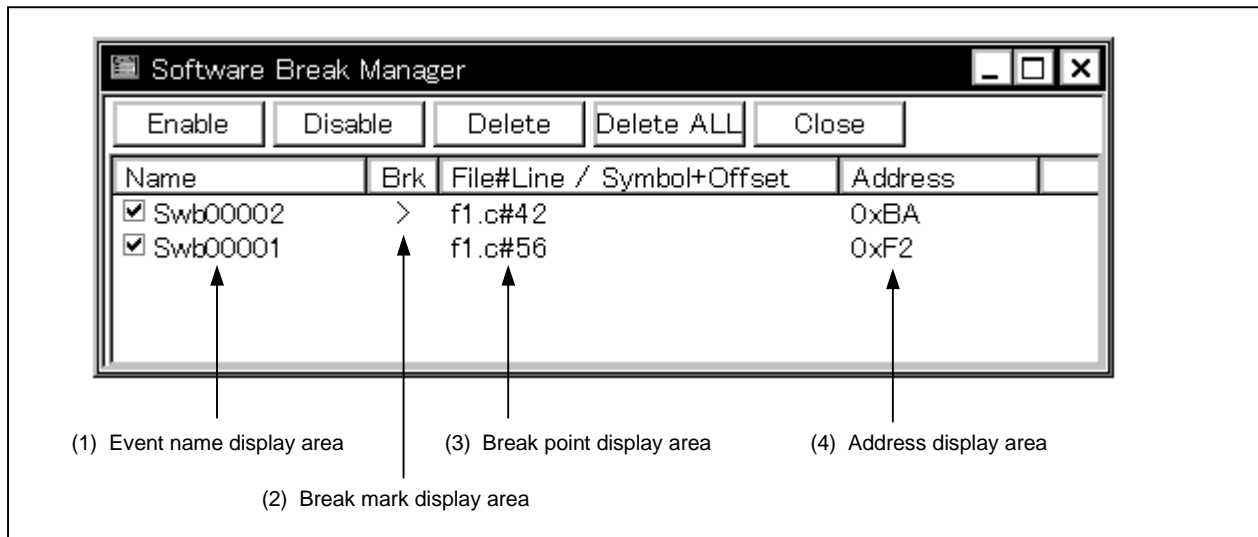
### Opening method

This window can be opened by either of the following methods.

- In the main window  
Select [Event] → [Software Break Manager] from the menu bar.  
Press the [GRPH] + [N], and [M] keys in that order.

### Window

**Figure 6-57. Software Break Manager**



### Function

Software breaks can be displayed, enable/disable switched, and deleted in this window.

Up to 100 software break events can be enabled at the same time.

Note that software breakpoints cannot be set in this window. Software breakpoints are set in either the Source or Assemble window.

The Software Break Manager consists of the following areas.

- Event name display area
- Break mark display area
- Break point display area
- Address display area

The functions of each area is explained below.

### (1) Event name display area

Name	Brk	File#Line / Symbol+Offset	Address
<input checked="" type="checkbox"/> Swb00002	>	f1.c#42	0xBA
<input checked="" type="checkbox"/> Swb00001		f1.c#56	0xF2

This area displays the names of recorded events and includes check boxes that indicate their enabled/disabled status.

The name of an event can be directly edited by clicking the selected name. Press the return button to set the change. The default setting is indicated by “Swb+[Number]”.

Enable events by selecting the relevant check box (check boxes are selected as the default setting for a new creation).

Double-clicking an event allows you to jump to the window in which the event was created (Source or Assemble window).

#### [Reference]

Click Name on the title bar to sort the displayed items into dictionary order (alphabetical– ascending or descending (switchable by clicking)).

### (2) Break mark display area

Name	Brk	File#Line / Symbol+Offset	Address
<input checked="" type="checkbox"/> Swb00002	>	f1.c#42	0xBA
<input checked="" type="checkbox"/> Swb00001		f1.c#56	0xF2

This area displays a “>” mark for the software break event set at the current PC position (making it easy to specify the software break event that caused the break).

### (3) Break point display area

Name	Brk	File#Line / Symbol+Offset	Address
<input checked="" type="checkbox"/> Swb00002	>	f1.c#42	0xBA
<input checked="" type="checkbox"/> Swb00001		f1.c#56	0xF2

The location at which the software break event is set is indicated by the following format.

File name#Line No. (When set in Source window)

Symbol+Offset (When set in Assemble window)

Events are reevaluated when symbols are redownloaded (etc.) based on these formats.

**[Reference]**

Click File#Line/Symbol+Offset on the title bar to sort the displayed items into dictionary order (alphabetical – ascending or descending (switchable by clicking)).

**(4) Address display area**

Name	Brk	File#Line / Symbol+Offset	Address
<input checked="" type="checkbox"/> Swb00002	>	f1.c#42	0xBA
<input checked="" type="checkbox"/> Swb00001		f1.c#56	0xF2

This area displays the address at which the software break event has been set.

**[Reference]**

Click Address on the title bar to sort the displayed items into numerical order (ascending or descending (switchable by clicking)).

**Function buttons**

Enable

Enables the selected event.

Disable

Disables the selected event.

Delete

Deletes the selected event.

Delete All

Deletes all the software break events that were set.

Close

Closes the Software Break Manager.

<b>Event Dialog Box</b>	<b>Modeless</b>
-------------------------	-----------------


### General

This dialog box registers and displays event conditions.

The event conditions created (registered) on this dialog box are automatically registered to the Event Manager.

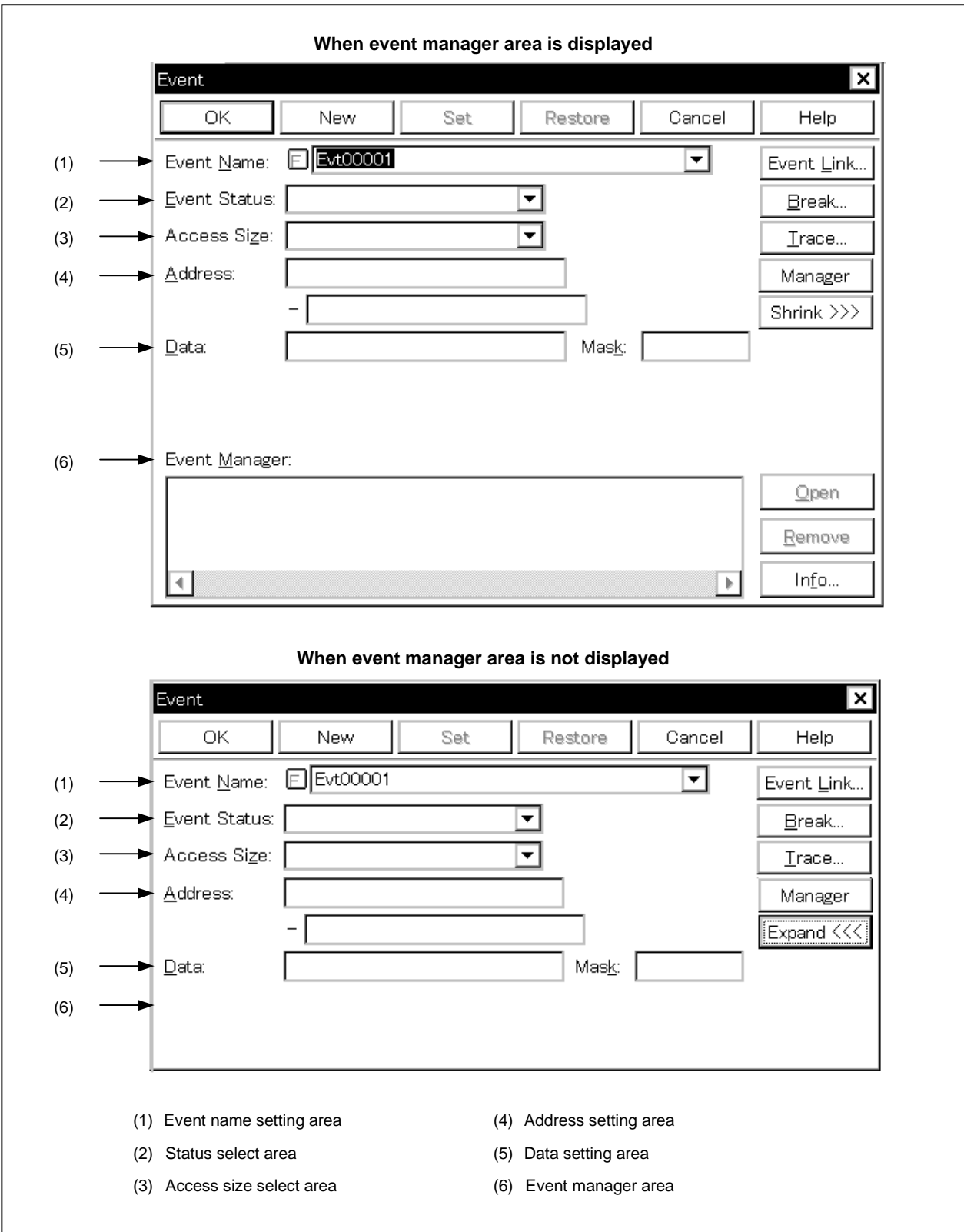
### Opening method

This dialog box can be opened in the following methods.

- In the main window
  - Select [Event] → [Event...] from the menu bar.
  - Press the **[GRPH]** + **[N]**, and **[E]** keys in that order.
  - Click the  button on the toolbar.
- Execute one of the following operations in the Break, Trace, or Event Link dialog box.
  - Click the **Add Event...** button.
  - Press the shortcut keys **[GRPH]** + **[E]**.
- Execute one of the following operations in the event area or the event manager area of the Break, Trace, or Event Link dialog box.
  - Select the event condition and click **Open**.
  - Select the event condition and press the shortcut keys **[GRPH]** + **[O]**.
  - Double-click the event condition.
- In the Event Manager, execute one of the following operations.
  - Select an event condition and click **Open**.
  - Double-click the event condition.
- Execute one of the following operations in the New Event dialog box or the Set Other dialog box.
  - Click the **Event...** button.
  - Press the shortcut keys **[GRPH]** + **[E]**.

Window

Figure 6-58. Event Dialog Box



<b>Function</b>
-----------------

This dialog box registers and displays event conditions.

Up to 256 conditions can be entered for event conditions. However, the maximum number of event conditions which can be used simultaneously in enabled break and trace events is as follows, even including event conditions used via the event link.


**Table 6-31. Number of Valid Event Conditions**

With/Without TEU	Maximum Valid Number
With TEU	8 (execution events <sup>Note 1</sup> , two access events)
Without TEU	2 <sup>Note 2</sup>


- Notes 1.** Events before execution: 2, Events after execution: 4. Events before execution can be used only as break events but cannot specify the address range.
- 2.** Exclusive use of events before execution and access events and address range specification are prohibited.


The Event dialog box has two modes, the normal mode and the select mode.

- **Normal mode**

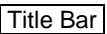
When  button from the toolbar or [Event] → [Event...] from the menu bar in the main window is selected and the Event dialog box is opened, event conditions can be entered without restriction as to purpose. This is the normal mode.

- **Select mode**

When  button in the Break, Trace, and Event Link dialog boxes is clicked and the Event dialog box is opened, the event conditions used by these are selected. This is the select mode.

In the select mode, when  is clicked, the event condition selected in the event name setting area of the Event dialog box is set in the event condition setting area of the setting dialog box that originally called the Event dialog box.

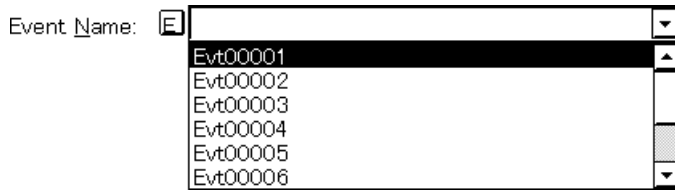
Even in the select mode, event conditions can be entered and displayed just as in the normal mode.

Depending on the mode, the title bar display on the dialog box will differ. Refer to the section entitled  described later on.

The Event dialog box consists of the following areas.

- Event name setting area
- Status select area
- Access size select area
- Address setting area
- Data setting area
- Event manager area

The function of each area is explained below.


**(1) Event name setting area**


This area sets an event name and selects an event condition.

When entering an event condition, it is necessary to set an event name. An event name may consist of up to eight alphanumeric characters.


When displaying event conditions which have already been created, either type the event name of the already created event name in the text box or select the event condition from the drop-down list in the event name setting area.

In the select mode, the selected event condition can be set in the event condition setting area of the original setting dialog box that called the Event dialog box.

The  mark on the left side of the event name setting area shows the use conditions of the event condition.

The relationship between the color of the character in the  mark and the status is shown below.

**Table 6-32. Meanings of Event Icons (Event Condition)**

Color of Character in  Mark	Condition
Red (valid)	That event condition is being used by break, trace, or event link and indicates that it is enabled.
Black (invalid)	Indicates that that event condition has not been enabled.
Gray (not entered)	Indicates that that event condition is currently being edited and that it has not been entered.
Yellow (pending)	Indicates that that event is an event which is in the hold state. An event in the hold state is in a state where the symbol specified by the event condition by program download, etc. cannot be referred to.



**(2) Status select area**

. Event Status:

This area sets and selects a status condition.

By specifying the status condition, the distinction between the execution event and access event is decided at the same time. If execution event is specified, the access size select area, the mask field of the address setting area and the data setting area are displayed in gray and are in a state where input is impossible.

The status conditions that can be specified are shown below. It is possible to specify a status condition in abbreviated form.

The input status condition is not case sensitive.

**Table 6-33. List of Status Conditions in Event Dialog Box**

Status	Abbr.	Event Type	Meaning
Execution	EX	Execution event	Program execution
Before Execution <sup>Note</sup>	EX-B		Program execution (Break before execution)
R/W	RW	Access event	Data read/write
Read	R		Data read
Write	W		Data write
Access	AC		All access statuses

**Note** Up to 2 events can be made to the enable status in Before Execution and only break event conditions can be used. Also, the address range cannot be set.

**(3) Access size select area**

Access Size:

Byte  
Half Word  
Word  
No Condition  
Bit

This is where the access size conditions are set and selected.

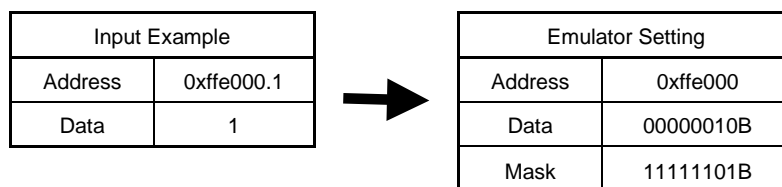
By specifying the access size conditions, the access width for data conditions during access data search is decided.

The access size conditions that can be specified are shown below. It is possible to specify the access size conditions in abbreviated form. The input access size condition is not case sensitive.

**Table 6-34. List of Access Size Conditions in Event Dialog Box**

Size	Abbr.	Meaning
Byte	B	Detection is made with an 8-bit width as the data condition. Detection is made only during 8-bit access.
Half Word	HW	Detection is made with a 16-bit width as the data condition. Detection is made only during 16-bit access.
Word	W	Detection is made with a 32-bit width as the data condition. Detection is made only during 32-bit access.
No Condition	NC	An access-size detection is not executed.
Bit	1	Detection is carried out with a 1-bit width as the data condition. Detection is carried out only during 8-bit access.

If Bit is specified as the access size condition, detection is made with a 1-bit width as the data condition, but due to emulator operation, since access to the bit itself cannot be detected directly, the debugger detects for a pseudo bit access by setting the address conditions and data conditions internally as follows.



For that reason, for access to other bits in the same address, or even for access to all 8 bits in the same address, if the values match the specified [Address.bit] value, an event occurs in accordance with the specified status.

Also, during bit writing when the device is in operation, reading and writing with a full 8 bits is executed, but if read or read/write is specified in the status, an event occurs if the specified [Address.bit] value matches with respect to the read operation at this time.

- If No Condition is specified, the data setting area is displayed in gray and input cannot be done.
- If setting the access size condition is omitted, it is judged automatically from the address conditions and data conditions and set.
- If a bit is set for the address condition, the condition becomes Bit.
- If 8 bits is set for the data condition, the condition becomes Byte.
- If 16 bits is set for the data condition, the condition becomes Half Word.
- If 32 bits is set for the data condition, the condition becomes Word.
- If the data condition specification is omitted, the condition becomes No Condition.

This area is blank in the case of newly created event conditions.

**(4) Address setting area**

Address: <input type="text"/> Mask: <input type="text"/> - <input type="text"/>
--

This area sets an address condition.  
The setting range is as shown below.

Setting range:  $0 \leq \text{Address value} \leq 0\text{xFFFF FFFF}$

Input the lower addresses and the higher addresses in that order.  
The following three types of settings can be made in specifying an address condition.

**<1> Point setting**

Only the lower address or the same address value is set to the lower and higher addresses when a point is set.

**<2> Range setting**

A range is specified by the lower and higher addresses.

Note that if the same value is input for the lower and higher addresses, it is not handled as a range setting. In this case, it becomes a point setting or a bit setting. Moreover, if an event condition is changed during use and if there was a range setting prior to the change, there must be a range setting after the change. Similarly, if there was no range setting prior to the change, neither the address condition nor the range setting can be changed.

**<3> Bit setting**

For a bit setting, set a value in the lower addresses only, or set the same value in the lower and higher addresses. Specify the value in the address.bit format. The value of the bit that shows the bit position must be  $0 \leq \text{bit} \leq 7$ .

For the address, it is possible to specify a symbol or expression. The specification method is the same as in the Address Move dialog box.

The default radix when numerical values are input is hexadecimal.

This area is blank in the case of newly created event conditions.

**(5) Data setting area**

Data:  Mask:

This area sets a data condition.  
The setting range is as follows.

**Table 6-35. List of Data Conditions in Event Dialog Box**

Access Size Condition	Setting Range
Byte	0 ≤ Data value ≤ 0xFF 0 ≤ Mask value ≤ 0xFF
Half Word	0 ≤ Data value ≤ 0xFFFF 0 ≤ Mask value ≤ 0xFFFF
Word	0 ≤ Data value ≤ 0xFFFF FFFF 0 ≤ Mask value ≤ 0xFFFF FFFF
Bit	Data value = 0, 1 Mask value = Cannot be specified

If the data condition setting has been omitted, data condition detection is not performed.

For a newly created event condition, the address value and the mask value input columns are both blank.

There are two types of data conditions: Data that sets a data value, and Mask that sets a mask value of data value. How to set each of these types is explained below.

**(a) Data**

This area sets a data value.

A data value can also be specified by a symbol or expression. The specification method is the same as in the case of the Symbol To Address dialog box, but the default radix when numerical values are input is hexadecimal.

**(b) Mask**

This area sets a mask value for the data value.

When mask setting is made, for a bit with a mask value of 1, it doesn't matter if the data value is 0 or 1.

**Example 1.** Data:

Mask:

In this kind of setting, the conditions match when the data value is 0x4000 to 0x40FF.

**2.** Data:

Mask:

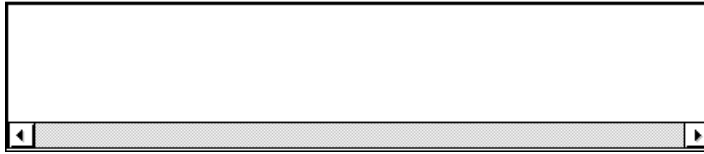
In this kind of setting, when the data are 0x4000, 0x4001, 0x4100, and 0x4101, they match the conditions.

If the mask value setting is omitted, masking is not executed.

The default radix when numerical values are input is hexadecimal.

**(6) Event manager area**

Event Manager:



This area displays a list of each of the entered events, event links, breaks, and trace events.

By selecting an event and clicking  , or by double-clicking the event, a setting dialog box opens corresponding to the selected event and the event setting contents can be displayed.

When the focus is on the event manager area, select the event icon, then click  , or press the  key and the selected event can be deleted.

By clicking  , it is possible to select the event manager area display mode or sequencing.

**Function buttons**

(Normal mode)

Closes the Event dialog box.

If there is an event condition that is being edited, it is entered automatically and the dialog box is closed.

(Select mode)

Sets the displayed event as the event condition of the original dialog box that called the Event dialog box.

If the Event dialog box is already open, the select mode returns to the normal mode only and the Event dialog box does not close. In other cases, it closes.

If there is an event condition being edited, enter/select is performed automatically and the program returns to the original setting dialog box.

Newly creates an event condition.

An event name is generated automatically and a newly created event condition is prepared. After it is created, be sure to enter the event condition using the  or the  button.

Enters an event condition.

When a new event condition has been created or when the contents of an event condition have been changed, be sure to enter the event condition using this button. The entered event condition is reflected in the Event Manager.

If event conditions that have already been entered are displayed, they cannot be selected.

Clears the contents of an event condition.

If an event condition is entered but not edited, this button is displayed in place of the  button.

<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 10px;">Restore</div>	<p>Returns the contents of an event condition to the original contents.</p> <p>If an event condition that has not been entered is displayed, all the area is made blank except the event name, or the settings return to the default settings.</p> <p>If an event condition is being edited, this button is displayed in place of the <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 5px;">Clear</div> button.</p>
<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 10px;">Close</div> (Normal mode)	<p>Closes the Event dialog box.</p> <p>If an event condition has been created/changed/deleted, after that, the <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 5px;">Cancel</div> button becomes this button.</p> <p>Even if there is an event condition that is being edited, the dialog box closes without the event condition being entered. The <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 5px;">ESC</div> key has the same operation.</p>
<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 10px;">Cancel</div> (Normal mode)	<p>Closes the Event dialog box.</p> <p>If an event condition is not being created/changed/deleted, this button becomes the <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 5px;">Close</div> button.</p> <p>Even if an event condition is being edited, the dialog box closes without the event condition being entered. The <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 5px;">ESC</div> key has the same operation.</p>
<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 10px;">Cancel</div> (Select mode)	<p>Returns to the original setting dialog box that called the Event dialog box without an event condition being selected.</p> <p>If an event condition is not being created/changed/deleted, the <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 5px;">Abort</div> button becomes this button.</p> <p>If the Event dialog box is already open, the select mode returns to the normal mode only, and the Event dialog box does not close. In other cases, the Event dialog box closes.</p> <p>Even if an event condition is being edited, it is not entered. The <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 5px;">ESC</div> key has the same operation.</p>
<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 10px;">Abort</div> (Select mode)	<p>Returns to the original setting dialog box without selecting the event condition.</p> <p>If an event condition has been edited/changed/deleted, the <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 5px;">Cancel</div> button becomes this button.</p> <p>If the Event dialog box is already open, the select mode returns to the normal mode only, and the Event dialog box does not close. In other cases, the Event dialog box closes.</p> <p>Even if an event condition is being edited, it is not entered. The <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 5px;">ESC</div> key has the same operation.</p>
<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 10px;">Help</div>	<p>Opens the help window.</p>
<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 10px;">Event Link...</div>	<p>Opens the Event Link dialog box.</p>
<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 10px;">Break...</div>	<p>Opens the Break dialog box.</p>
<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 10px;">Trace...</div>	<p>Opens the Trace dialog box.</p>
<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 10px;">Manager</div>	<p>Opens the Event Manager.</p>

Expand <<<

Displays the event manager area. Expands the size of the dialog box. When the event manager area is not displayed, this button is displayed in place of the **Shrink >>>** button.

Shrink >>>

Does not display the event manager area. Shrinks the size of the dialog box. When the event manager area is displayed, this button is displayed in place of the **Expand <<<** button.

Open

Opens the setting dialog boxes and displays the event, event link, break, or trace conditions selected in the event manager area.

Remove

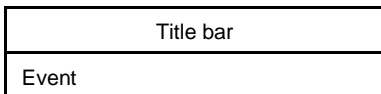
Deletes the event, event link, break, or trace conditions selected in the event manager area.

Info...

Opens the Event Info dialog box.  
 The **Sort by Name** button rearranges events in name order and displays them.  
 The **Sort by Kind** button rearranges the events in order by kind and displays them.  
 The **Unsort** button displays the events in the order in which they were entered without rearranging them.  
 The **Detail** button changes the event manager area to the detailed display mode.  
 The **Overview** button changes the event manager area to the list display mode.  
 The **Cancel** button or the **ESC** key closes the dialog box.

**Title bar**

In the normal mode, [Event] is displayed as the title in the title bar.



In the select mode, the title in the title bar becomes as follows in accordance with the original setting dialog box that called the Event dialog box.

Title Bar	Original Setting Dialog Box
Event – Break	Break dialog box
Event – Trace	Trace dialog box
Event – Event Link	Event Link dialog box
Event – Event Link – Break	Event Link dialog box (While in the Break dialog box select mode)
Event – Event Link – Trace	Event Link dialog box (While in the Trace dialog box select mode)

<b>Event Link Dialog Box</b>	<b>Modeless</b>
------------------------------	-----------------

### General

This dialog box registers and displays event link conditions.

The event link conditions created (registered) in this dialog box are automatically registered to the Event Manager.

### Opening method

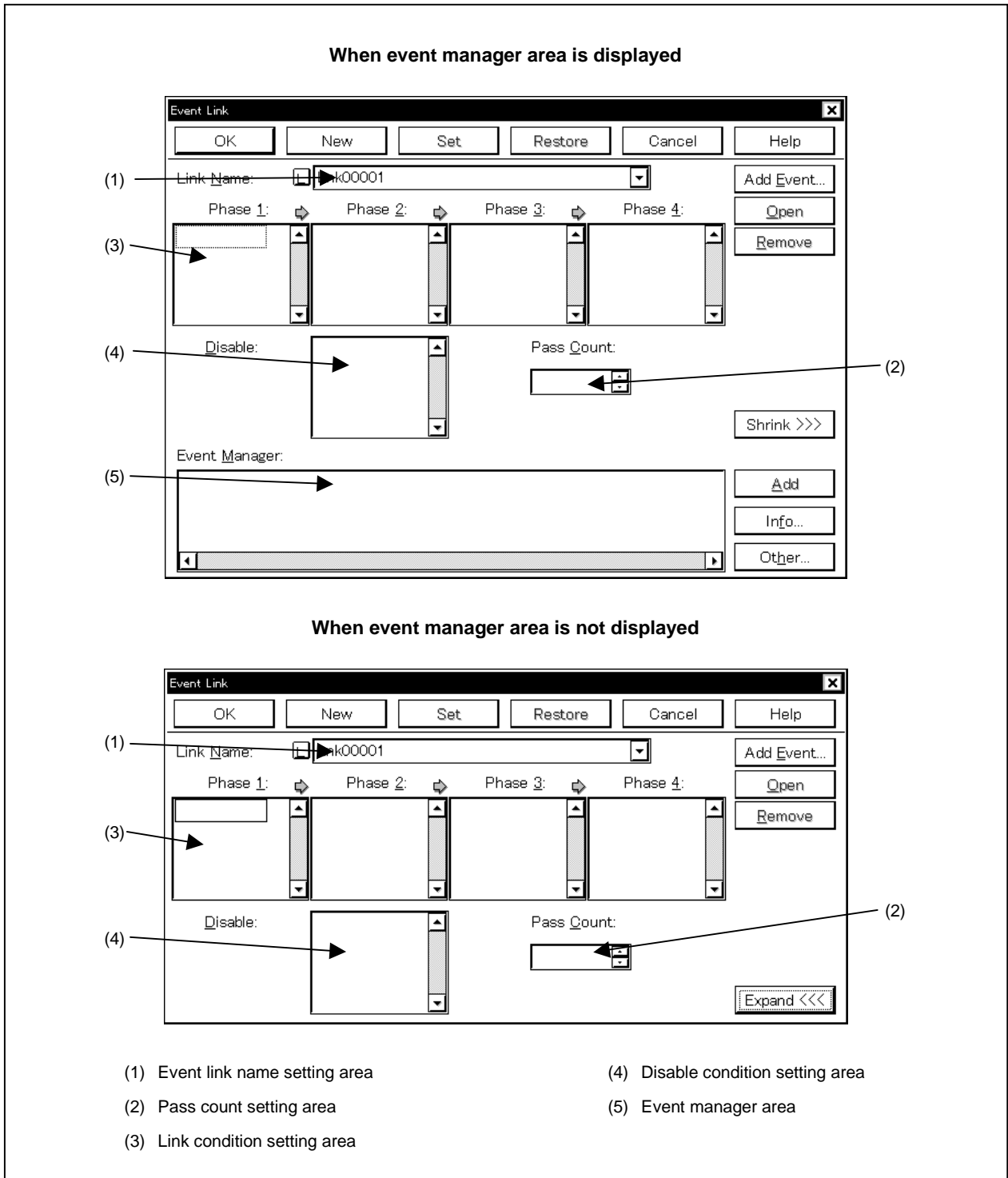
This dialog box can be opened in the following methods.

- In the main window  
Select [Event] → [EventLink...] from the menu bar.  
Press the **[GRPH]** + **[N]**, and **[L]** keys in that order.
- In the Event dialog box  
Click the **Event Link...** button.  
Press the shortcut keys **[GRPH]** + **[L]**.
- Execute one of the following operations in the Break or Trace dialog box.  
Click the **Add Link...** button.  
Press the shortcut keys **[GRPH]** + **[L]**.
- Execute one of the following operations in the event setting area or the event manager area of the Event, Break or Trace dialog box.  
Select the event link condition and click **Open**.  
Select the event link condition and press the shortcut keys **[GRPH]** + **[O]**.  
Double-click the event link condition.
- In the Event Manager  
Select an event link condition and click **Open**.  
Double-click the event link condition.
- In the New Event dialog box or the Set Other dialog box  
Click the **Event Link...** button.  
Press the shortcut keys **[GRPH]** + **[L]**.



Window

Figure 6-59. Event Link Dialog Box



<b>Function</b>
-----------------

This dialog box registers and displays event link conditions.

If the event link conditions are satisfied, an event occurs only if the user program is executed as the specified order of the event conditions that have been set.

Event link conditions can be set up to four stages.

If a disable condition is detected midway, the event conditions that have been satisfied until that point are initialized, and the search is started at the first event condition. Note that if the link condition and the disable condition are detected at the same time, the disable condition is given priority.

Up to 256 event link conditions can be registered. However, the number of event link conditions which can be used simultaneously in enabled break and trace events is as follows.

**Table 6-36. Number of Valid Event Link Conditions**

With/Without TEU	Maximum Valid Number
With TEU	3
Without TEU	Setting disabled

The Event Link dialog box has two modes, the normal mode and the select mode.

- Normal mode

When [Event] → [Event Link...] from the menu bar in the main window is selected and the Event Link dialog box is opened, event link conditions can be entered without restriction as to purpose. This is the normal mode.

- Select mode

When  in the Break dialog box is clicked and the Event Link dialog box is opened, the event conditions used by these are selected. This is the select mode.

In the select mode, when  is clicked, the event link condition selected in the event link name setting area of the Event Link dialog box is set in the event condition setting area of the Break dialog box.

Even in the select mode, event conditions can be entered and displayed just as in the normal mode.


Depending on the mode, the title bar display on the dialog box will differ. Refer to the section entitled  described later on.

The Event Link dialog box consists of the following areas.

- Event link name setting area
- Pass count setting area
- Link condition setting area
- Disable condition setting area
- Event manager area

The function of each area is explained below.

**(1) Event link name setting area**

Link Name:   

Lnk00001

Lnk00002

Lnk00003

Lnk00004

This area sets event link names and selects event link conditions.

When entering an event link condition, it is necessary to set an event link name. An event link name may consist of up to eight alphanumeric characters.

When displaying event link conditions which have already been created, either type the event link name of the already created event link condition in the text box or select the event link condition from the drop-down list in the event link name setting area.

In the select mode, the selected event link condition can be set in the event link condition setting area of the original setting dialog box that called the Event Link dialog box.

The  mark on the left side of the event link name setting area shows the use conditions of the event link condition.


The relationship between the color of the character in the  mark and the status is shown below.

**Table 6-37. Meanings of Event Icons (Event Link Condition)**

Color of Character in <input type="checkbox"/> Mark	Condition
Red (valid)	That event link condition is being used by break and indicates that it is enabled.
Black (invalid)	Indicates that that event link condition has not been enabled.
Gray (not entered)	Indicates that that event link condition is currently being edited and that it has not been entered.
Yellow (pending)	Indicates that that event is an event which is in the hold state. An event in the hold state is in a state where the symbol specified by the event condition by program download, etc. cannot be referred to.

**(2) Pass count setting area**

Pass Count:

This area sets a pass count condition. The range which can be set is as follows.

Setting range:  $1 \leq \text{pass count} \leq 32,767$

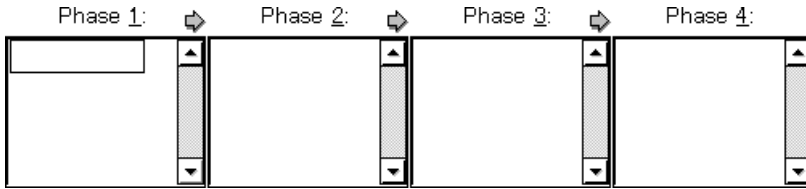
A pass count condition specifies how many times the event link condition set in this dialog box must be matched while the user program is executed, in order that a given condition is satisfied.

If 1 is set as the pass count, the condition is satisfied when each condition is matched. This area is valid only for CPU cores with the TEU incorporated.

The pass count is explained below.

- When inputting a value directly to the text box, input a decimal character string.
- If the pass count condition is omitted, it is taken that 1 is set.
- With a newly created event link condition, the pass count becomes 1.

**(3) Link condition setting area**



This area sets the sequence of event detection as a link condition. Event conditions are set in three steps, from Phase 1 to Phase 4. Event conditions need not be set up to Phase 4. If Phase 4 is not set, an event occurs when the event condition set to the last Phase is detected. Event conditions can be set in only Phase 1, and same event conditions can be set in two or more Phases. The number of event conditions that can be set to each Phase in the link condition setting area is as follows.

**Table 6-38. Number of Events That Can Be Set in Link Condition Setting Area**

Type	Number of Events That Can Be Set
Event condition (with TEU)	Up to 6 for each phase
Event condition (without TEU)	Up to 1 for each Phase 1 and Phase 2 <sup>Note</sup> Setting disabled for Phase 3 and Phase 4

**Note** Can be used only for a break condition.

To set an event condition, use one of the following methods.

- Click  and open the Event dialog box in the “Select mode”, then select the event conditions to be set.
- In the event manager area, select one (or more than one) event condition, then click  or set it by a drag-and-drop operation.  
Setting of an event condition by a drag-and-drop operation is possible even from outside the Event Link dialog box’s event manager area; event conditions can be set from the event manager area of each setting dialog box and from the Event Manager.

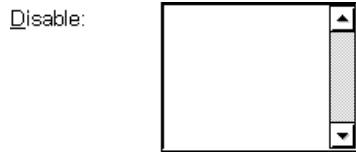
When the focus is on each Phase of the link condition setting area, click  and the event condition selected by the Phase which caused the focus will be deleted.

It can also be deleted by pressing the  key.

Each Phase of the link condition setting area and the event conditions in the disable condition setting area can be copied mutually or moved by a drag-and-drop operation. The contents are shown below.

- If an event condition is dropped only by the mouse, it is moved.
- If it is dropped while pressing the  key, the event condition is moved.
- If it is dropped while pressing the  key, the event condition is copied.

Selecting only one event condition in the link condition setting area and clicking  or double-clicking it opens the Event dialog box and displays the contents of the event condition.

**(4) Disable condition setting area**

This area is used to specify an event condition that invalidates the event conditions that have been established until that point.

The number of event conditions that can be set in the disable condition setting area is as follows.

**Table 6-39. Number of Events That Can Be Set in Disable Condition Setting Area**

Type	Number of Events That Can Be Set
Event condition (with TEU)	Up to 6
Event condition (without TEU)	Setting disabled

To set a disable condition, use one of the following methods.

- Click  and open the Event dialog box in the “Select mode”, then select the event conditions to be set.
- Select an event condition in the event manager area and click , or make the setting by a drag-and-drop operation.

When setting the event condition by a drag-and-drop operation, besides the event manager area in the Event Link dialog box, it is possible to set it from the event manager area and Event Manager in each setting dialog box.

When the focus is on the disable condition setting area, if  is clicked, the selected event condition is deleted. It can also be deleted by pressing the  key.

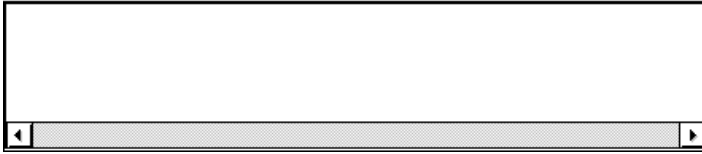
It is possible to copy and move event conditions in each Phase in the disable condition setting area and link condition setting area by a drag-and-drop operation. The contents of this operation are shown below.

- If an item is dropped with a mouse only, the event condition is moved.
- If it is dropped while holding down the  key, the event condition is moved.
- If it is dropped while holding down the  key, the event condition is copied.

If only one event condition in the disable condition setting area is selected and  is clicked or the item is double-clicked, the Event dialog box is opened and the event condition contents are displayed.

**(5) Event manager area**

Event Manager:



This displays a list of each of the entered events, event links, breaks, and traces events.

An event condition that is displayed in the event manager area can be set in the link condition setting area or disable condition setting area by clicking .

By dragging and dropping it from the event manager area in the Event Link dialog box, it can be set in the event setting area in each setting dialog box for break, and trace.

When the focus is on the event manager area, select the event icon, then click , or press the  key and the selected event can be deleted.

Also, when the focus is on the event manager area, you can select one event icon, then click , or double-click the event icon to open the setting dialog box corresponding to the selected event and display the event's setting contents.

By clicking , it is possible to select the event manager area display mode or sequencing.

**Function buttons**


(Normal mode)

Closes the Event Link dialog box.

If there is an event link condition that is being edited, it is entered automatically and the dialog box is closed.

(Select mode)

Sets the displayed event as the event link condition of the original dialog box that called the Event Link dialog box.

If the Event Link dialog box is already open, the select mode returns to the normal mode only and the Event Link dialog box does not close. In other cases, it closes. If there is an event link condition being edited, enter/select is performed automatically and the program returns to the original setting dialog box.

Newly creates an event link condition.

An event name is generated automatically and a newly created event link condition is prepared. After it is created, be sure to enter the event link condition using the  or the  button.

Enters an event link condition.

When a new event link condition has been created or when the contents of an event link condition have been changed, be sure to enter the event link condition using this button. The entered event link condition is reflected in the Event Manager.

If event link conditions that have already been entered are displayed, they cannot be selected.

<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 10px;">Clear</div> 	<p>Clears the contents of an event link condition.</p> <p>If an event link condition is entered but not edited, this button is displayed in place of the <div style="border: 1px solid black; padding: 2px; display: inline-block;">Restore</div> button.</p>
<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 10px;">Restore</div> 	<p>Returns the contents of an event link condition to the original contents.</p> <p>If an event link condition that has not been entered is displayed, all the area is made blank except the event link name, or the settings return to the default settings.</p> <p>If an event link condition is being edited, this button is displayed in place of the <div style="border: 1px solid black; padding: 2px; display: inline-block;">Clear</div> button.</p>
<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 10px;">Close</div> (Normal mode)	<p>Closes the Event Link dialog box.</p> <p>If an event link condition has been created/changed/deleted, after that, the <div style="border: 1px solid black; padding: 2px; display: inline-block;">Cancel</div> button becomes this button.</p> <p>Even if there is an event link condition that is being edited, the dialog box closes without the event link condition being entered. The <div style="border: 1px solid black; padding: 2px; display: inline-block;">ESC</div> key has the same operation.</p>
<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 10px;">Cancel</div> (Normal mode)	<p>Closes the Event Link dialog box.</p> <p>If an event link condition is not being created/changed/deleted, this button becomes the <div style="border: 1px solid black; padding: 2px; display: inline-block;">Close</div> button.</p> <p>Even if an event link condition is being edited, the dialog box closes without the event link condition being entered. The <div style="border: 1px solid black; padding: 2px; display: inline-block;">ESC</div> key has the same operation.</p>
<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 10px;">Cancel</div> (Select mode)	<p>Returns to the original setting dialog box that called the Event Link dialog box without an event link condition being selected.</p> <p>If an event link condition is not being created/changed/deleted, the <div style="border: 1px solid black; padding: 2px; display: inline-block;">Abort</div> button becomes this button.</p> <p>If the Event Link dialog box is already open, the select mode returns to the normal mode only, and the Event Link dialog box does not close. In other cases, the Event Link dialog box closes.</p> <p>Even if an event link condition is being edited, it is not entered. The <div style="border: 1px solid black; padding: 2px; display: inline-block;">ESC</div> key has the same operation.</p>
<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 10px;">Abort</div> (Normal mode)	<p>Returns to the original setting dialog box without selecting the event link condition.</p> <p>If an event link condition has been created/changed/deleted, the <div style="border: 1px solid black; padding: 2px; display: inline-block;">Cancel</div> button becomes this button.</p> <p>If the Event Link dialog box is already open, the select mode returns to the normal mode only, and the Event Link dialog box does not close. In other cases, the Event Link dialog box closes.</p> <p>Even if an event link condition is being edited, it is not entered. The <div style="border: 1px solid black; padding: 2px; display: inline-block;">ESC</div> key has the same operation.</p>
<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 10px;">Help</div>	<p>Opens the help window.</p>

Add Event...

Opens the select mode in the Event dialog box and makes additional settings in the link condition setting area or disable condition setting area. The area where additional settings are made becomes the selected area when **Add Event...** is clicked.

Open

Opens each setting dialog box for the event selected in the link condition setting area, disable condition setting area or event manager area and displays the contents.

If an event is not selected, or if multiple events are selected, selection cannot be made.

Operation is the same as double-clicking the event icon.

Remove

Deletes the event selected in the link condition setting area, disable condition setting area or event manager area.

The same operation is performed by pressing the **DEL** key.

Expand &lt;&lt;&lt;

Displays the event manager area. Expands the size of the dialog box.

When the event manager area is not displayed, this button is displayed in place of the **Shrink >>>** button.

Shrink &gt;&gt;&gt;

Does not display the event manager area. Shrinks the size of the dialog box.

When the event manager area is displayed, this button is displayed in place of the **Expand <<<** button.

Add

Adds an event condition or event link condition selected in the event manager area to the link condition setting area or disable condition setting area and sets it. The area where it is added becomes the selected area when **Add** is clicked.

Info...

Opens the Event Info dialog box.

The **Sort by Name** button rearranges events in name order and displays them.

The **Sort by Kind** button rearranges the events in order by kind and displays them.

The **Unsort** button displays the events in the order in which they were entered without rearranging them.

The **Detail** button changes the event manager area to the detailed display mode.

The **Overview** button changes the event manager area to the list display mode.

The **Cancel** button or the **ESC** key closes the dialog box.

Other...

This opens the Set Other dialog box.

Clicking each button opens the Event Manager and the Event, Break, or Trace dialog box, and closes the dialog box.

Close the dialog box by clicking **Cancel** or **Event Link...** and return to the Event Link dialog box.



**Title bar**

In the normal mode, [Event Link] is displayed as the title in the title bar.

Title bar
Event Link

In the select mode, the title in the title bar becomes as follows in accordance with the original setting dialog box that called the Event Link dialog box.

Title Bar	Original Setting Dialog Box
Event Link – Break	Break dialog box
Event Link – Trace	Trace dialog box

<b>Break Dialog Box</b>	<b>Modeless</b>
-------------------------	-----------------





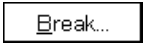
### General

This dialog box registers, sets, and displays break event conditions.

The break event conditions registered in this dialog box are automatically registered to the Event Manager.

### Opening method

This dialog box can be opened in the following methods.

- In the main window
  - Select [Event] → [Break...] from the menu bar.
  - Press the **[GRPH]** + **[N]**, and **[B]** keys in that order.
  - Click the  button on the toolbar.
- In the Event dialog box
  - Click the  button.
  - Press the shortcut keys **[GRPH]** + **[B]**.
- Execute one of the following operations in the event manager area in the Event, Event Link, or Trace dialog box.
  - Select the break event condition and click .
  - Select the break event condition and press the shortcut keys **[GRPH]** + **[O]**.
  - Double-click the break event condition.
- In the Event Manager, execute one of the following operations.
  - Select the break event condition and click .
  - Double-click the break event condition.
- Execute one of the following operations in the New Event dialog box or the Set Other dialog box.
  - Click the  button.
  - Press the shortcut keys **[GRPH]** + **[B]**.



<b>Function</b>
-----------------

This dialog box registers, sets, and displays break event conditions.

Up to 256 break event conditions can be entered. However, the number of break event conditions that can be used simultaneously is as follows.

**Table 6-40. Number of Valid Break Event Conditions**

With/Without TEU	Maximum Valid Number
With TEU	8
Without TEU	2 <sup>Note</sup>

**Note** Exclusive use of events before execution and access events and address range specification are prohibited.

Break event conditions become valid automatically when they are entered. However, if the number of break event conditions exceeds the number that can be used simultaneously, or if the number of event conditions or event link conditions in use exceeds the number that can be used simultaneously, the break event conditions are entered in the disabled state.


Note that if [Run] → [Ignore Break Point] is selected, the break event condition will not operate even if it is enabled.



The Break dialog box consists of the following areas.

- Break event name setting area
- Break condition setting area
- Event manager area

The function of each area is explained below.

**(1) Break event name setting area**


Break Name:   

Brk00001	
Brk00002	
Brk00003	
Brk00004	
Brk00005	
Brk00006	

This area sets break event names and selects break event conditions.


When entering a break event condition, it is necessary to set a break event name. The break event name that can be set must be of 8 alphanumeric characters or less.

When displaying break event conditions which have already been created, either type the break event name of the already created break event condition in the text box or select the break event condition from the drop-down list in the break event name setting area.

Clicking the  mark in the break event name setting area switches between enabling or disabling the break event condition.

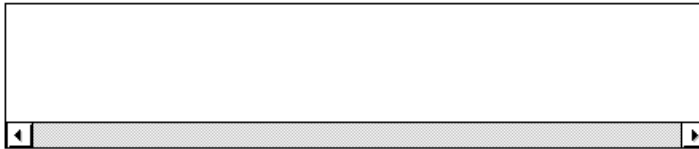
The relationship between the color of the character in the  mark and the status is shown below.

**Table 6-41. Meanings of Event Icons (Break Event Condition)**

Color of Character in  Mark	Condition
Red	Indicates that the break event condition has been enabled. A break event occurs if the condition is satisfied.
Black	Indicates that the break event condition has not been enabled. No break event occurs if the condition is satisfied.
Gray	Indicates that the break event condition is currently being edited and that it has not been entered.
Yellow	Indicates that the break event is an event which is in the hold state. An event in the hold state is in a state where the symbol specified by the event condition by program download, etc. cannot be referred to.

**(2) Break condition setting area**

Break Event:



This area sets an event condition for a break.




The number of event conditions that can be set in the break condition setting area is as follows.

**Table 6-42. Number of Events That Can Be Set in Break Condition Setting Area**

Type	Number of Events That Can Be Set
Execution condition (with TEU)	8
Event condition (without TEU)	2 <sup>Note</sup>
Event link condition	1

**Note** Exclusive use of events before execution and access events.

To set an event condition, use one of the following methods.

- Click  and open the Event dialog box in the select mode, then select the event condition which is to be set.
- Click  and open the Event Link dialog box in the select mode, then select the event link condition which is to be set.
- Select an event link condition in the event manager area and click , or make the setting by a drag-and-drop operation.

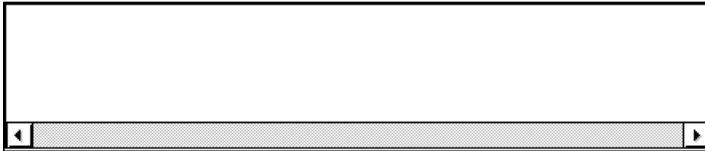
When setting the event condition or event link condition by a drag-and-drop operation, besides the event manager area in the Break dialog box, it is possible to set it from the event manager area and Event Manager in each setting dialog box.

When the focus is on the break condition setting area, clicking  deletes the selected event condition or event link condition. It can also be deleted by  key.

If only one event condition or event link condition in the break condition setting area is selected and  is clicked or the item is double-clicked, the Event dialog box or Event Link dialog box is opened and the event condition or event link condition contents are displayed.

### (3) Event manager area

Event Manager:



This area displays a list of each of the entered events, event links, breaks, and traces events.

An event or event link condition that is displayed in the event manager area can be set in the break condition setting area by clicking .

Also, event conditions or event link conditions in the event manager area can be dragged and dropped directly to the break condition setting area.

When the focus is on the event manager area, select the event icon, then click , or press the  key and the selected event can be deleted.

Also, when the focus is on the event manager area, you can select one event icon, then click , or double-click the event icon to open the setting dialog box corresponding to the selected event and display the event's setting contents.

By clicking , it is possible to select the event manager area display mode or sequencing.

#### Function buttons

Closes the Break dialog box.

If there is a break event condition that is being edited, it is entered automatically and the dialog box is closed.

Break event conditions are enabled as soon as it is entered.

Newly creates a break event condition.

A break event name is generated automatically and a newly created break event condition is prepared. After it is created, be sure to enter the break event condition using the  or the  button.

Enters the break event condition.

When a new break event condition is created or when the contents of a break event condition have been changed, be sure to enter the break event condition using this button.

If a break event condition that has already been entered is displayed, this button changes to either the  button or the  button.

Enable	<p>Enables a break event condition.</p> <p>This button is displayed in cases where a break event condition is in the disabled state.</p> <p>It changes to either the <input data-bbox="850 268 989 300" type="button" value="Set"/> button or the <input data-bbox="1138 268 1263 300" type="button" value="Disable"/> button.</p>
Disable	<p>Disables a break event condition.</p> <p>This button is displayed in cases where a break event condition is in the enabled state.</p> <p>It changes to either the <input data-bbox="850 459 989 491" type="button" value="Set"/> button or the <input data-bbox="1138 459 1263 491" type="button" value="Enable"/> button.</p>
Clear	<p>Clears the contents of a break event condition.</p> <p>If a break event condition is entered but not edited, this button is displayed. It changes to the <input data-bbox="764 619 902 651" type="button" value="Restore"/> button.</p>
Restore	<p>Restores the break event condition to its original contents.</p> <p>If a break event condition that has not been entered is displayed, the display area is restored to the blank state except for the break event name.</p> <p>This button is displayed in the case where a break event condition is being edited. It changes to the <input data-bbox="786 852 924 884" type="button" value="Clear"/> button.</p>
Cancel	<p>Closes this dialog box.</p> <p>If a break event condition is not being created/changed/deleted, the <input data-bbox="1354 970 1474 1001" type="button" value="Close"/> button becomes this button.</p> <p>Even if a break event condition is being edited, this dialog box is closed without entering the condition. The <input data-bbox="899 1085 964 1117" type="button" value="ESC"/> key has the same operation.</p>
Close	<p>Closes this dialog box.</p> <p>If a break event condition has being created/changed/deleted, the <input data-bbox="1330 1203 1474 1234" type="button" value="Cancel"/> button becomes this button.</p> <p>Even if a break event condition is being edited, this dialog box is closed without entering the condition. The <input data-bbox="899 1318 964 1350" type="button" value="ESC"/> key has the same operation.</p>
Help	<p>Opens the help window.</p>
Add Event...	<p>Opens the Event dialog box in the select mode and selects the set event condition.</p>
Add Link...	<p>Opens the Event Link dialog box in the select mode and selects the set event link condition.</p>
Open	<p>Displays the contents of an event selected in the break condition setting area or the event manager area by opening each dialog box. When the event is not selected, or if multiple events have been selected, this button is disabled. Its operation is the same as double-clicking the event icon.</p>

Remove

Deletes an event selected in the break condition setting area or the event manager area.

Pressing the **DEL** key performs the same operation.

Expand &lt;&lt;&lt;

Displays the event manager area. Expands the size of the dialog box.

When the event manager area is not displayed, this button is displayed in place of the **Shrink >>>** button.

Shrink &gt;&gt;&gt;

Does not display the event manager area. Shrinks the size of the dialog box.

When the event manager area is displayed, this button is displayed in place of the **Expand <<<** button.

Add

Adds the event condition or event link condition selected in the event manager area to the break condition setting area. The area where the event condition is added is the selected area when **Add** button is clicked.

An event other than an event condition or an event link condition cannot be added.

Info...

Opens the Event Info dialog box.

The **Sort by Name** button rearranges events in name order and displays them.

The **Sort by Kind** button rearranges the events in order by kind and displays them.

The **Unsort** button displays the events in the order in which they were entered without rearranging them.

The **Detail** button changes the event manager area to the detailed display mode.

The **Overview** button changes the event manager area to the list display mode.

The **Cancel** button or the **ESC** key closes the dialog box.

Other...

Opens the Set Other dialog box.

Clicking each button opens the Event Manager and the Event, Event Link, or Trace dialog box, and closes the dialog box.

Close the dialog box by clicking **Cancel** or **Break...** and return to the Break dialog box.



<b>Trace Dialog Box</b>	<b>Modeless</b>
-------------------------	-----------------

### General






This dialog box registers, sets, and displays trace event conditions.

The trace event conditions registered in this dialog box are automatically registered to the Event Manager.

This dialog box is valid only for CPU cores with the TRCU incorporated.

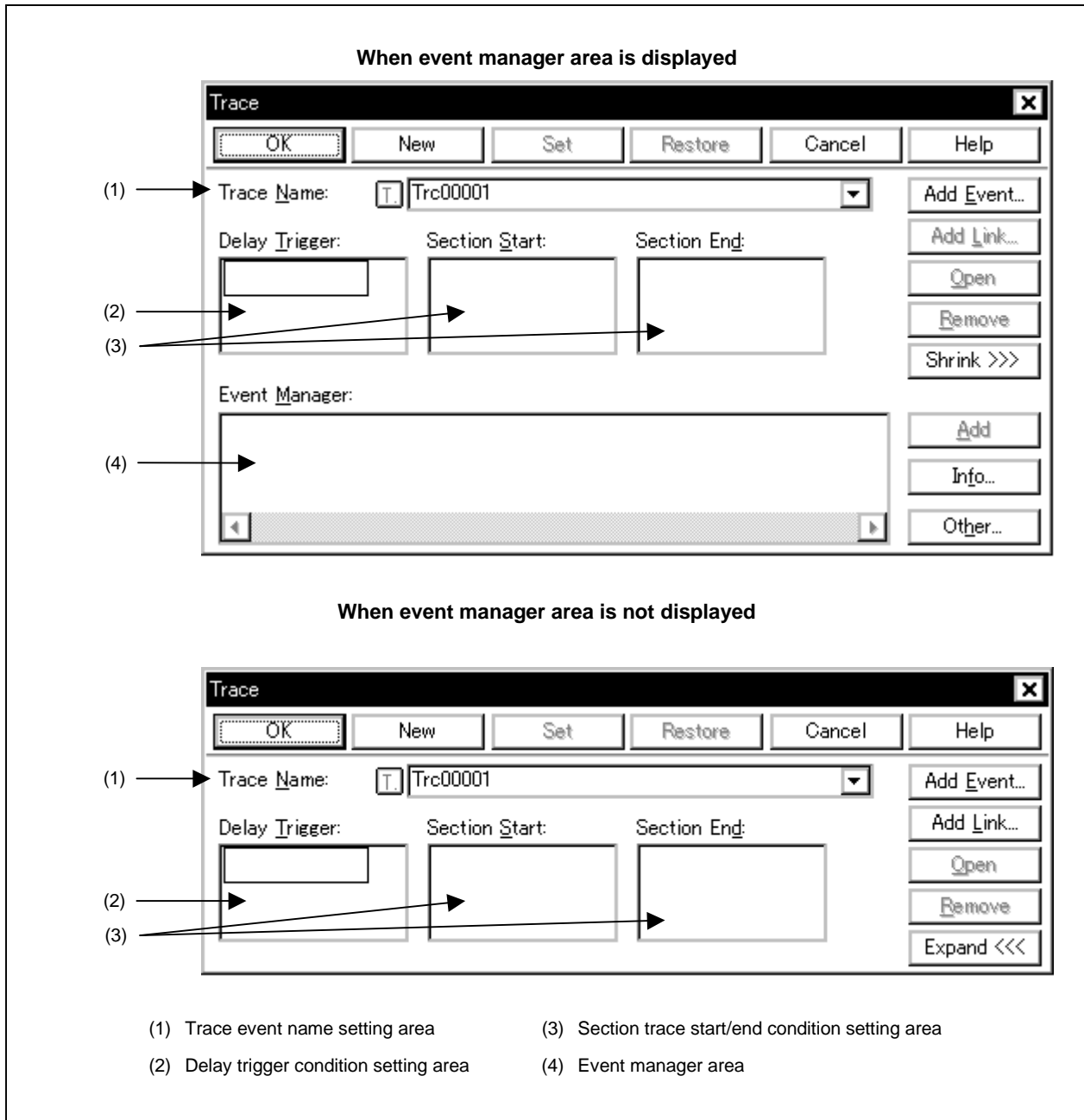
### Opening method

This dialog box can be opened in the following methods.

- In the main window
  - Select [Event] → [Trace...] from the menu bar.
  - Press the **GRPH** + **N**, and **T** keys in that order.
  - Click the  button on the toolbar.
- In the Event dialog box
  - Click the  button.
  - Press the shortcut keys **GRPH** + **T**.
- Execute one of the following operations in the event manager area in the Event, Event Link, or Break dialog box.
  - Select the trace event condition and click the  button.
  - Select the trace event condition and press the shortcut keys **GRPH** + **O**.
  - Double-click the trace event condition.
- Execute one of the following operations in the Event Manager.
  - Select the trace event condition and click the  button.
  - Double-click the trace event condition.
- Execute one of the following operations in the New Event dialog box or Set Other dialog box.
  - Click the  button.
  - Press the shortcut keys **GRPH** + **T**.

## Window

Figure 6-61. Trace Dialog Box



## Function

This dialog box registers, sets, and displays trace event conditions.  
 The following 2 types of condition traces can be set.

- Section trace
- Delay trace

Up to 256 trace event conditions can be entered. However, the number of trace event conditions that can be used simultaneously is 1.

Trace event conditions become valid automatically when they are entered. However, if the enabled trace event conditions exist, or if the number of event conditions in use exceeds the number that can be used simultaneously, the trace event conditions are entered in the disabled state. Also, in this case, a trace event condition cannot be enabled.

A trace is performed only when the tracer is in the enabled state. Set it by the procedure shown below.

- <1> Select [Option] → [Tracer ON] from the menu bar in the main window and enable the tracer.
- <2> When performing a conditional trace, select [Run] → [Cond. Trace ON], and in other cases, select [Run] → [Uncond. Trace ON].

While the user program is executed, in order to stop the currently operating tracer, select [Run] → [Tracer Stop] from the menu bar. Also, during program execution, in order to run the currently stopped tracer, select [Run] → [Tracer Start] from the menu bar. These two menus toggle and their display changes according to the current tracer state.

One of the following can be selected as the tracer control mode. Select [Run] menu → [Tracer Control Mode].

- [Non Stop]                      The total trace memory is traced and then overwritten from the oldest frame (default).
- [Full Stop]                     The total trace memory is traced and then the tracer is stopped.
- [Delay Trigger Stop]         The tracer is stopped after tracing for the delay count when a delay trigger event occurs.

This dialog box consists of the following areas.

- Trace event name setting area
- Delay trigger condition setting area
- Section trace start/end condition setting area
- Event manager area

Multiple event conditions can be set to the delay trigger condition setting area and section trace start/end condition setting area. However, the total number of event conditions that can be set to the delay trigger condition setting area and section trace start/end condition setting area is as follows.

**Table 6-43. Total Number of Events That Can Be Set to Each Area in Trace Dialog Box**

Type	Number of Events That Can Be Set
Execution condition (with TEU)	6 <sup>Note</sup>
Event condition (without TEU)	Setting disabled
Event link condition	Setting disabled

**Note** As long as the number of events being used does not exceed the number of events that can be used simultaneously.

The function of each area is explained below.

**(1) Trace event name setting area**

Trace Name:

This area sets trace event names and selects trace event conditions.

When entering a trace event condition, it is necessary to set a trace event name. A trace event name that can be set must be of 8 alphanumeric characters or less.

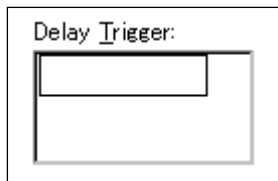
When displaying trace event conditions which have already been created, either type the trace event name of the already created trace event condition in the text box or select the drop-down list in the trace event name setting area.

Clicking the  mark in the trace event name setting area switches between enabling or disabling the trace event condition.

The relationship between the color of the character in the  mark and the status is shown below.

**Table 6-44. Meanings of Event Icons (Trace Event Condition)**

Color of Character in <input type="checkbox"/> Mark	Condition
Red	Indicates that the trace event condition has been enabled. A trace event occurs if the condition is satisfied.
Black	Indicates that the trace event condition has not been enabled. No trace event occurs if the condition is satisfied.
Gray	Indicates that the trace event condition is currently being edited and that it has not been entered.
Yellow	Indicates that the trace event is an event which is in the hold state. An event in the hold state is in a state where the symbol specified by the event condition by program download, etc. cannot be referred to.

**(2) Delay trigger condition setting area**

This area sets an event condition for delay trigger.

The delay count is set in the Delay Count Setting dialog box.

**(3) Section trace start/end condition setting area**

Section Start:      Section End:



This area sets an event condition to start section trace to Section Start: and an event condition to end section trace to Section End:.

To set a delay trigger condition and section trace start/end condition, use one of the following methods.

- Click  and open the Event dialog box in the select mode, then select the event condition which is to be set.
- Select one or more event condition in the event manager area and click , or make the setting by a drag-and-drop operation.  
When setting the event condition by a drag-and-drop operation, besides the event manager area in the Trace dialog box, it is possible to set it from the event manager area and Event Manager in each setting dialog box.

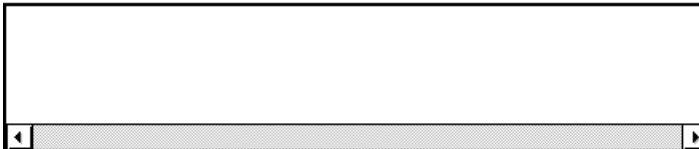
When the focus is on the condition setting area, click the  button, or press the  key and the selected event condition can be deleted.

Selecting an event condition in the condition setting area and clicking  or double-clicking it opens the Event dialog box and displays the contents of the event condition.

The condition setting area is left blank when there is a newly created trace event condition.

**(4) Event manager area**

Event Manager:



In this area a list of each event, event link, break, and trace event is displayed.

By selecting an event condition displayed in the event manager area and clicking , the event condition can be set in each condition setting area.

Also the event condition in the event manager area can be directly dragged and dropped in each event condition setting area.

When the focus is on the event manager area, select an event icon, then click the  button, or press the  key and the selected event can be deleted.

When the focus is on the event manager area, selecting an event icon and clicking  or double-clicking an event icon opens the setting dialog box corresponding to the selected event, and displays the contents of the settings of the event.

By clicking , it is possible to select the event manager area display mode or sequencing.

**Function buttons**

OK

Closes the Trace dialog box.

If there is a trace event condition that is being edited, it is entered automatically and the dialog box is closed.

The trace event condition becomes enabled as soon as it is entered.

New

Newly creates a trace event condition.

A trace event name is generated automatically and a newly created trace event condition is prepared. After it is created, be sure to enter the trace event condition using the  or the  button.

Set

Enters a trace event condition.

When a new trace event condition has been created or when the contents of a trace event condition have been changed, be sure to enter the trace event condition using this button.

If trace event conditions that have already been entered are displayed, it changes to either the  button or  button.

Enable

Enables a trace event condition.

This button is displayed in cases where a trace event condition is in the disabled state.

It changes to either the  button or the  button.

Disable

Disables a trace event condition.

This button is displayed in cases where a trace event condition is in the enabled state.

It changes to either the  button or the  button.

Clear

Clears the contents of a trace event condition.

If a trace event condition is entered but not edited, this button is displayed. It changes to the  button.

Restore

Returns the contents of a trace event condition to the original contents.

If a trace event condition that has not been entered is displayed, all the area is made blank except the trace event name.

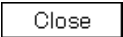
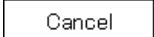

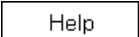



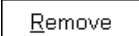

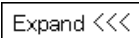
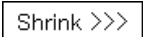
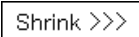
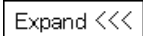


If a trace event condition is being edited, this button is displayed in place of the  button.

Cancel

Closes the Trace dialog box.

If a trace event condition is not being created/changed/deleted, this button becomes the  button.

Even if a trace event condition is being edited, the dialog box closes without the trace event condition being entered. The  key has the same operation.

	<p>Closes the Trace dialog box.</p> <p>If a trace event condition has been created/changed/deleted, after that, the  button becomes this button.</p> <p>Even if there is a trace event condition that is being edited, the dialog box closes without the trace event condition being entered. The  key has the same operation.</p>
	<p>Opens the help window.</p>
	<p>Opens the Event dialog box in the select mode and selects an event condition to be set.</p>
	<p>Opens the Event Link dialog box in the select mode and selects an event link condition to be set.</p>
	<p>Displays the event contents selected in each condition setting area or event manager area by opening each setting dialog box.</p> <p>When the event is not selected, or if multiple events have been selected, this button is disabled. Its operation is the same as double-clicking the event icon.</p>
	<p>Deletes the event selected in each condition setting area or event manager area. The  key has the same operation.</p>
	<p>Displays the event manager area. Expands the size of the dialog box.</p> <p>When the event manager area is not displayed, this button is displayed in place of the  button.</p>
	<p>Does not display the event manager area. Shrinks the size of the dialog box.</p> <p>When the event manager area is displayed, this button is displayed in place of the  button.</p>
	<p>Adds the event condition or event link condition selected in the event manager area to each condition setting area.</p> <p>The area where the event condition is added is the selected area when  button is clicked.</p> <p>An event other than an event condition or an event link condition cannot be added.</p>

## Info...

Opens the Event Info dialog box.

The  button rearranges events in name order and displays them.

The  button rearranges the events in order by kind and displays them.

The  button displays the events in the order in which they were entered without rearranging them.

The  button changes the event manager area to the detailed display mode.

The  button changes the event manager area to the list display mode.

The  button or the  key closes the dialog box.

## Other...

Opens the Set Other dialog box.

Clicking each button opens the Event, Event Link, or Break dialog box and the Event Manager, and closes the Set Other dialog box.

Close the dialog box by clicking  or  and return to the Trace dialog box.



**Delay Count Setting Dialog Box****Modeless****General**

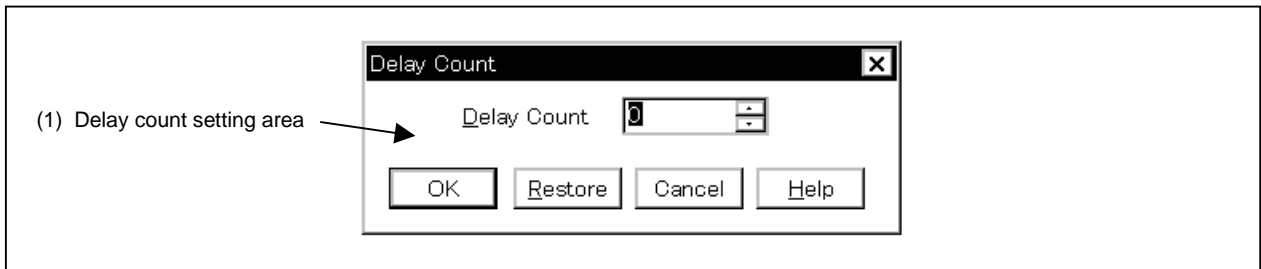
This dialog box sets and displays the delay count to perform trace.

This dialog box is valid only for CPU cores with the TRCU incorporated.

**Opening method**

This dialog box can be opened in the following methods.

- In the main window  
Select [Event] → [Delay Count] from the menu bar.  
Press the shortcut keys **[GRPH] + [F]** and **[Y]**.

**Window****Figure 6-62. Delay Count Setting Dialog Box****Function**

This dialog box sets and displays delay count.

Setting the delay count enables a trace to be performed for specified delay count after the delay trigger event condition set in the Trace dialog box is satisfied.

The Delay Count Setting dialog box consists of the following area.

- Delay count setting area

The function of this area is explained below.

**(1) Delay count setting area**

This area sets a delay count (in frame units). The settable range is from 0 (default) to 32767.

**Function buttons**

OK

Closes the Delay Count Setting dialog box after setting the edited delay count.

Restore

Returns the delay count to the previous value.

Cancel

Closes the Delay Count Setting dialog box.

Help

Opens the help window.

<b>Reset Debugger Dialog Box</b>	<b>Modal</b>
----------------------------------	--------------

### General

This dialog box initializes the debugger, symbol information, and CPU.

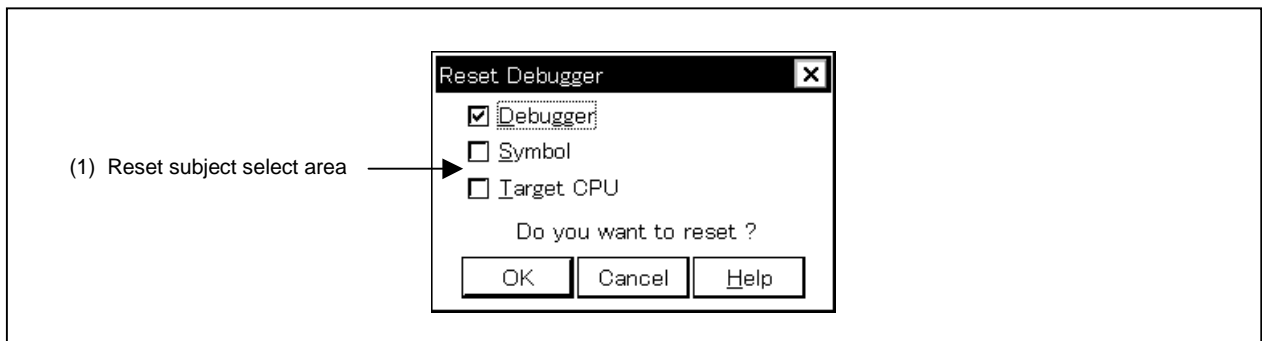
### Opening method

This dialog box can be opened in the following methods.

- In the main window  
Select [File] → [Debugger Reset...] from the menu bar.  
Press the [GRPH] + [F], and [R] keys in that order.

### Window

**Figure 6-63. Reset Debugger Dialog Box**



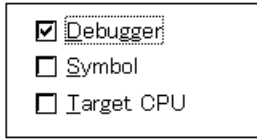
### Function

This dialog box specifies the subject to be initialized using a check box. As the default, only the debugger is initialized.

The Reset Debugger dialog box consists of the following area.

- Reset subject select area

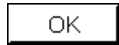
The function of this area is explained below.

**(1) Reset subject select area**

Debugger  
 Symbol  
 Target CPU

This area selects the subject to be reset.

Debugger:            Initializes the debugger (default).  
Symbol:             Resets symbol information.  
Target CPU:         Resets CPU.

**Function buttons**

Initializes the selected subject.



Ignores the setting and closes this dialog box.



Opens the help window.

<b>About Dialog Box</b>	<b>Modal</b>
-------------------------	--------------

### General

This dialog box displays the version information of the debugger.

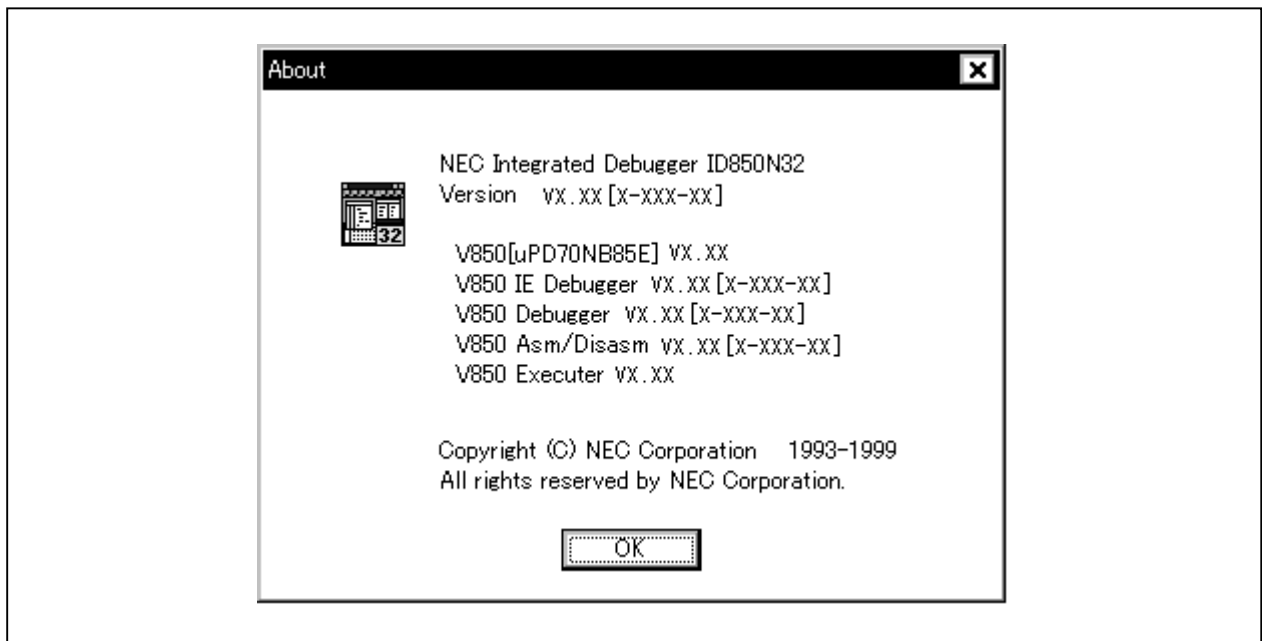
### Opening method

This dialog box can be opened in the following methods.

- In the main window  
Select [H]elp → [A]bout... from the menu bar.  
Press the [GRPH] + [H], and [A] keys in that order.

### Window

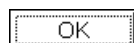
**Figure 6-64. About Dialog Box**



### Function

This dialog box displays the version of the debugger and the version information of the device files, GUI, project manager tool DLLs, debugger DLLs, assembler DLLs, executor, and monitor.

### Function button



Closes the About dialog box.

<b>Exit Debugger Dialog Box</b>	<b>Modal</b>
---------------------------------	--------------

### General

This dialog box terminates the debugger.

When terminating the debugger, the debugging environments can be saved in a project file.

### Opening method

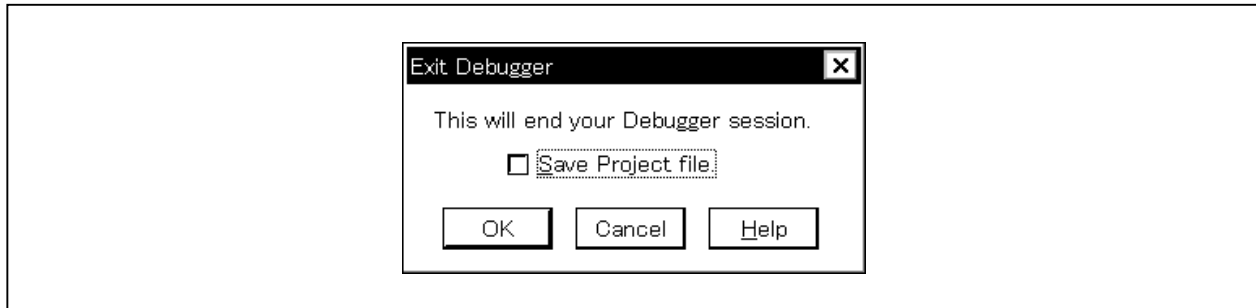
This dialog box can be opened in the following methods.

- In the main window  
Select [File] → [Exit] from the menu bar.  
Press the [GRPH] + [F], and [X] keys in that order.

Also, the Exit Debugger dialog box opens when a compulsory close is executed, such as when Windows is closed or when an application is closed in the task list.

### Window

**Figure 6-65. Exit Debugger Dialog Box**



### Function

This dialog box specifies, by using a check box, whether the current debugging environments are saved in a project file or not when the debugger is terminated. As the default, the debugging environments are not saved in a project file. If the project file has been saved or loaded during a debugging operation, the default is to save the debugging environments.

When the check box is checked and the **OK** button is clicked, the Project File Save dialog box is opened, the current debugging environments are saved in the project file, and all the windows are closed. The debugger is then terminated.

If the project file has been saved or loaded during a debugging operation, however, the Project File Save dialog box is not opened, and the debugging environments are saved in the project file.

If the **OK** button is clicked without checking the check box, all the windows are closed, and the debugger is terminated.

Note that the Exit Debugger dialog box can be set not to open in the Debugger Option dialog box.

**Function buttons****OK**

When the check box is checked, the Project File Save dialog box is opened, the current debugging environments are saved in the project file, and all the windows are closed. The debugger is then terminated. When the check box is not checked, all the windows are closed, and the debugger is terminated.

**Cancel**

Closes this dialog box without performing anything.

**H**elp

Opens the help window.

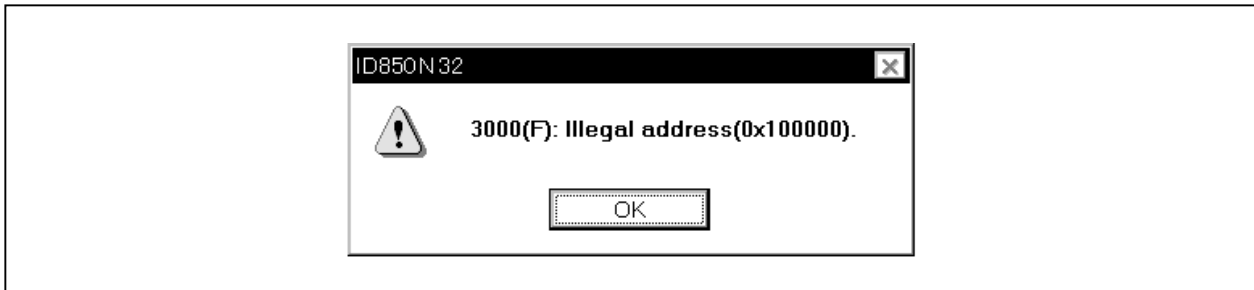
<b>Error/Warning Dialog Box</b>	<b>Display dialog (Modal)</b>
---------------------------------	-------------------------------

**General**

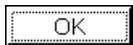
This dialog box is opened if an error or a warning occurs, to inform the user to check the message.

**Window**

**Figure 6-66. Error/Warning Dialog Box**

**Function**

Contents of the error or warning are displayed in the message display area. Refer to the list of error messages, if an error occurs.

**Function button**

Closes this dialog box.



## APPENDIX A DEBUGGING WITH ID850NW

This appendix explains the basic debugging operation using the ID850NW.

For the details of each window and dialog box mentioned in this section, refer to **CHAPTER 6 WINDOW REFERENCE**.

### (1) Setting debug environment

Set the operating environment of debugger.

If a previous debug environment has been saved to the project file, the debug environment can be restored by loading that file. The procedure differs as follows depending on whether the project file is used or not.

#### (a) When project file is not used

- <1> Set all the items in the Configuration dialog box.
- <2> Set the extended options in the Extended Option dialog box.
- <3> Set the debug options in the Debugger Option dialog box.

**Caution** Always specify the startup symbol.

#### (b) When project file is used


- **When project file is used at startup**

- <1> Click  in the Configuration dialog box.
  - The Project File Load dialog box will open.
- <2> Specify the project file name and click .
- The project file will be loaded.

The following procedure can also be used.

- <1> Select [Run] from the Start menu in Windows.
- <2> Use the  button to specify the file to be executed, id850n32.EXE.
- <3> After id850n32.EXE, specify the name of the project file to be used (eg. A:\nectools32\bin\id850n32.EXE\B:\Test\Test1.prj (\Delta = space)).
- <4> Click .

- **When project file is used during debugging**

- <1> Select [File] → [Project] → [Open] in the main window, or click the  button.
  - The Project File Load dialog box will open.
- <2> Specify the project file name and click .
- The project file will be loaded.

### (2) Download load module file

#### (a) Download the load module file to be debugged.

→ Download dialog box

#### (b) Confirming loaded files

→ Load Module List dialog box

**(3) Display disassemble result**

Display the result of disassembling the user program downloaded.

→ Assemble window

The method of displaying the disassemble display label column and mnemonic column can be changed.

→ Debugger Option dialog box

**(4) Display source program**

Display the source program. The program whose source can be debugged must have debug information compiled with the -g option specified.

→ Source window

**(a) Setting source path information, display font, and source file extension**

→ Debugger Option dialog box

**(b) Selecting source file**

→ Browse dialog box

**(c) Source file and disassemble**

→ Select [V]iew → [M]ix when the source text file is the current window.

**(5) Setting of event****(a) Registering event conditions and event link conditions**

→ Event dialog box

Event Link dialog box

**(b) Setting of event conditions**

Set event conditions by using the event icons in the Event Manager or in the event manager area of various event setting dialog boxes (however, breakpoints can be set in the Source window/Assemble window).

- Trace event condition  
→ Trace dialog box
- Break event condition  
→ Break dialog box

**(c) Checking setting of event conditions**

→ Event Manager, the event manager area in each event setting dialog box

**(6) Checking setting of software break**

→ Software Break Manager

**(7) Executing user program**

By executing the user program, the value of the PC advances to the set breakpoint, break event, or forced break.

- Types of execution  
→ Execution control buttons on the toolbar on Main window, or [R]un menu

**(8) Checking trace data**

→ Trace View window

**(9) Editing, checking, and changing contents of memory**

→ Memory window, Memory Copy dialog box, Memory Fill dialog box, Memory Compare dialog box, Memory Compare Result dialog box

**(10) Registering, checking, and changing variable values**

→ Quick Watch dialog box, Add Watch dialog box, Watch window, Local Variable window

**(11) Checking contents of stack**

→ Stack window

**(12) Checking and changing of registered contents of register value**

→ Register Select dialog box, Register window, IOR window, The PC register value can also be checked in the Source and Assemble windows.

**(13) Checking and changing peripheral I/O register value**

→ IOR window, IOR Select dialog box

**(14) Registering, checking, and changing user-defined I/O port value**

→ Add I/O Port dialog box, IOR window

**(15) Changing mnemonics and executing on-line assemble**

→ Assemble window, Source window

**(16) Searching character string in each window**

→ Source Search dialog box, Assemble Search dialog box, Memory Search dialog box

**(17) Referencing “display file” saved in previous debugging process**

→ View File Load dialog box

**(18) Saving screen contents to “display file” during debugging**

→ View File Save dialog box

**(19) Uploading memory contents**

→ Upload dialog box

**(20) Saving debug environment to project file**

→ Project File Save dialog box

**(21) Terminating ID850NW**

→ Exit Debugger dialog box

[MEMO]

## APPENDIX B TERMINOLOGY

Words and window types (refer to **CHAPTER 6 WINDOW REFERENCE**) used in this document are described below.

### B.1 Terminology

The following words are described in this document.

- TEU
- TRCU
- Current window
- Delimiter symbol (separator)
- Program
- Current program
- File
- Current file
- Function
- Current function
- Structure
- Stack frame number
- Line
- Register name
- Numeric value
- Address
- Symbol
- Expression and operator
- Character set

Each word and the input conventions is explained below.

**(1) TEU and TRCU**

Both are parts of the DCU (Debug Control Unit) function.

TEU: Unit to realize the event function

TRCU: Unit to realize the trace function

The number of events that can be set is limited when the TEU is not incorporated in the CPU core.

The trace function cannot be used when the TRCU is not incorporated in the CPU core.

**(2) Current window**

The current window is the window in which an operation is currently being performed. In this window, information can be input from the keyboard and menus can be selected.

**(3) Delimiter symbol (separator)**

A file, function, variable, or line can be specified by using the delimiter symbols listed in Table B-1.

**Table B-1. Delimiter Symbols**

Symbol	Specification
#	Used as a separator between file names and variable/function names and line numbers.
\$	Used as a separator between load module names and file, variable, and function names.

**(4) Program**

A program is the executable unit that is subject to debugging. The ID850NW deals with each load module file specified when downloading as one program. The name of a program is specified by using the name of the load module file that is downloaded.

**(5) Current program**

The current program is the program that contains the instruction for a break currently taking place (i.e., the instruction indicated by the program counter (PC)).

When specifying a file in the current program, specifying the program name may be omitted.

**(6) File**

The allowable number of characters in a file name are up to 127 characters in a full path.

Depending on the file type, it may be possible to specify a file name more than 127 characters.

**(7) Current file**

The current file is the source file that has the instruction for a break currently taking place (i.e., the instruction indicated by the program counter (PC)).

When specifying a line or function in the current file by using a command, the file name can be omitted.

**(8) Function**

Functions constitute a C source program.

**(9) Current function**

The current function is the function that has the instruction for a break currently taking place (i.e., the instruction indicated by the program counter (PC)). To access a local variable in the current function, specifying the function name can be omitted.

**(10) Structure**

Structures and unions of the C language are generically referred to as “structures”. When a structure or union variable is used without explicitly specifying a member, the term “structure” is used.

**(11) Stack frame number**

A stack frame number is a decimal number starting from 1. The stack contents are displayed numbered. The shallower the nesting of the stack, the higher the number. Therefore, a function having a stack number 1 less than the stack number of another function calls that function.

**(12) Line**

A line is specified to identify a certain line in the source file.  
Line numbers are specified by an integer constant that starts from 1.

**(13) Register name**

A register name is specified by the absolute name of a system register or general-purpose register of the V850 Family microprocessors listed in Table B-2, or by the byname of a register.

**Table B-2. Register Set of V850 Family**

Register Type	Register Name		Register Type	Register Name	
	Absolute Name	Byname		Absolute Name	Byname
System registers	EIPC		General-purpose registers	r13	
	EIPSW			r14	
	FEPC			r15	
	FEPSW			r16	
	ECR			r17	
	PSW			r18	
	CTPC (V850E) <sup>Note</sup>			r19	
	CTPSW (V850E) <sup>Note</sup>			r20	
	DBPC (V850E) <sup>Note</sup>			r21	
	DBPSW (V850E) <sup>Note</sup>			r22	
	CTBP (V850E) <sup>Note</sup>			r23	
Program counter	PC		r24		
General-purpose registers	r0	zero	r25		
	r1		r26		
	r2		r27		
	r3	sp	r28		
	r4	gp	r29		
	r5	tp	r30	ep	
	r6		r31	lp	
	r7		/		
	r8				
	r9				
	r10				
	r11				
	r12				

**Note** These system registers exist in addition to the system registers described above in the V850E.

**(14) Numeric value**

The following four types of numeric values can be used.

- Binary input format:           nY  
  n...nY (n = 0, 1)
  
- Octal input format:            nO  
  n...nO (n = 0 to 7)  
  nQ  
  n...nQ (n = 0 to 7)



- Decimal input format:
  - n
  - n...n
  - nT
  - n...nT (n = 0 to 9)
- Hexadecimal input format:
  - n
  - n...n
  - nH
  - n...nH
  - 0xn
  - 0xn...n (n = 0 to F)

- Cautions**
1. **Suffixes (Y, O, Q, T, H, 0x) and the alphabet letters in hexadecimal numbers are not case sensitive.**
  2. **If the first character is one of A to F, a 0 must be affixed in front.**
  3. **The radix in the input column switches between decimal and hexadecimal in accordance with the default radix.**

#### (15) Address

- An address is specified by directly specifying a numeric value.
- An address can also be specified by using a symbol or expression.
- When specifying an address by using a numeric value, hexadecimal, decimal, octal, or binary numbers can be used.

#### (16) Symbol

- A symbol consists of characters A to Z, a to z, @, \_ (underline), and 0 to 9.
- A symbol must begin with a character other than numerals 0 through 9.
- This software distinguishes between uppercase (A to Z) and lowercase (a to z) characters.
- A symbol name can consist of up to 256 characters.
- If a symbol exceeding 256 characters is specified, the first 256 characters are valid.
- A symbol is defined by loading a load module file.
- Symbols can be classified into the following types by the valid range.

##### (a) Global symbols (assembly language, C language)

##### (b) Static symbols (C language)

Static symbols in files

Static symbols in functions

##### (c) Local symbols (C language)

Local symbols in file

Local symbols in function

Local symbols in block

- The following symbols exist in each of the languages used.

**(a) Assembly language**

Label name

**(b) C language**

Variable name (including pointer function name, enumerate type variable name, array name, structure name, and union name)

Function name, label name

Array element, structure element, union element (if the symbol is an array, structure, or union)

- A symbol can be described instead of an address and numeric value.
- The valid range of symbols is determined based on the source debugging information when the source file is assembled or compiled.
- A global symbol only describes a symbol name.
- A local symbol is expressed as a pair with a file name.

**(17) Expression and operator**

**<Expression>**

- An expression consists of constants, register names, peripheral I/O register name, and symbols combined by operators.
- If a peripheral I/O register name, label name, function name, or variable name is described as a symbol, an address is calculated as the value of the symbol.
- Elements constituting an expression, other than operators, are called terms (constants and labels). Terms are called the first term, second term, and so on, in the sequence they are described, starting from the one on the left.

**<Operator>**

- The following operators of C language can be used.

**Arithmetic operators**

Symbol	Meaning	Remark
+	Addition	Returns sum of values of first and second terms
–	Subtraction	Returns difference in value between first and second terms
*	Multiplication	Returns product of values of first and second terms
/	Division	Divides value of first term by value of second term, and returns integer of result
MOD, %	Remainder	Divides value of first term by value of second term, and returns remainder of result
– sign	Unary operation (negative)	Returns two's complement of value of term
+ sign	Unary operation (positive)	Returns two's complement of value of term

**Logical operators**

Symbol	Meaning	Remark
NOT ~	Negation	NOTs each bit of term, and returns result.
AND &	Logical product	ANDs each bit of values of first and second terms, and returns result
OR 	Logical sum	ORs each bit of values of first and second terms, and returns result
XOR ^	Exclusive logical sum	Exclusive-ORs each bit of values of first and second terms, and returns result

**Shift operators**

Symbol	Meaning	Remark
SHR >>	Right shift	Right shifts the first term the amount of the value (bit number) expressed by the second term and returns the result. The number of higher bits shifted are replaced with 0.
SHL <<	Left shift	Left shifts the first term the amount of the value (bit number) expressed by the second term and returns the result. The number of lower bits shifted are replaced with 0.

**Byte resolution operators**

Symbol	Meaning	Remark
HIGH	Higher byte	The higher 8 bits of the term's lower 16 bits are returned.
LOW	Lower byte	The lower 8 bits of the term's lower 16 bits are returned.

**Word resolution operators**

Symbol	Meaning	Remark
HIGHW	Higher word	The higher 16 bits of the term's 32 bits are returned.
LOWW	Lower word	The lower 16 bits of the term's 32 bits are returned.

**Others**

Symbol	Meaning	Remark
(	Left parenthesis	Executes operation in ( ) before operations outside ( )
)	Right parenthesis	

- Cautions**
1. Parentheses "(" and ")" must be always used in pairs.
  2. Operations are performed in compliance with the following rules.
    - The operation sequence is in accordance with the priority of the operator.
    - When the priority of operators is the same, the operation sequence is from left to right.

- The operation within parentheses is executed before the operations outside the parentheses.
- Each term of an operation is treated as unsigned 32-bit data.
- All the operation results are treated as unsigned 32-bit data.
- If an overflow occurs during operation, the lower 32 bits are treated as valid, and the overflow is not detected.

3. The priority of the operator is as follows.

Priority		Operator
1	↑ Higher	(, )
2		+ sign, – sign, NOT, ~, HIGH, LOW, HIGHW, LOWW
3		*, /, MOD, %, SHR, >>, SHL, <<
4	Lower ↓	+, –
5		AND, &
6		OR, I, XOR, ^

<Term>

- The following numeric values can be described when a constant is described as a term.

(a) Binary

0Y ≤ numeric value ≤ 11111111111111111111111111111111Y (32 digits)

(b) Octal

0O ≤ numeric value ≤ 3777777777O

(c) Decimal

–2147483648 ≤ numeric value ≤ 4294967295

A negative decimal number is internally converted into two’s complement.

(d) Hexadecimal

0H ≤ numeric value ≤ 0FFFFFFFFH

(18) Character set

With the ID850NW, the following character set can be used.

Character set

Symbol		Description
Alphabet character	Uppercase characters	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
	Lowercase characters	a b c d e f g h i j k l m n o p q r s t u v w x y z
Numerals		0 1 2 3 4 5 6 7 8 9
Alphabet equivalent characters		@ _
Special characters		. , ; * / % + – ‘ ” < > ( ) ! [ ] & \ ^   ~ # = \$ ? ‘ { }

**Other character set**

Character	Name	Main Usage
.	Period	Direct member operator, bit position specifier
,	Comma	Delimiter of operands
*	Asterisk	Multiplication operator or indirect reference operator
/	Slash	Division operator
%	Percent	Remainder operator
+	Plus	Positive sign or addition operator
-	Minus	Negative sign or subtraction operator
(	Left parenthesis	Changes operation sequence
)	Right parenthesis	Changes operation sequence
[	Left bracket	Array subscript operator, base register specification symbol
]	Right bracket	Array subscript operator, base register specification symbol
&	Ampersand	Address operator or bit AND operator
^	Control	Bit XOR operator
	Vertical line	Bit OR operator
~	Tilde	Complement operator

## B.2 Window Type and Configuration

The ID850NW debugger consists of windows and dialog boxes.

Basically, a window can be opened by using an icon, while a dialog box cannot.

### B.2.1 Windows

Windows are broadly divided into the following 2 types.

- Main windows
- MDI child windows

MDI child windows cannot be displayed outside of the main window frame.

### B.2.2 Dialog boxes

Dialog boxes can be classified into the following two types.

- Modal dialog boxes
- Modeless dialog boxes

#### (1) Modal dialog boxes

When this type of dialog box is open, the user cannot access other windows or dialog boxes until the dialog box is exited.

When the dialog box is closed after the operation in the dialog box has been completed or by selecting the  button of the dialog box, other windows and dialog boxes can be accessed.

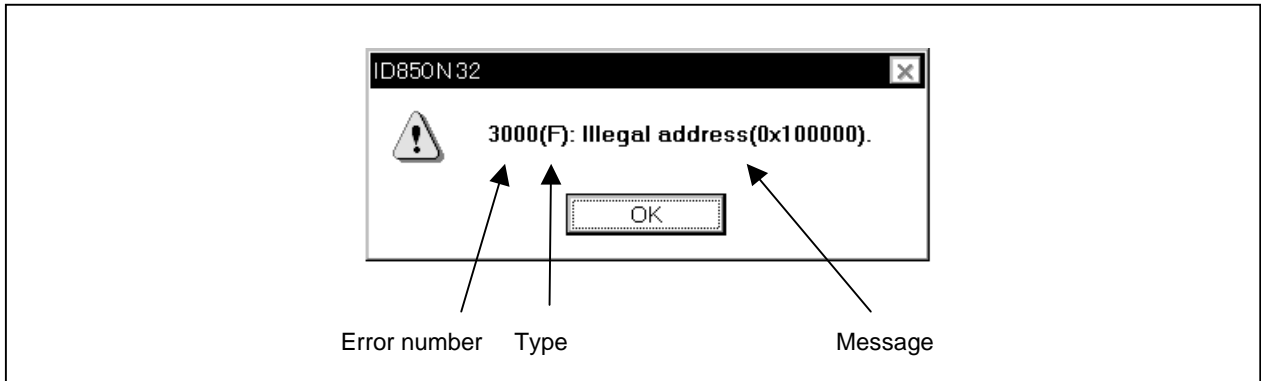
#### (2) Modeless dialog boxes

Unlike a modal dialog box, the user can access other windows and dialog boxes even if the operation in the dialog box has not been completed.

## APPENDIX C ERROR MESSAGE LIST

The error messages displayed in the ID850NW consist of **Error No.** + **Type** + **Message**.

**Figure C-1. Display Format of Error Message**



When the **OK** button is clicked, the Error Message window is closed, and the operation is performed in accordance with the message.

The error type is identified by the following three uppercase characters.

**Table C-1. Error Message Type**

Type	Meaning
A	Fatal error ( <u>A</u> bort Error). The processing is aborted and the debugger is terminated. If this error occurs, the debugging work cannot be continued.
F	Syntax error ( <u>F</u> atal Error). The processing is aborted, and all the windows and dialog boxes are closed.
W	<u>W</u> arning. The processing is aborted. All the windows and dialog boxes are not closed.

Table C-2 lists the error messages.

Table C-2. Error Message List (1/22)

Error No.	Type	Error Message/Meaning
0002	F	This feature is not supported. This function is not supported.
0100	A	Can not communicate with ICE. Please confirm the installation of the device driver for the PC interface board. Communication with the in-circuit emulator cannot be established. Check that the device driver for the interface board has been correctly installed.
0101	A	Can not find initialization file (expc.ini). An initialization file (EXPC.INI) cannot be found.
0102	A	Host name not found. No host name is found.
0103	A	Data transfer to ICE is timed out. Please confirm the power of ICE, connection of the interface cable, or I/O address of the PC interface board. Data could not be transmitted to the in-circuit emulator. Check the in-circuit emulator's power supply, cable connection, I/O address settings, etc.
0104	A	Data receive from ICE is timed out. Please confirm the power of ICE, connection of the interface cable, or I/O address of the PC interface board. The in-circuit emulator returns no response. Check the in-circuit emulator's power supply, cable connection, I/O address settings, etc.
0105	A	Failed in reading device file (d3xxx.800). The device file (d3xxx.800) could not be correctly read.
0106	A	Illegal data received. Received data is abnormal.
01a0	A	No response from the evachip. Please confirm the signal of the CLOCK or RESET, WAIT, HLDRQ and so on. There was no response from the evaluation chip. Check that the <u>RESET</u> , <u>WAIT</u> , <u>HLDRQ</u> (etc) signals or the clock signal are not abnormal.
01a1	A	Failed in reading ie703000.ie. ie703000.ie could not be correctly read.
01a2	A	Break board is not connected. The BK board is not connected.
01a3	A	Emulation board is not connected. The emulation board is not connected.
01a4	A	Board configuration of ICE is not consistent. The board configuration in the in-circuit emulator is not consistent.
01a5	A	POD/EM1 board is not connected. The POD/EM1 board is not connected.
01a6	A	Executor is running. The executor is running.
01a8	A	Failed in reading initialization file (expc.ini). The initialization file (EXPC.INI) has not been correctly read.
01d0	A	Failed in starting simulator. Simulator startup has failed.
01d1	A	Not enough memory in starting simulator. There is not enough memory in starting simulator.



Table C-2. Error Message List (2/22)

Error No.	Type	Error Message/Meaning
0200	F	Verification error. A verify error.
02a0	F	Bus hold error. The bus is in the hold status.
02d2	F	Not enough memory for trace-buffer. Memory for trace frame cannot be allocated.
0300	F	User program is running. The user program is being executed.
0301	F	User program is being broken. The user program is being broken.
0302	F	User program is being traced. The user program is being traced.
0303	F	Not traced. Tracer measurement is not performed.
0304	F	Trace memory is not set. The tracer is off.
0306	F	No trace block exists. There is no trace block.
0307	F	No event condition exists. There is no event condition.
0308	F	No timer measurement is done. Timer measurement is not performed.
0309	F	No trigger frame exists. There is no trigger frame.
030a	F	Tracer is being stopped. The tracer is stopped.
030b	F	Specified snap-event has not been registered. The snap event to be referenced/deleted is not registered.
030c	F	Specified stub-event has not been registered. The stub event to be referenced/deleted is not registered.
030d	F	Timer is running. The timer is operating.
030e	F	Memory copy area is overlapped. The memory copy range has been overlapped.
030f	F	Trace has been already set. Trace is already in the ON status.
0310	F	Event condition is not set. No event condition is set.
0311	F	Too many valid timer event conditions. The number of valid timer event conditions exceeds the maximum settable number.
0312	F	Specified timer event is not set. This timer event is not set.

**Table C-2. Error Message List (3/22)**

Error No.	Type	Error Message/Meaning
0313	F	Illegal map range. The map setting is wrong.
03a0	W	Target is not turned on. Power is not supplied to the target device.
03a1	F	Step execution is being done. This command is under step execution.
03a2	F	Timer and Tracer are running. The timer and tracer are operating.
03d0	A	Backtrace is being executed. Back trace is under execution.
03d1	A	Backtrace is being stopped. Back trace is stopped.
03d2	A	Backtrace execution point overrun oldest frame. The back trace execution stop position exceeds the oldest frame.
03d3	A	Illegal AND condition. AND condition is illegal.
03d4	A	No backtrace information exists. The back trace information does not exist.
03d5	A	Last command can not be backstepped. Back step of the last instruction could not be performed.
0400	F	Illegal condition. The specified condition is illegal.
0401	F	Result of timer measurement overflowed. The timer measurement result has overflow.
0402	F	Too many event conditions with path count. Too many event conditions that set path count have been used.
0403	F	Too many address range conditions. The maximum value of the address range specification condition is exceeded.
0404	F	Too many simultaneously-usable-event conditions. The maximum number of event conditions that can be used simultaneously is exceeded.
0405	F	Too many snap-events. The maximum number of snap events that can be set is exceeded.
0406	F	Too many stub-events. The maximum number of stub events that can be set is exceeded.
0407	F	Too many initialization data. The number of initialized data exceeds the initialization area.
0408	F	Too large search data (> 16 byte). The searched data exceed 16 bytes.
0409	F	Too large search data (> search range). The size of the searched data exceeds the size of the search range.
040a	F	Too many Linking-event conditions. The maximum number of integrated events that can be set is exceeded.

**Table C-2. Error Message List (4/22)**

Error No.	Type	Error Message/Meaning
04a0	F	Too many software breaks (> 100). The number of software breaks exceeds 100.
04a1	F	Not enough memory for emulation. The memory for emulation is insufficient.
04a2	F	Too many partition of bus size. The maximum number of delimiters of bus size has been exceeded.
04a3	F	Too many execution-event conditions. The maximum number of execution event conditions has been exceeded.
04a4	F	Too many bus-event conditions. The maximum number of bus event conditions has been exceeded.
0600	A	Can not allocate buffer. A buffer could not be allocated.
0c00	F	Monitor file read error. The monitor file could not be read.
0c20	F	Guarded area can not be accessed. There was an attempt to access a guarded area.
0c40	F	Different address condition is specified in re-registering event. The address condition when the event was reregistered was different to the previous one.
0c41	F	Coverage test is being executed. A coverage test is under execution.
0c60	F	Event number with different condition exists. An event number with a different condition exists.
0c61	F	Can not register event numbers which can not be used for hardware break. It is impossible to register event numbers that cannot be used for hardware breaks.
0c62	F	Event numbers reserved for hardware breaks can not be used. It is impossible to use event numbers that are used for hardware breaks.
0ca0	F	Can not communicate with ICE. Please confirm the power of ICE, connection of the interface cable, or I/O address of the PC interface board. Communication with the in-circuit emulator could not be established. Check the in-circuit emulator's power supply, cable connection, I/O address settings, etc.
0ca1	F	Monitor file not found. The monitor file could not be found.
1000	A	Failed in initializing ICE. Initializing the in-circuit emulator has failed.
1001	A	No entry exists for specified number. The entry of the specified number does not exist.
1002	A	Can not relocate internal RAM. The internal RAM cannot be relocated.
1003	F	Illegal relocation address. The relocation address is illegal.
1004	F	Illegal condition. The specified condition is illegal.
1005	A	Invalid attribute. The attribute is invalid.

Table C-2. Error Message List (5/22)

Error No.	Type	Error Message/Meaning
1006	F	Illegal address. The address is illegal.
1007	A	Not enough memory on ICE. In-circuit emulator alternate memory is short.
1008	A	Can not allocate memory for tables. Memory for tables cannot be allocated.
1009	A	Already initialized. Already initialized.
100a	A	Not initialized. Not initialized yet.
100b	F	User program is running. The user program is being executed.
100c	F	Different bus size has been already specified. An attempt was made to set in duplicate an area with different bus size.
100d	F	Too large bus size. The bus size exceeds the maximum value able to be defined.
100e	F	Too large bus partition size. The delimiter of bus size has exceeded the maximum value.
100f	W	Target is not turned on. The power supply of the target is off.
1010	A	Illegal map range. An illegal map range specification was made.
1011	F	Failed in setting internal ROM and RAM. Setting the internal ROM and RAM has failed.
1012	F	This feature is not supported. This function is not supported.
1013	F	No terminal name. This pin name does not exist.
10ff	A	Can not communicate with ICE. A communication error has occurred. Communication with the in-circuit emulator cannot be performed.
1dbe	A	Internal error. Internal error of the debugger.
2000	F	Illegal I/O register name. The I/O register name is illegal.
2001	A	Illegal address. The address is illegal.
2002	F	User program is running. The user program is being executed.
2003	F	Illegal I/O register number. An illegal number of I/O registers were specified.
2004	F	Illegal bit number. The bit position is illegal.

Table C-2. Error Message List (6/22)

Error No.	Type	Error Message/Meaning
2005	W	Redraw I/O register name. An I/O register that must not be redrawn is specified.
2006	F	This I/O register is hidden I/O register. An I/O register with a hidden attribute was specified.
2007	F	Can't read/write. A read/write disabled I/O register was specified.
2008	F	Too big number. The specified I/O register does not exist.
2009	A	Error in device file. A device file error.
200a	F	Illegal value specified for I/O register. An illegal value was specified for the I/O register. An attempt was made to set the value that can't be written to IOR.
200b	A	Can not copy. Copying has failed.
200c	A	Can not allocate memory. Memory allocation has failed.
20ff	A	Can not communicate with ICE. A communication error has occurred. Communication with in-circuit emulator cannot be established.
2222	A	Illegal condition. The specified condition is illegal.
3000	F	Illegal address. The address is illegal.
3001	F	Memory has different value. The memory contents do not match.
3002	F	Illegal source address. The source address is illegal.
3003	F	Illegal destination address. The destination address is illegal.
3004	F	Illegal address (source & destination). Both source address and destination address are illegal.
3005	F	Illegal condition. The specified condition is illegal.
3006	F	User program is running. The user program is being executed.
3007	A	Verification error. Verify error.
3008	F	No condition specified. A condition has not been specified.
3009	F	Parameter size does not align with access size alignment. The size of the parameter is not access size alignment.
300a	F	Specified address does not align with access size alignment. The specified address is not access size alignment.

**Table C-2. Error Message List (7/22)**

Error No.	Type	Error Message/Meaning
300b	F	Source address does not align with access size alignment. The specified address (Source) is not access size alignment.
300c	F	Destination address does not align with access size alignment. The specified address (Destination) is not access size alignment.
300d	F	Illegal end address. The end address is illegal.
300e	F	Different access size in specified area. The access size differs in the specified range.
300f	F	Different access size both in source and destination areas. The access size differs in the specified range (Source).
3010	F	Different access size in destination area. The access size differs in the specified range (Destination).
3011	F	Different access size, source & destination. The access size differs between the specified range (Source) and specified range (Destination).
3012	A	Can not allocate memory. Memory allocation has failed.
3013	A	Failed in writing DMM. Writing DMM has failed.
3014	F	Overflowed mapping area. The mapping area is exceeded.
3015	A	Interrupted. Processing is aborted.
3016	F	This feature is not supported. This function is not supported.
30ff	A	Can not communicate with ICE. A communication error has occurred. Communication with the in-circuit emulator cannot be established.
4000	F	Can not delete specified event number. The specified event number cannot be deleted.
4001	F	Illegal table number. The specified table number is illegal.
4002	F	Illegal start address. The start address is illegal.
4003	F	Illegal end address. The end address is illegal.
4004	F	Illegal status. The status is illegal.
4005	F	Illegal data. The data is illegal.
4006	F	Specified event number has been already used. An attempt was made to specify an event number that has already been used.
4007	F	Too many same events are registered. The maximum of the number of registered events of the same type is exceeded.

Table C-2. Error Message List (8/22)

Error No.	Type	Error Message/Meaning
4008	F	Specified event has not been registered. The specified event is not registered.
4009	F	Illegal data size. The data size is illegal.
400a	F	Illegal mode. The mode is illegal.
400b	F	Illegal condition. The specified condition is illegal.
400c	F	Illegal type. The type is illegal.
400d	F	Too many identical events are registered ( $\geq 32767$ ). An attempt was made to make registration exceeding 32,767 of the same registered events.
400e	F	Specified event condition does not exist. The specified event condition does not exist.
400f	F	Illegal event link condition. An event link condition is illegal.
4010	F	Function not found. The specified function was not found.
4011	F	Not enough memory. Application has run out of memory.
4012	F	Timer is being disabled. The timer is invalid.
4013	W	Access size is different from its mapped bus size. The access size and mapped bus size differ.
4014	F	Can not use software break. Software break cannot be used.
4015	F	Can not use event condition specifying address range. An event condition with a range specified cannot be used for an address condition.
4016	F	Can not change event condition. The event condition cannot be changed.
4017	F	Can not access word at odd address. Word access to an odd address cannot be performed.
4018	A	Not enough memory. Application has run out of memory.
4019	F	Not support. This function is not supported.
401a	F	No Event. There is no event.
401b	F	Can not use tag-event. A tag event cannot be used.
4318	F	Illegal memory bank setting. The setting of the memory bank is illegal.

Table C-2. Error Message List (9/22)

Error No.	Type	Error Message/Meaning
5000	A	Illegal type. The type is illegal.
5001	A	Can not allocate memory. Memory allocation has failed.
5002	A	Can not open device file. Opening the device file has failed.
5003	A	Can not seek device file. Seeking the device file has failed.
5004	A	Can not close device file. Closing the device file has failed.
5005	A	Illegal device file format. The format of the device file is wrong.
5006	A	Failed in initializing ICE. Initializing the in-circuit emulator has failed.
5007	A	Device information does not exist. Device information is missing.
5008	F	Can not open device file. The specified device file cannot be opened.
5009	F	Can not open ie703000.ie. ie703000.ie cannot be opened.
500a	F	Specified device file is illegal version. The version of the device file is illegal.
500b	W	Specified device file does not relocate IRAM. This device file does not relocate IRAM.
500c	A	Failed in reading expc.ini. The initialization file (EXPC.INI) cannot be correctly read.
500d	A	Not enough memory. The memory has run short.
5300	A	Illegal type. The type is illegal.
5301	A	Can not allocate memory. Memory allocation has failed.
5302	A	Can not open database file. The database file could not be opened.
5303	A	Can not seek database file. The database file could not be searched.
5304	A	Can not close database file. The database file could not be closed.
5305	A	Illegal device format. The format of the database file is illegal.
5306	A	Database information has been already initialized. Initialization of the database information is already finished.



Table C-2. Error Message List (10/22)

Error No.	Type	Error Message/Meaning
5307	A	Database information does not exist. There is no database information.
5308	F	Can not open specified database file. The specified database file could not be opened.
5309	F	Specified database file is illegal version. The version of the database file is illegal.
6000	F	Current function does not exist. The current function does not exist.
6001	F	Illegal symbol name. The symbol name is illegal.
6002	F	Illegal condition. The specified condition is illegal.
6003	F	Illegal function name. The function name is illegal.
6004	F	Overflowed output buffer size. The data size exceeds that of the output buffer.
6005	F	Illegal expression. The expression is illegal.
7000	F	Illegal mode. The specified mode is wrong.
7001	F	User program is running. The user program is being executed.
7002	F	User program has been stopped. The user program is being stopped.
7003	F	Trace enabled. Trace is being executed.
7004	F	Trace memory is not set. The trace memory is OFF.
7005	F	Function return address does not exist, can not do step execution. The return address of the function does not exist. Step execution is not performed.
7010	W	No source information exists. The source information is missing.
7011	W	Unknown result of step execution. The result of step execution is unknown.
7012	A	Not enough memory. Application has run out of memory.
70fe	A	Bus hold error. The bus is in the hold status.
70ff	A	Can not communicate with ICE. A communication error has occurred. Communication with the in-circuit emulator cannot be established.
7801	F	Canceled step wait. The wait status following completion of step execution has been cancelled.

Table C-2. Error Message List (11/22)

Error No.	Type	Error Message/Meaning
7802	F	Interrupted step execution. Step execution processing was aborted.
7f00	F	Aborted step execution. Step execution processing was forcibly terminated.
7f02	F	Suspended step execution. Step execution was aborted.
7f03	A	Failed in canceling RUN/STEP. Canceling RUN/STEP has failed.
7f04	F	Can not execute non-mapped area. An attempt is made to execute from an area not mapped.
7f05	F	This feature is not supported. This function is not supported.
8000	F	File not found. The specified file is not found.
8001	F	Illegal line number. The line number is illegal.
8002	F	Current information is not set. The current information is not set.
8003	F	Illegal address. The address is illegal.
8004	F	This feature is not supported. This function is not supported.
9000	A	Specified register symbol does not exist. The specified register symbol does not exist.
9001	A	Specified register symbol ID does not exist. The specified register symbol ID does not exist.
9002	F	Illegal value. The specified value is illegal.
9003	A	Illegal condition. The specified condition is illegal.
9004	A	Too large register size. The register size is too big.
9005	F	This feature is not supported. This function is not supported.
a001	F	Illegal expression. The expression is illegal.
a002	F	Start address is bigger than the end address. The start and end address specifications have been reversed.
a003	F	Illegal source path. The specified source path information is illegal.
a004	F	Too long expression. The expression is too long.

Table C-2. Error Message List (12/22)

Error No.	Type	Error Message/Meaning
a005	A	Not enough memory. Application has run out of memory.
a006	F	Illegal argument. The argument is illegal.
a007	A	Illegal program number. The program number is wrong.
a008	F	Source path is not set. The source path is not set.
a009	F	File not found. The file is not found.
a00a	F	Can not open file. Opening the file has failed.
a00b	A	Can not close file. Closing the file has failed.
a00c	A	Failed in reading file. Reading the file has failed.
a00d	F	Not source file of load module. This is not the load module's source file.
a00e	F	Illegal line number. The line number is illegal.
a00f	F	Variable does not exist. The variable does not exist.
a010	A	Can not communicate with ICE. A communication error has occurred. Communication with the in-circuit emulator cannot be established.
a011	F	Can not access register. The register cannot be accessed.
a012	F	Can not access memory. The memory (variable) cannot be accessed.
a013	F	Can not seek file. Seeking a file has failed.
a014	F	This is binary file. This is a binary file.
a015	F	Can not get temporary path. A temporary path could not be allocated.
a016	F	Can not create temporary file. A temporary file could not be created.
a017	F	Can not remove temporary file. A temporary file could not be deleted.
a020	F	This feature is not supported. This function is not supported.
a021	F	Specified symbol is assigned to register. The specified symbol has already been assigned to a register.

Table C-2. Error Message List (13/22)

Error No.	Type	Error Message/Meaning
b000	F	Illegal command line. The command line is illegal.
b001	F	Program information does not exist in specified load module file. The load module file does not have program information.
b002	F	File not found. The file is not found.
b003	F	Function not found. The function is not found.
b004	F	Illegal magic number. The magic number is illegal.
b005	F	Symbol not found. The symbol is not found.
b008	F	Illegal expression. The expression is illegal.
b009	A	Not enough memory. Application has run out of memory.
b00a	F	Illegal symbol in load module file. An illegal symbol exists in the load module file.
b00b	F	Current program does not exist. The current program does not exist.
b00c	F	Current file does not exist. The current file does not exist.
b00d	A	Current function does not exist. A current function does not exist.
b00e	A	Current line does not exist. A current line does not exist.
b00f	A	Tag not found. No tag is found.
b010	A	Failed in loading symbol table. Loading the symbol table has failed.
b011	A	Illegal line number. The line number is illegal.
b012	F	Too large line number. The line number is too large.
b015	A	Failed in reading file. Reading the file has failed.
b016	A	Can not open file. Opening the file has failed.
b017	A	Failed in writing file. Writing the file has failed.
b019	A	Can not seek file. Seeking the file has failed.

Table C-2. Error Message List (14/22)

Error No.	Type	Error Message/Meaning
b01a	A	Can not close file. Closing the file has failed.
b01b	A	Too long load module file name. The file name of the load module is too long.
b01c	A	Too many entries of the task kind. There are too many task classification entries.
b01d	F	Address not found. The address does not exist.
b01e	W	No line information (not compile with -g). There is no line information (it was not compiled using -g).
b01f	F	Can not find structure member. The member of the structure cannot be found.
b020	F	Can not find value. The value could not be found.
b021	F	No symbol information exists in load module file. The load module file does not have symbol information.
b022	F	Illegal line number. The line number is illegal.
b023	A	Current stack frame is not active. The current stack frame is not active.
b024	A	Different section. The section is wrong.
b026	F	Too many array dimensions (> 4). The displayed array exceeds the fourth dimension.
b027	F	Found end of file. The file ends in the middle.
b028	F	This feature is not supported. This function is not supported.
b029	F	Illegal address. The address is illegal.
b02a	A	Can not communicate with ICE. A communication error has occurred. Communication between in-circuit emulator cannot be established.
b02b	F	Can not stack trace with current PC value. Stack trace cannot be executed with the current PC value.
b02c	F	Too many blocks for one function. The maximum number of blocks allowable in 1 function has been exceeded.
b02d	F	Illegal argument. The argument passed to the function is illegal.
b02e	A	Path not found. The specified path is not found.
b02f	A	Information has been deleted because of optimization. Information has been deleted as a result of optimization.

Table C-2. Error Message List (15/22)

Error No.	Type	Error Message/Meaning
b030	A	Monitor timed out. Time out of the monitor.
b031	A	Already set in memory. Already set on the memory.
b032	A	Out of scope. Out of scope.
b033	A	LP is not stored. LP has not been stored.
b034	A	Inside of prologue or epilogue. Within the prologue or epilogue.
b036	W	Out of variable region. The specified variable cannot be referenced by the current PC.
b037	F	Too Many Line-Numbers Information. There is too much line information.
b038	F	Compiler version mismatch. Because this compiler version is not supported, the debug information could not be loaded.
b039	A	Failed in loading debug information. The debug information could not be loaded.
b03a	A	No more section information. There is no section information.
b040	A	Specified file is not load module. The specified file is not in a load module format.
b041	A	Too many files in load module to download. The number of files in the load module exceeds the maximum number the debugger can handle.
b042	W	Symbol module is not initialized. The SYM module has not been initialized.
b32e	F	Illegal port number. An illegal port number.
b32f	F	Illegal port name. An illegal port name.
b330	F	Illegal port position. The bit position of the specified port is illegal.
b331	F	Illegal increment number. The specified increment count is illegal.
b332	F	Port for memory bank is not set. The port for memory bank is not set.
b333	F	Illegal bank number. An illegal bank number is specified.
b334	F	Area for memory bank is not set. An area for memory bank is not set.
b335	W	Too long symbol name. The symbol name is longer than the maximum allowable length.

Table C-2. Error Message List (16/22)

Error No.	Type	Error Message/Meaning
c001	F	Can not open file. Opening the file has failed.
c002	A	Can not close file. Closing the file has failed.
c003	A	Failed in reading file. Reading the file has failed.
c004	A	Can not seek file. Seeking the file has failed.
c005	F	Illegal file type. A non-existent file type has been specified.
c006	F	Illegal magic number. The magic number is illegal.
c007	F	Specified file is not load module. The specified file is not in a load module format.
c008	F	Specified load module file (ELF) is old version. The version of the load module file (ELF) is old.
c009	F	Not enough memory. Application has run out of memory.
c00a	F	Illegal address. The address is illegal.
c00b	F	Load module is not loaded. The load module is not loaded.
c00c	F	Illegal argument. The argument is illegal.
c00d	F	User program is being emulated. The user program is being emulated.
c00e	F	User program is being traced. The user program is being traced.
c00f	A	Interrupted. Under processing.
c010	A	Can not communicate with ICE. A communication error has occurred. Communication between the in-circuit emulator cannot be established.
c011	F	Illegal load module file format. The format of the load module file is wrong.
c012	F	Check sum error. A check sum error has occurred.
c013	F	Too wide address range to upload (> 1M byte) The address range to be uploaded exceeds 1 Mbyte.
c014	F	Failed in writing file. Writing the file has failed.
c015	F	Illegal program number. The program number is illegal.

Table C-2. Error Message List (17/22)

Error No.	Type	Error Message/Meaning
c016	F	Load information is full. The load information is full.
c017	W	Symbol information is duplicated, please reset symbols. The symbol information overlaps. Reset the symbol.
c018	F	Specified file is not load module. The specified file is not in a load module format.
c019	F	Failed in writing memory. Memory could not be written to.
c01a	W	Illegal address. Address is illegal. The BSS section of the load module includes an illegal address.
c01b	F	Programmable-IOR address not specified. An attempt was made to download a load module that did not contain programmable I/O area data.
c01c	W	Programmable IOR address mismatch. The address of the programmable IOR differs from the one set. The address of the programmable I/O area set in the Configuration dialog box does not match the address of the programmable I/O area included in the load module.
c100	F	This feature is not supported. This function is not supported.
d000	A	Internal error. An error occurs inside the debugger.
d001	F	Not enough memory. Application has run out of memory.
d002	A	Failed in reading initialization file (expc.ini). The initialization file (EXPC.INI) cannot be read correctly.
d003	A	ICE is not connected. The in-circuit emulator is not connected.
d004	A	Can not find Dynamic Link Library. The dynamic link library could not be found.
e000	F	Illegal argument. The argument is illegal.
e001	F	Illegal start address. The start address is illegal.
e002	F	Illegal end address. The end address is illegal.
e003	F	Too large size. The address size is too large.
e004	F	Can not open file. Opening the file has failed.
e005	F	Failed in reading file. Reading the file has failed.
e006	F	Can not seek file. Seeking the file has failed.
e007	F	Failed in writing file. Writing the file has failed.



Table C-2. Error Message List (18/22)

Error No.	Type	Error Message/Meaning
e008	F	Not enough memory. Application has run out of memory.
e009	F	Illegal file format. The file format is wrong.
e00a	F	Verification error. Verify error.
e010	F	This feature is not supported. This function is not supported.
f000	F	Not enough memory. Application has run out of memory.
f001	F	XXX not found. XXX could not be found.
f002	F	not found XXX. Search from the beginning? XXX could not be found. Do you want to search from the beginning?
f003	W	Already exceed search region. The search range has already been exceeded.
f004	F	Missing parameter. The parameter is illegal.
f005	F	Illegal name. The name is illegal.
f006	F	Illegal number. The number is illegal.
f007	F	Start address is bigger than end address. Start address is bigger than end address.
f008	F	Illegal expression. The expression is illegal.
f009	F	XXX This file is illegal type. XXX This file is an illegal type.
f100	F	Disk cannot write or full. Either the disk could not be written to or it is full.
f101	F	File not found. This file could not be found.
f102	F	File not Create. This file could not be created.
f103	F	Old file version. This file is an old version.
f104	F	Illegal file type. The file format is illegal.
f105	F	This file is a project file for XXXX Please select a correct file. This project file is a file for XXX. Specify a correct file.
f200	W	No differences encountered. There were no differences found.
f201	F	Memory mapping error. The memory could not be mapped.

Table C-2. Error Message List (19/22)

Error No.	Type	Error Message/Meaning
f202	F	Verify error. Verify error.
f300	F	Would you like to save the changes made in XXX? Do you want the changes made in XXX to be saved?
f301	F	The symbol being used on the event condition can't be evaluated. The symbol being used in the event condition could not be evaluated.
f302	F	Delete XXX? Do you want XXX to be deleted?
f303	F	XXX is edited. Delete YYY? XXX is being edited; do you want YYY to be deleted?
f304	F	XXX is edited. Save YYY? XXX is being edited; do you want YYY to be saved?
f305	F	XXX is already exist. Do you replace it? Do you want to replace the existing XXX?
f306	F	This name is too long. This name is too long.
f307	F	There is the same name in other kinds. The same name is being used in other conditions.
f308	F	An address can't be omitted. An address cannot be omitted.
f309	F	Illegal address mask. The address mask is illegal.
f30a	F	Illegal data mask. The data mask is illegal.
f30b	F	Illegal ext probe mask. The external probe mask is illegal.
f30c	F	Illegal ext probe data. The external probe data is illegal.
f30d	F	Illegal pass count. The pass count is illegal.
f30e	F	Illegal register name. The register name is illegal.
f30f	F	Illegal register bank. The register bank is illegal.
f310	F	Illegal delay count. The delay count is illegal.
f311	F	Only one XXX can be enabled. Do you make this YYY to enable? Only one XXX can be enabled; do you want to enable YYY?
f312	F	XXX is already there. XXX already exists.
f313	F	Event number already exist. This event number already exists.
f314	F	Event name is not set. The event name has not been set.

Table C-2. Error Message List (20/22)

Error No.	Type	Error Message/Meaning
f315	F	XXX is already there. XXX already exists.
f316	F	Max number of XXX enabled event is over. Please disable other enabled event. The maximum allowable number of valid event conditions has been exceeded. Disable other event conditions.
f317	F	Max number of set event is over. The maximum number of settable event conditions has been exceeded.
f318	F	Max number of set event link is over. The maximum number of settable event link conditions has been exceeded.
f319	F	Max number of set break event is over. The maximum number of settable break conditions has been exceeded.
f31a	F	Max number of set trace event is over. The maximum number of settable trace conditions has been exceeded.
f31b	F	Max number of set snap event is over. The maximum number of settable snap conditions has been exceeded.
f31c	F	Max number of set stub event is over. The maximum number of settable stub conditions has been exceeded.
f31d	F	Max number of set timer event is over. The maximum number of settable timer conditions has been exceeded.
f31e	F	Illegal start address. The start address is illegal.
f31f	F	Illegal end address. The end address is illegal.
f320	F	Illegal bit address. The bit address is illegal.
f321	F	Specified read-protect I/O register. A read-protected I/O register has been specified.
f350	F	There is a phase which event are not in the middle. An event condition setting is missing from the middle of a phase.
f351	F	The same event is contained in Link and Disable. The same event condition is already being used.
f352	F	An event isn't specified. An event condition has not been set.
f353	F	Incompatible event is specified in phase 1. An incompatible event condition has been set in Phase 1.
f354	F	AND event is in Phase 1. Can't specify event with exception of phase 1. An AND condition event has already been set in Phase 1. Event conditions cannot be set anywhere except in Phase 1.
f355	F	REG/MEM event in Disable. The REG/MEM status event has been set to Disable.
f356	F	AND event is in Disable. The AND condition event has been set to Disable.
f400	F	Coverage mapping error. Coverage mapping could not be performed.

Table C-2. Error Message List (21/22)

Error No.	Type	Error Message/Meaning
f401	F	Clear coverage? Do you want to clear the coverage?
f500	F	Illegal symbol. The symbol is illegal.
f501	F	Illegal value. The value is illegal.
f502	F	Illegal parameter. The parameter is illegal.
f600	F	Save project file? Do you want to save the project file?
f601	W	When connecting the target system, please turn on the target system. When connecting the target system, be sure to turn its power supply on.
f700	F	Download failed. Data could not be downloaded.
f800	F	Configuration of Memory Bank is not set. The memory bank setting has not been made.
f801	F	BANK address must be in target memory. The address of the memory bank must be inside the target memory.
f802	F	All events are deleted because the use of external probe was changed. All event conditions have been deleted because the external probe has changed.
f803	F	This event address is invalid on current configuration. The address of this event condition is illegal.
f804	F	Invalid PC value. The PC value is illegal.
f805	F	Cannot set temporary break on this address. A temporary break cannot be set to this address.
f806	F	External data is being used by Debugger. External data is being used by the debugger.
f900	F	Missing parameter. The name is illegal.
f901	F	Memory mapping error. There is a fault with the memory mapping.
f902	F	Illegal access size. The access size is illegal.
f903	F	Illegal access type. The access type is illegal.
f904	F	There is the same name. The same name cannot be registered.
f905	W	XXX is already exist. Do you replace it? Do you want to replace the existing XXX?
f906	W	Would you like to register the change made in XXX? Do you want to register the changes made to XXX?
fa00	F	The "main( )" function of current program on PC position not found. The function "main( )" of the PC position program could not be found.

**Table C-2. Error Message List (22/22)**

Error No.	Type	Error Message/Meaning
fa01	F	The line information on PC position not found. The line information of the PC position could not be found.

[MEMO]

## APPENDIX D KEY FUNCTION LIST

Efficient debugging can be realized by performing operations using the special function keys.

Note that because the key explanations differ depending on the type of keyboard in IBM PC/AT and compatible machines, the key inscriptions employed here are general or generic key inscriptions.









### D.1 Special Function Key Function List

A list of the special function key functions of the PC-9821 series and IBM PC/AT and compatibles is shown in the table below.

**Table D-1. Special Function Key Function List (1/2)**

Key Type		Function
PC-9821 Series	IBM PC/AT Compatibles	
<b>BS</b>	<b>BackSpace</b>	Deletes a character preceding the cursor and moves the cursor to the position of the deleted character. At this time, the string of characters following the cursor moves front one space.
<b>DEL</b>	<b>Delete</b>	<ul style="list-style-type: none"> <li>• Deletes a character at the cursor position and moves the following characters front one space.</li> <li>• Deletes the event conditions selected in the Event Manager and each event dialog box.</li> <li>• Deletes the data selected in the Watch window.</li> </ul>
<b>INS</b>	<b>Insert</b>	Switches the modes between insert mode and overwrite mode in the Source and Assemble windows. However, this key is disabled in the Memory, Register, and IOR windows, where the input mode is overwrite mode only.
<b>COPY</b>	<b>PrintScreen</b>	Places the entire screen display on the clipboard as a bitmap image (Windows function).
<b>ESC</b>	<b>Esc</b>	<ul style="list-style-type: none"> <li>• Closes a pull-down menu.</li> <li>• Closes a modal dialog box.</li> <li>• Restores input data.</li> </ul>
<b>GRPH</b>	<b>Alt</b>	Moves the cursor on the menu bar.
<b>HELP</b>	<b>End</b>	Moves the cursor to the end of the line.
<b>HOME CLR</b>	<b>Home</b>	Moves the cursor to the start of the line.
<b>ROLL DOWN</b>	<b>PageUp</b>	Scrolls the screen up one screen. Simultaneously moves the cursor to the top of the screen.
<b>ROLL UP</b>	<b>PageDown</b>	Scrolls the screen down one screen. Simultaneously moves the cursor to the top of the screen.
<b>SPACE</b>	<b>Space</b>	Inserts a 1-character space.
<b>TAB</b>	<b>Tab</b>	Moves the cursor to the next item.
<b>↑</b>	<b>↑</b>	Moves the cursor up. When the cursor is at the top of the screen, the screen is scrolled down one line at a time.



















**Table D-1. Special Function Key Function List (2/2)**

Key Type		Function
PC-9821 Series	IBM PC/AT Compatibles	
		Moves the cursor down. When the cursor is at the bottom of the screen, the screen is scrolled up one line at a time.
		Moves the cursor to the left. When the cursor is on the far left of the screen, the screen is scrolled to the right one item at a time.
		Moves the cursor to the right. When the cursor is on the far right of the screen, the screen is scrolled to the left one item at a time.
		<ul style="list-style-type: none"> <li>• Sets the input data.</li> <li>• Presses the default push button.</li> </ul>

## D.2 Function Key Function List







A list of the function key functions of the PC-9821 series and IBM PC/AT and compatibles is shown in the table below.

**Table D-2. Function Key Function List (1/2)**


Key Type		Function
PC-9821 Series	IBM PC/AT Compatibles	
		Opens the Help window.
		Forcibly stops program execution. The same function as [Run] → [Stop] on the menu bar.
		Resets the emulation CPU. The same function as [Run] → [CPU Reset] on the menu bar.
		Executes the program after resetting the emulation CPU. The same function as [Run] → [Restart] on the menu bar.
		Executes the program. The same function as [Run] → [Go & Go] on the menu bar.
		Executes the program as far as the cursor position in the Source or Assemble window. The same function as [Run] → [Come Here] on the menu bar.
		Executes the program in real-time until execution returns to the calling origin. The same function as [Run] → [Return Out] on the menu bar.
		Performs step execution. The same function as [Run] → [Step In] on the menu bar.
		Sets/cancels a breakpoint at the cursor position in the Source or Assemble window. The same function as [Run] → [Break Point] on the menu bar.











**Table D-2. Function Key Function List (2/2)**

Key Type		Function
PC-9821 Series	IBM PC/AT Compatibles	
		Performs next step execution. The same function as [Run] → [Next Over] on the menu bar.
		Sets/cancels a software breakpoint. The same function as [Run] → [Software Break Point] on the menu bar.
		Enables/disables a breakpoint. The same function as [Run] → [Ignore Break Point] on the menu bar.


**D.3 Special Function Key Function List ( + Key)**

A list of the special function key functions ( + key) of the PC-9821 series and IBM PC/AT and compatibles is shown in the table below.



**Table D-3. Special Function Key Function List ( + Key)**

Key Type		Function
PC-9821 Series	IBM PC/AT Compatibles	
		Extends the selected range to the end of the line.
		Extends the selected range to the start of the line.
		Extends the selected range one character to the left.
		Extends the selected range one character to the right.

**D.4 Function Key Function List ( + Key)**

A list of the function key functions ( + key) of the PC-9821 series and IBM PC/AT and compatibles is shown in the table below.

**Table D-4. Function Key Function List ( + Key)**

Key Type		Function
PC-9821 Series	IBM PC/AT Compatibles	
		Executes the program from the cursor position in the Source or Assemble window. The same function as [Run] → [Start From Here] on the menu bar.

### D.5 Special Function Key Function List ( **CTRL** + Key)

A list of the special function key functions ( **CTRL** + key) of the PC-9821 series and IBM PC/AT and compatibles is shown in the table below.

**Table D-5. Special Function Key Function List ( **CTRL** + Key)**

Key Type		Function
PC-9821 Series	IBM PC/AT Compatibles	
<b>HELP</b>	<b>End</b>	Displays the last line. Simultaneously moves the cursor to the start of the last line.
<b>HOME CLR</b>	<b>Home</b>	Displays the first line. Simultaneously moves the cursor to the start of the first line.
<b>←</b>	<b>←</b>	Moves the cursor one word to the left. When the cursor is on the far left of the screen, the screen is scrolled one item to the right.
<b>→</b>	<b>→</b>	Moves the cursor one word to the right. When the cursor is on the far right of the screen, the screen is scrolled one item to the left.

### D.6 Function Key Function List ( **CTRL** + Key)

A list of the function key functions ( **CTRL** + key) of the PC-9821 series and IBM PC/AT and compatibles is shown in the table below.

**Table D-6. Function Key Function List ( **CTRL** + Key)**

Key Type		Function
PC-9821 Series	IBM PC/AT Compatibles	
<b>f-9</b>	<b>F9</b>	Sets the address of the cursor position in the Source or Assemble window in the PC. The same function as [Run] → [Change PC] on the menu bar.

### D.7 Control Key Function List ( **CTRL** + Key)

A list of the control key functions ( **CTRL** + key) of the PC-9821 series and IBM PC/AT and compatibles is shown in the table below.

**Table D-7. Control Key Function List ( **CTRL** + Key) (1/2)**

Key Type	Function
PC-9821 Series IBM PC/AT Compatibles	
<b>D</b>	Shows a disassemble display from the jump destination address of the data value selected in the current window. Opens the Assemble window. The same function as [Jump] → [Assemble] on the menu bar.
<b>C</b>	Copies the selected character string and places it in the clipboard buffer. The same function as [Edit] → [Copy] on the menu bar.

**Table D-7. Control Key Function List ( **CTRL** + Key) (2/2)**

Key Type		Function
PC-9821 Series	IBM PC/AT Compatibles	
<b>G</b>		Performs a search. Opens the search dialog box appropriate to the current window. The same function as [View] → [Search...] on the menu bar.
<b>J</b>		Moves the display position. Opens the Source Move, Address Move, and Trace Move dialog box appropriate to the current window. The same function as [View] → [Move...] on the menu bar.
<b>M</b>		Displays the memory contents from the jump destination address of the data value selected in the current window. Opens the Memory window. The same function as [Jump] → [Memory] on the menu bar.
<b>O</b>		Loads a display, source, or text file. Opens the View File Load dialog box. Operations differ depending on the file extension. For display files: Displayed in corresponding window. For other files: Displayed in Source window. The same function as [File] → [Open...] on the menu bar.
<b>U</b>		Displays the corresponding source text and source line from the jump destination address of the data value selected in the current window. Opens the Source window. The same function as [Jump] → [Source] on the menu bar.
<b>V</b>		Pastes the contents of the clipboard buffer at the cursor position. The same function as [Edit] → [Paste] on the menu bar.
<b>W</b>		Temporarily displays the contents of the specified data. Opens the Quick Watch dialog box. The same function as [View] → [Quick Watch...] on the menu bar.
<b>X</b>		Removes the selected character string and places it on the clipboard buffer. The same function as [Edit] → [Cut] on the menu bar.

**D.8 Special Function Key Function List ( **CTRL** + **SHIFT** Key)**

A list of the special function key functions ( **CTRL** + **SHIFT** key) of the PC-9821 series and IBM PC/AT and compatibles is shown in the table below.

**Table D-8. Special Function Key Function List ( **CTRL** + **SHIFT** Key)**

Key Type		Function
PC-9821 Series	IBM PC/AT Compatibles	
<b>←</b>	<b>←</b>	Extends the selected range one word to the left.
<b>→</b>	<b>→</b>	Extends the selected range one word to the right.

[MEMO]

## APPENDIX E INDEX

### A

About dialog box ..... 285  
Access size conditions..... 249  
Active state .....141, 158, 170  
Add I/O Port dialog box..... 213  
Add Watch dialog box..... 198  
Address Move dialog box ..... 132  
All trace mode..... 43  
ASCII display ..... 172  
Assemble Search dialog box ..... 166  
Assemble window ..... 157  
Auto load ..... 51

### B

Balloon watch function.....151, 164  
Break before execution..... 249  
Break dialog box..... 266  
Break function..... 41  
Breakpoint setting/deleting function.....143, 149, 162  
Browse dialog box ..... 127

### C

Cautions on re-downloading ..... 51  
Changing mnemonic.....147, 161  
comctl32.dll..... 32  
Come function .....36, 41, 149, 162  
Compulsion read ..... 207  
Configuration dialog box ..... 31, 81  
Correction of source file..... 51  
Current PC.....144, 159  
Current program ..... 294

### D

Debugger Option dialog box ..... 90  
Delay count..... 44  
Delay Count Setting dialog box ..... 281  
Delay trace .....44, 274  
Detailed display mode ..... 237  
Device driver..... 21  
Device file ..... 19  
Display file .....46, 109, 112  
Display font..... 91  
DMM dialog box..... 185  
DMM function ..... 187  
Download dialog box ..... 118

Drag-and-drop function ..... 164

### E

Emulation bus size ..... 83  
Error message list ..... 304  
Error/Warning dialog box..... 288  
Event before execution..... 247, 268, 269  
Event dialog box.....245  
Event function ..... 38  
Event icon ..... 237  
Event link..... 40, 256  
Event Link dialog box ..... 245, 256  
Event Manager..... 235  
Exit Debugger dialog box ..... 33, 286  
Extended Option dialog box ..... 86

### F

Font dialog box..... 100  
Format of symbol specification..... 138

### H

Host machine ..... 19

### I

IE ..... 19  
IE-70000-MC-NW-A ..... 18  
In-circuit emulator..... 18, 19  
Information file..... 46  
Initialize ..... 177  
Interface board..... 19  
Internal RAM ..... 35, 82  
Internal ROM..... 35, 82  
Interrupt Mask dialog box ..... 98  
I/O protect ..... 35, 84  
IOR Select dialog box..... 209  
IOR window ..... 206

### J

Jump function..... 150, 172, 220, 226, 238

### L

Load module ..... 118  
Load Module List dialog box..... 122  
Load/save function ..... 46  
Local Variable window..... 216

**M**

Main window .....	32, 56
Mapping function.....	35, 84
Mask .....	83
Memory Compare dialog box.....	181
Memory Compare Result dialog box.....	183
Memory Copy dialog box .....	179
Memory Fill dialog box.....	177
Memory Search dialog box .....	174
Memory window .....	169
Menu bar.....	63
Mixed display .....	142, 145
Mixed display mode .....	141, 147, 235
Modal dialog boxes .....	302
Modeless dialog boxes .....	302

**P**

Pass count .....	259
PM .....	49
Program code .....	144
Program counter setting function .....	149, 162
Program execution function .....	36
Programmable IOR .....	83, 207
Project file .....	33, 93, 104, 286
Project File Load dialog box.....	103
Project File Save dialog box .....	106
Project Manager.....	49

**Q**

Quick Watch dialog box .....	193
Quick watch function.....	151, 164

**R**

Real-time RAM sampling function.....	47, 87
Real-time trace mode.....	232
Register Select dialog box .....	204
Register window.....	201
Reset Debugger dialog box .....	283
ROM emulation RAM .....	35, 84
ROM probe .....	19, 35

**S**

Section trace .....	44, 274, 277
Software break.....	41, 242
Software break manager.....	242
Source debug.....	17, 20

Source Move dialog box .....	129
Source path .....	91
Source Search dialog box .....	154
Source window .....	140
Stack window .....	218
Start function .....	36, 149, 162
Startup symbol .....	95
Static state.....	142, 158, 170
Status condition .....	39, 249
Symbol To Address dialog box.....	137

**T**

Target .....	35, 84
TEU ....	39, 40, 247, 258, 260, 261, 268, 269, 275, 294
TIP.....	17
Toolbar .....	57, 93
Trace data .....	43
Trace Data Select dialog box .....	228
Trace dialog box.....	273
Trace function.....	43
Trace memory .....	87
Trace mode .....	43
Trace Move dialog box .....	134
Trace Option dialog box .....	43, 231
Trace View window.....	222
Tracer control mode .....	44
TRCU .....	36, 43, 273, 281, 294
Types of files that can be downloaded .....	120
Types of files that can be uploaded.....	125

**U**

Upload dialog box.....	124
------------------------	-----

**V**

Verification check .....	89
View file extension.....	110
View File Load dialog box.....	109
View File Save dialog box .....	112

**W**

Warning .....	288, 303
Watch function.....	150, 163
Watch window .....	188
Window link function.....	146, 160, 172, 227
Window list .....	53

## Facsimile Message

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

From:

Name

Company

Tel.

FAX

Address

**Thank you for your kind support.**

<p><b>North America</b>          NEC Electronics Inc.          Corporate Communications Dept.          Fax: 1-800-729-9288          1-408-588-6130</p>	<p><b>Hong Kong, Philippines, Oceania</b>          NEC Electronics Hong Kong Ltd.          Fax: +852-2886-9022/9044</p>	<p><b>Asian Nations except Philippines</b>          NEC Electronics Singapore Pte. Ltd.          Fax: +65-250-3583</p>
<p><b>Europe</b>          NEC Electronics (Europe) GmbH          Technical Documentation Dept.          Fax: +49-211-6503-274</p>	<p><b>Korea</b>          NEC Electronics Hong Kong Ltd.          Seoul Branch          Fax: 02-528-4411</p>	<p><b>Japan</b>          NEC Semiconductor Technical Hotline          Fax: 044-435-9608</p>
<p><b>South America</b>          NEC do Brasil S.A.          Fax: +55-11-6462-6829</p>	<p><b>Taiwan</b>          NEC Electronics Taiwan Ltd.          Fax: 02-2719-5951</p>	

I would like to report the following error/make the following suggestion:

Document title: \_\_\_\_\_

Document number: \_\_\_\_\_ Page number: \_\_\_\_\_

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Technical Accuracy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>