

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

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

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

April 1<sup>st</sup>, 2010  
Renesas Electronics Corporation

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

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

## Notice

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

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

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



# User's Manual

# ID850NWC Ver. 2.51

## Integrated Debugger

## Operation

---

### Target Device V850 Series™

Document No. U16525EJ1V0UM00 (1st edition)  
Date Published February 2003 CP(K)

© NEC Electronics Corporation 2003

Printed in Japan

[MEMO]

**V850 Series, V853, V850/SA1, V850/SB1, V850/SB2, V850/SC1, V850/SC2, V850/SC3, V850/SF1, V850/SV1, V850E/MS1, V850E/MA1, V850E/MA2, V850E/IA1, and V850E/IA2 are trademarks of NEC Electronics Corporation.**

**Windows and WindowsNT are either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.**

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

These commodities, technology or software, must be exported in accordance with the export administration regulations of the exporting country. Diversion contrary to the law of that country is prohibited.

- **The information in this document is current as of February, 2003. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics 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 the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics 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 Electronics 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 a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics 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 Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".

The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics 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 Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)

- (1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.
- (2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E 02.11-1

# Regional Information

Some information contained in this document may vary from country to country. Before using any NEC Electronics product in your application, please contact the NEC Electronics 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 America, Inc. (U.S.)**

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

## **NEC Electronics (Europe) GmbH**

Duesseldorf, Germany  
Tel: 0211-65 03 01  
Fax: 0211-65 03 327

### **• Sucursal en España**

Madrid, Spain  
Tel: 091-504 27 87  
Fax: 091-504 28 60

### **• Succursale Française**

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

### **• Filiale Italiana**

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

### **• Branch The Netherlands**

Eindhoven, The Netherlands  
Tel: 040-244 58 45  
Fax: 040-244 45 80

### **• Tyskland Filial**

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

### **• United Kingdom Branch**

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

## **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 Shanghai, Ltd.**

Shanghai, P.R. China  
Tel: 021-6841-1138  
Fax: 021-6841-1137

## **NEC Electronics Taiwan Ltd.**

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

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

Novena Square, Singapore  
Tel: 6253-8311  
Fax: 6250-3583

[MEMO]



## PREFACE

<b>Target Readers</b>	This manual is intended for user engineers who design and develop application systems of the V850 Series.	
<b>Purpose</b>	This manual is intended for users to understand the functions of the ID850NWC in the organization below.	
<b>Organization</b>	This manual consists of the following chapters: <ul style="list-style-type: none"><li>• OVERVIEW</li><li>• INSTALLATION</li><li>• STARTING AND TERMINATING</li><li>• ASSOCIATION WITH PM plus</li><li>• FUNCTIONS OF ID850NWC</li><li>• OPERATION</li><li>• WINDOW REFERENCE</li><li>• COMMAND REFERENCE</li></ul>	
<b>How to Use This Manual</b>	It is assumed that the readers of this manual have general knowledge of electrical engineering, logic circuits, microcontrollers, C language, and assemblers.  To understand the functions of the V850 Series → Refer to Hardware User's Manual for each product..  To understand the instruction functions of the V850 Series → Refer to <b>V850ES Architecture User's Manual (U15943E)</b> or <b>V850E1 Architecture User's Manual (U14559E)</b> .	
<b>Conventions</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
	<b>Remark:</b>	Supplementary information
	Numerical representation:	Binary ... XXXX or XXXXB
		Decimal ... XXXX
		Hexadecimal ... 0XXXXX
	Prefix indicating the power of 2 (address space, memory capacity):	
	K (Kilo): $2^{10} = 1024$	
	M (Mega): $2^{20} = 1024^2$	

**Related Documents**

Refer to the documents listed below when using this manual.

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 V853 <sup>TM</sup> , V850/SA1 <sup>TM</sup> , V850/SB1 <sup>TM</sup> , V850/SB2 <sup>TM</sup> , V850/SC1 <sup>TM</sup> , V850/SC2 <sup>TM</sup> , V850/SC3 <sup>TM</sup> , V850/SF1 <sup>TM</sup> , V850/SV1 <sup>TM</sup> )		U11595E
IE-V850E-MC (In-circuit emulator for V850E/IA1 <sup>TM</sup> , V850E/IA2 <sup>TM</sup> ), IE-V850E-MC-A (In-circuit emulator for V850E/MA1 <sup>TM</sup> , V850E/MA2 <sup>TM</sup> )		U14487E
IE-V850E1-CD-NW(The N-wire emulator for V850E1,V850ES cores which carried on-chip debug unit)		Under preparation
IE-703003-MC-EM1 (In-circuit emulator option board for V853)		U11596E
IE-703017-MC-EM1 (In-circuit emulator option board for V850/SA1)		U12898E
IE-703037-MC-EM1 (In-circuit emulator option board for V850/SB1, V850/SB2)		U14151E
IE-703040-MC-EM1 (In-circuit emulator option board for V850/SV1)		U14337E
IE-703079-MC-EM1 (In-circuit emulator option board for V850/SF1)		U15447E
IE-703102-MC (In-circuit emulator for V850E/MS1, V850E/MS2 <sup>TM</sup> )		U13875E
IE-703102-MC-EM1, IE-703102-MC-EM1-A (In-circuit emulator option board for V850E/MS1, V850E/MS2)		U13876E
IE-703107-MC-EM1 (In-circuit emulator option board for V850E/MA1)		U14481E
IE-703116-MC-EM1 (In-circuit emulator option board for V850E/IA1)		U14700E
CA850 Ver. 2.50 C Compiler Package	Operation	U16053E
	C Language	U16054E
	PM plus	U16055E
	Assembly Language	U16042E
ID850 Ver. 2.50 Integrated Debugger	Operation	U16217E
ID850NWC Ver. 2.51 Integrated Debugger	Operation	This manual
SM850 Ver. 2.50 System Simulator	Operation	U16218E
SM850 Ver. 2.00 or Later System Simulator	External Part User Open Interface Specifications	U14873E
RX850 Ver. 3.13 or Later Real-Time OS	Basics	U13430E
	Installation	U13410E
	Technical	U13431E
RX850 Pro Ver. 3.13 Real-Time OS	Fundamental	U13773E
	Installation	U13774E
	Technical	U13772E
RD850 Ver. 3.01 Task Debugger		U13737E
RD850 Pro Ver. 3.01 Task Debugger		U13916E
AZ850 Ver. 3.10 System Performance Analyzer		U14410E
PG-FP4 Flash Memory Programmer		U15260E

## CONTENTS

CHAPTER 1 OVERVIEW	16
1.1 Overview	16
1.2 Features	16
1.3 System Configuration	17
1.3.1 Example of system configuration	17
1.4 Operating Environment	18
1.4.1 Hardware environment	18
1.4.2 Software environment	18
1.5 Notes on Source-Level Debugging	18
CHAPTER 2 INSTALLATION	19
2.1 Installing ID850NWC	19
2.1.1 Installing procedure	19
2.2 ID850NWC Uninstalling	20
2.2.1 Uninstalling procedure	20
CHAPTER 3 STARTING AND TERMINATING	21
3.1 Cautions Before Starting	21
3.2 Starting	21
3.3 Terminating	23
CHAPTER 4 ASSOCIATION WITH PM plus	24
4.1 Setting Build Mode	24
4.2 Registering Debugger to PM plus Project	24
4.2.1 Selecting debugger	24
4.2.2 Downloading multiple load module files	25
4.3 To Start ID850NWC from PM plus	26
4.3.1 Restoring debugging environment	26
4.4 Auto Load	27
4.4.1 Auto load by correcting source code	27
4.4.2 Auto load by starting debugger	27
CHAPTER 5 FUNCTIONS OF ID850NWC	28
5.1 Mapping Function	28
5.2 Flash Memory Writing Function	29
5.3 Setting Debugging Environment	29
5.4 Program Execution Function	29
5.5 Break Function	31
5.6 Event Function	33
5.6.1 Using event function	33
5.6.2 Event condition	33
5.6.3 Event link condition	33
5.6.4 Setting of various event conditions	34
5.7 Register Manipulation Function	34
5.8 Memory Manipulation Function	34
5.9 Watch Function	34
5.10 Load/Save Function	35
5.10.1 view files	35
5.10.2 Information files	36
5.10.3 Setting files	36
5.11 RAM Sampling Function	36
CHAPTER 6 OPERATION	37
6.1 Flow of Debugging Operations	37
6.2 Active Status and Static Status	39
6.3 Jump Function	39
CHAPTER 7 WINDOW REFERENCE	41
7.1 Window List	41

7.2 This section explains each debug window as follows: ... 42

Main window	... 44
Configuration dialog box	... 62
Extended Option dialog box	... 68
Debugger Option dialog box	... 71
Font dialog box	... 78
Project File Save dialog box	... 80
Project File Load dialog box	... 83
View File Save dialog box	... 85
View File Load dialog box	... 88
Download dialog box	... 91
Upload dialog box	... 95
Load Module List dialog box	... 97
Browse dialog box	... 99
Source Text move dialog box	... 101
Address move dialog box	... 104
Symbol To Address dialog box	... 106
Source window	... 109
Source Search dialog box	... 121
Assemble window	... 124
Assemble Search dialog box	... 129
Memory window	... 132
Memory Search dialog box	... 137
Memory Fill dialog box	... 140
Memory Copy dialog box	... 142
Memory Compare dialog box	... 144
Memory Compare Result dialog box	... 146
DMM dialog box	... 148
Watch window	... 150
Quick Watch dialog box	... 156
Add Watch dialog box	... 159
Change Watch dialog box	... 163
Local Variable window	... 166
Stack Trace window	... 169
Register window	... 173
Register Select dialog box	... 176
IOR window	... 178
IOR Select dialog box	... 183
Add I/O Port dialog box	... 186
Event Manager	... 189
Software Break Manager	... 196
Event dialog box	... 199
Event Link dialog box	... 206
Break dialog box	... 212
Reset Debugger dialog box	... 216
Exit Debugger dialog box	... 218
About dialog box	... 220
Console window	... 221

CHAPTER 8 COMMAND REFERENCE ... 222

8.1 Command Line Rules	... 222
8.2 Command List	... 222
8.3 List of Variables	... 224
8.4 List of Packages	... 224
8.5 Key Bind	... 224
8.6 Expansion window	... 225
8.7 Callback Procedure	... 225
8.8 Hook Procedure	... 226
8.9 Related Files	... 227
8.10 Cautions	... 227
8.11 Commands	... 227
address	229
assemble	230

- batch 231
- breakpoint 232
- cache 234
- dbgexit 236
- download 237
- extwin 238
- finish 239
- go 240
- help 241
- hook 242
- ie 243
- jump 244
- mdi 246
- memory 247
- module 248
- next 249
- refresh 250
- register 251
- reset 252
- run 253
- step 254
- stop 255
- upload 256
- version 257
- watch 258
- where 259
- wish 260

## **APPENDIX A N-WIRE CHECKER ... 261**

- A.1 Overview ... 261
- A.2 Operation ... 262
- A.3 Explanation of each area and button ... 264
- A.4 Corrective Actions in Case of NG ... 266
- A.5 Restrictions ... 267

## **APPENDIX B EXPANSION WINDOW ... 268**

- B.1 Overview ... 268
- B.2 Activation ... 268
- B.3 Explanation of each sample window ... 268
  - List window 269
  - Grep window 270
  - RRM window 271
  - Hook window 272
  - Memory Mapped I/O window 274
  - Memory Mapped I/O dialog box 275

## **APPENDIX C INPUT CONVENTIONS ... 276**

- C.1 Character set ... 276
- C.2 Symbols ... 277
- C.3 Numeric Values ... 277
- C.4 Expressions and Operators ... 278

## **APPENDIX D TERMINOLOGY ... 281**

- D. 1 Terminology ... 281

## **APPENDIX E MESSAGES ... 283**

- E.1 Error/Warning Messages ... 283
  - E.1.1 Display format ... 283
  - E.1.2 Error messages at startup ... 284
  - E.1.3 Fatal error messages ... 285
  - E.1.4 Syntax error messages ... 291
  - E.1.5 Warning message or question message ... 309

**APPENDIX F KEY FUNCTION LIST ... 312**

- F.1Function List of Special Function Keys ... 312
- F.2Function List of Function Keys ... 313
- F.3Function List of Special Function Keys (Shift + key) ... 313
- F.4Function List of Function Keys (Shift + key) ... 314
- F.5List of Special Function Keys (Ctrl + key) ... 314
- F.6Function List of Function Keys (Ctrl + key) ... 314
- F.7Function List of Control Keys (Ctrl + key) ... 314
- F.8List of Special Function Keys (Ctrl + Shift + key) ... 315

**APPENDIX G INDEX ... 316**

## LIST OF FIGURES

### Figure No. Title Page

---

1-1	Example of ID850NWC System Configuration ...	17
3-1	Configuration Dialog Box ...	21
3-2	Main Window ...	22
3-3	Exit Debugger Dialog Box ...	23
3-4	Message Box for Confirmation of Stopping Execution ...	23
4-1	Debugger Settings Dialog Box (PM plus) ...	25
7-1	Main Window ...	44
7-2	Toolbar with Only Graphics (default) ...	45
7-3	Toolbar with Graphics and Characters ...	45
7-4	Status Bar ...	59
7-5	Configuration Dialog Box ...	62
7-6	Extended Option Dialog Box ...	68
7-7	Debugger Option Dialog Box ...	71
7-8	Add Source path Dialog Box ...	73
7-9	Font Dialog Box ...	78
7-10	Project File Save Dialog Box ...	80
7-11	Project File Load Dialog Box ...	83
7-12	View File Save Dialog Box (Example: to Save Memory Window) ...	85
7-13	View File Load Dialog Box ...	88
7-14	Download Dialog Box ...	91
7-15	The Progress of Download ...	94
7-16	Upload Dialog Box ...	95
7-17	Load Module List Dialog Box ...	97
7-18	Browse Dialog Box ...	99
7-19	Source Text Move Dialog Box ...	101
7-20	Address Move Dialog Box ...	104
7-21	Symbol To Address Dialog Box ...	106
7-22	Source Window ...	109
7-23	Source Search Dialog Box ...	121
7-24	Assemble Window ...	124
7-25	Assemble Search Dialog Box ...	129
7-26	Memory Window ...	132
7-27	Memory Search Dialog Box ...	137
7-28	Memory Fill Dialog Box ...	140
7-29	Memory Copy Dialog Box ...	142
7-30	Memory Compare Dialog Box ...	144
7-31	Confirmation Dialog Box ...	145
7-32	Memory Compare Result Dialog Box ...	146
7-33	DMM Dialog Box ...	148
7-34	Watch Window ...	150
7-35	Quick Watch Dialog Box ...	156
7-36	Add Watch Dialog Box ...	159
7-37	Change Watch Dialog Box ...	163
7-38	Local Variable Window ...	166
7-39	Stack Trace Window ...	169
7-40	Register Window ...	173
7-41	Register Select Dialog Box ...	176
7-42	IOR Window ...	178
7-43	IOR Select Dialog Box ...	183
7-44	Add I/O Port Dialog Box ...	186
7-45	Event Manager ...	189
7-46	Software Break Manager ...	196
7-47	Event Dialog Box ...	199
7-48	Event Link Dialog Box ...	206
7-49	Break Dialog Box ...	212
7-50	Reset Debugger Dialog Box ...	216
7-51	Exit Debugger Dialog Box ...	218
7-52	About Dialog Box ...	220

7-53	Console Window ... 221
A-1	N-Wire Checker ... 261
A-2	Waveform of N-Wire Checker ... 263
B-1	List Window ... 269
B-2	Grep Window ... 270
B-3	RRM Window ... 271
B-4	Hook Window ... 272
B-5	Memory Mapped I/O Window ... 274
B-6	Memory Mapped I/O Dialog Box ... 275
E-1	Error/Warning Messages ... 283



## LIST OF TABLES

### Table No. Title Page

---

5-1	Various Event Conditions ...	33
5-2	List of View Files ...	35
5-3	List of Information Files ...	36
5-4	List of Setting Files ...	36
7-1	Window List ...	41
7-2	Break Cause ...	60
7-3	Mapping Unit in Configuration Dialog Box ...	67
7-4	Contents Saved to Project File ...	80
7-5	Extension Corresponding to Current Window ...	86
7-6	Type of File That Can Be Loaded on View File Load Dialog Box ...	89
7-7	Type of File That Can Be Downloaded ...	92
7-8	Type of File That Can Be Uploaded ...	96
7-9	Type of File That Can Be Displayed in Browse Dialog Box ...	100
7-10	Specifying Symbols ...	107
7-11	Display Color of Breakpoint and Its Meaning ...	111
7-12	Details of Setting/Deleting Breakpoint ...	112
7-13	Meaning of Event Display Mark ...	112
7-14	Details of Drag & Drop Function (Line/Address) ...	119
7-15	Details of Drag & Drop Function (Character String) ...	120
7-16	Data Display Format (Watch-Related) ...	152
7-17	How a Variable Is Handled When a Scope Is Specified ...	160
7-18	Maximum Number of Valid Events for Each Event Condition ...	189
7-19	Marks of Event Icon ...	191
7-20	Color of Character on Mark and its Meaning ...	191
7-21	Address Condition and Jump Pointer ...	192
7-22	Separator for Displaying Event Details ...	193
7-23	List of Status Conditions ...	201
7-24	List of Access Size Conditions (Event) ...	201
7-25	Settable Range of Data Condition ...	203
7-26	The Number of Event Conditions in Event Link Dialog Box ...	208
7-27	Number of Events Settable in Break Condition Setting Area ...	213
8-1	List of Debugger Control Commands ...	222
8-2	List of Console/Tcl Commands ...	224
8-3	Message ID ...	225
A-1	Errors(N-Wire Checker) ...	266
B-1	List of Expansion Window ...	268
C-1	Usable Character Set ...	276
C-2	List of Special Characters ...	276
C-3	Input Format of Numeric Values ...	278
C-4	List of Operators ...	278
C-5	Operator Priority ...	279
C-6	Range of Radixes ...	280
D-1	Delimiter ...	281
F-1	Function List of Special Function Keys ...	312
F-2	Function List of Function Keys ...	313
F-3	Function List of Special Function Keys (Shift + key) ...	313
F-4	Function List of Function Keys (Shift + key) ...	314
F-5	List of Special Function Keys (Ctrl + key) ...	314
F-6	Function List of Function Keys (Ctrl + key) ...	314
F-7	Function List of Control Keys (Ctrl + key) ...	314
F-8	List of Special Function Keys (Ctrl + Shift + key) ...	315

## CHAPTER 1 OVERVIEW

### 1.1 Overview

The integrated debugger ID850NWC for the V850 Series. (hereafter referred to as the ID850NWC) is a software tool developed for NEC Electronics V850 Series. of 32-bit micro controllers for embedded control. This software tool is intended to enable efficient debugging of user programs.

### 1.2 Features

The ID850NWC has the following features:

#### (1) Source debugging

A C source program and source program in assembly language can be debugged.

#### (2) Support of on-chip debugging

A debugging function implemented by the on-chip debug unit of the Nx85ET(RCU0+TEU+TRCU), Nx85E901(RCU0), RCU1 is provided.

#### (3) Flash memory writing function

The internal flash memory can be written and the load module can be downloaded by the same access method as an ordinary memory operation.

#### (4) Security function

The ID code stored in the internal ROM or internal flash memory of a product with a security unit can be authenticated.

#### (5) Execution time measurement function

Time from the start of user program execution to occurrence of a break (Run-Break time) is measured in a range of 100 ns to 214 seconds.

#### (6) Automatic display updating function when execution is stopped

If execution of the user program is stopped, the values of the window displayed on the screen are automatically updated.

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

The debugging environment including information on setting of breakpoints and events, downloading files, and display status and position of windows can be saved to a file (project file).

By loading this project file, the debugging environment can be restored.

**(8) Batch execution by command and creation of custom window**

A console window that allows batch processing on the command line and creation of the user's own custom window has been added.

**1.3 System Configuration**

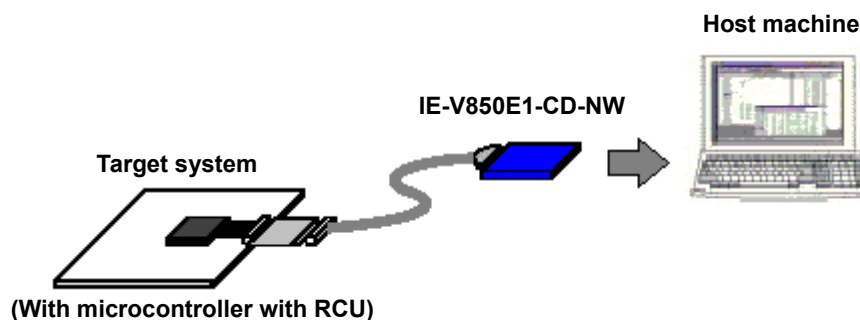
The ID850NWC provides a comfortable debugging environment for user programs and target systems developed for the V850 Series, by manipulating the PC card type emulator IE-V850E1-CD-NW that is inserted in the host machine.

The IE-V850E1-CD-NW is an emulator for on-chip debugging that provides debugging functions by being connected to the target system that incorporates the V850ES or V850E1 with a built-in RCU (Run Control Unit).

**1. 3. 1 Example of system configuration**

Figure 1-1 shows an example of the system configuration of the ID850NWC.

Figure 1-1 Example of ID850NWC System Configuration



## 1.4 Operating Environment

The ID850NWC has the following hardware and software requirements:

### 1. 4. 1 Hardware environment

#### (1) Host machine

- IBM PC/AT<sup>TM</sup>-compatible machine

**Caution** Because it is assumed that the ID850NWC is used with a notebook PC, the host machine must have a PC card slot (TYPEII).

#### (2) In-circuit emulator

- IE-V850E1-CD-NW

### 1. 4. 2 Software environment

#### (1) Operating system (any of the following)

Windows 98, Windows 2000, Windows NT 4.0 (Service Pack 5 or above), WindowsMe, WindowsXP Home Edition, WindowsXP Professional Edition

**Caution** Regardless of which of the OS's above is used, we recommend that the latest Service Pack is installed.

#### (2) device file (Individual acquisition)

- The device file of the target device to be used.

This file is available from the following Web site of NEC Electronics Corporation.

[http://www.necel.com/micro/index\\_e.html](http://www.necel.com/micro/index_e.html)

## 1.5 Notes on Source-Level Debugging

To perform source-level debugging, add an option (-g option) to generate debug information when the source file is compiled.

If this option is not added, the source file cannot be debugged at the source level.

In addition, the following processing must be performed for the ID850NWC.

- Embedding ID code (refer to "IE-V850E1-CD-NW User's Manual")
- Storing program in ROM (recommended) (refer to "CA850 C Compiler Package - Operation")

## CHAPTER 2 INSTALLATION

This chapter explains the procedure of installing and uninstalling ID850NWC.

### 2.1 Installing ID850NWC

To use the ID850NWC, the ID850NWC and a device file are necessary. Install the device file with the dedicated installer "DFINST.exe" (supplied with the ID850NWC).

#### 2. 1. 1 Installing procedure

To install the ID850NWC, prepare the following:

- ID850NWC system disk

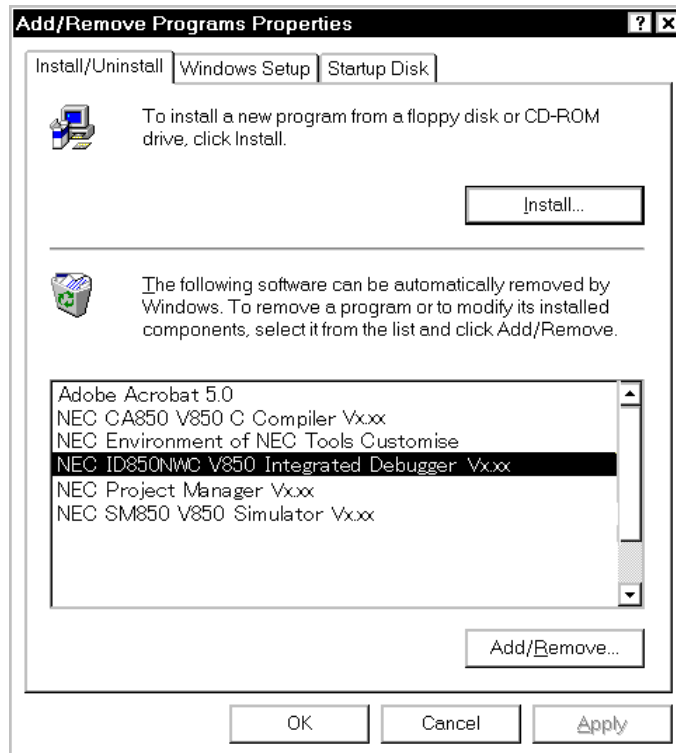
Install the ID850NWC by inserting the ID850NWC system disk in the CD-ROM drive and then following the installer's instructions.

**Caution** To install the ID850NWC again after the ID850NWC has been installed once, be sure to uninstall the ID850NWC. If the ID850NWC is installed in a directory different from that, without uninstalling, the ID850NWC that has already been installed cannot be uninstalled.

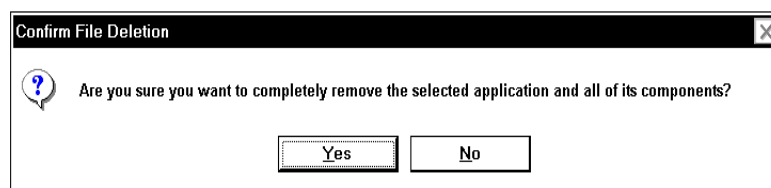
## 2.2 ID850NWC Uninstalling

### 2. 2. 1 Uninstalling procedure

- 1 Start [Add/Remove Programs] from the Control Panel.
- 2 Select [NEC ID850NWC V850 Integrated Debugger] from the list displayed at [Install and Uninstall], and click the < Add/Remove...> button.



- 3 [Confirm File Deletion] dialog box will be displayed. Click the <Yes > button.



- 4 The program will be deleted. When the message " Uninstalling is complete." is displayed, click the <OK> button.

## CHAPTER 3 STARTING AND TERMINATING

This chapter explains the procedure of starting and terminating the ID850NWC.

### 3.1 Cautions Before Starting

Start the N-Wire Checker before the starting the ID850NWC to check that the in-circuit emulator and the target system can be normally debugged (refer to "APPENDIX A N-WIRE CHECKER").

**Caution** For the connection between the in-circuit emulator and the target system and the power application sequence, refer to "IE-V850E1-CD-NW User's Manual". Incorrect connection may damage the in-circuit emulator and the target system.

### 3.2 Starting

- 1 Select [Start] -> [Program] -> [NEC Tools32] -> [ID850NWC] from the Windows start menu. Start the ID850NWC, the Configuration dialog box will be opened.  
In this case, the Configuration dialog box should not be displayed, but an error message should be displayed, please cope with it with reference to "E. 1. 2 Error messages at startup"

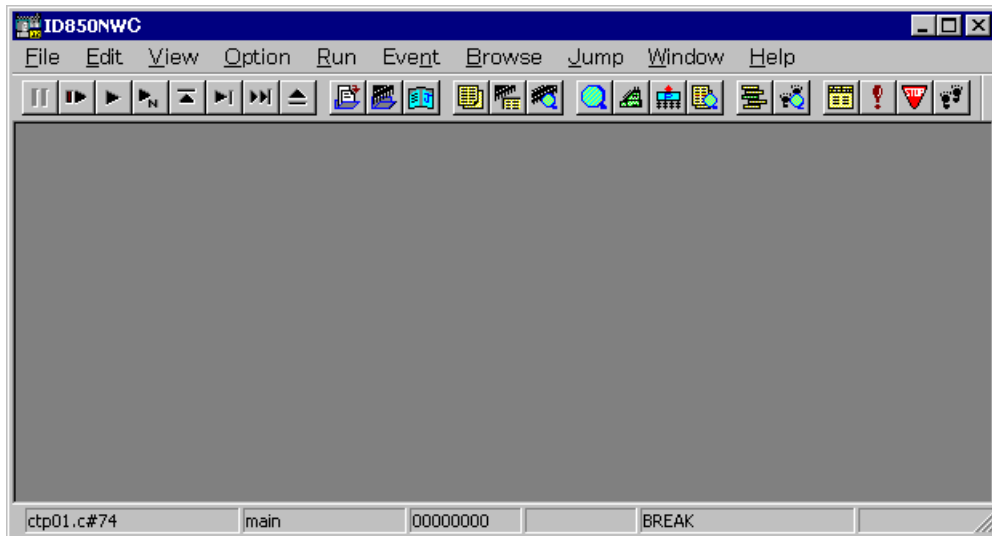
Figure 3-1 Configuration Dialog Box

The Configuration dialog box is titled "Configuration" and includes a close button (X) in the top right corner. It is organized into several sections:

- Chip:** Name: uPD 70F3259Y (dropdown menu).
- Internal ROM/RAM:** Internal ROM: 384\* KByte (dropdown menu), Internal RAM: 32768\* Byte (dropdown menu).
- Clock:** Main OSC (MHz): 5.000 (dropdown menu), Sub (KHz): 32.768 (dropdown menu).
- Programmable I/O Area:** Start Address: (text field).
- ID Code:** \*\*\*\*\* (text field).
- Peripheral Break:** Break (radio button), Non Break (radio button).
- Monitor Clock:** System (radio button), User (radio button).
- Mask:** NMIO (checkbox), NMI1 (checkbox), NMI2 (checkbox), STOP (checkbox), HOLD (checkbox), RESET (checkbox), DBINT (checkbox), WAIT (checkbox).
- Memory Mapping:** Access Size: 8Bit (radio button), 16Bit (radio button), 32Bit (radio button). Memory Attribute: (text field), Mapping Address: (text field), Target: (dropdown menu), Add (button), Delete (button).

- 2 Set the items related to the operating environment of the ID850NWC in the Configuration dialog box. After setting each item, click the <OK> button in the dialog box.
- 3 The Main window will be opened and the ID850NWC can be operated. Mainly use this window for debugging.

Figure 3-2 Main Window



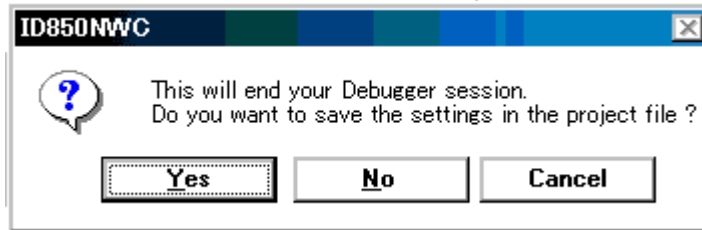


### 3.3 Terminating

Terminate the ID850NWC as follows:

- 1 Select [File] -> [Exit] on the Main window. The following Exit Debugger dialog box will be opened:

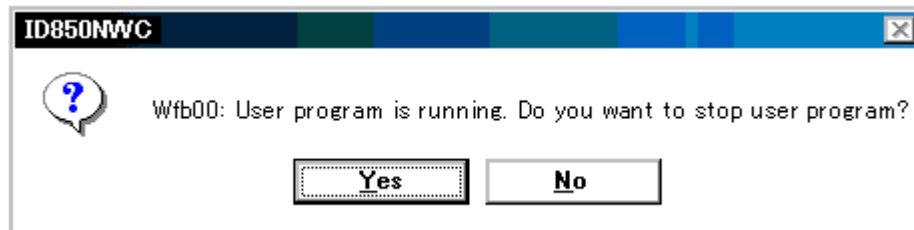
Figure 3-3 Exit Debugger Dialog Box



If the above operation is performed while the user program is being executed, the dialog box shown in Figure 3-4 is opened.

In this case, clicking the <Yes> button opens the Exit Debugger dialog box.

Figure 3-4 Message Box for Confirmation of Stopping Execution



- 2 To save the current debugging environment to a project file, click the <Yes> button. If the <No> button is clicked, all the windows are closed the ID850NWC terminated.

## CHAPTER 4 ASSOCIATION WITH PM plus

This chapter explains the procedure and points to be noted when a function to associate the PM plus is used.

The ID850NWC can automatically perform a series of operations in development processes, such as **creating source files -> compiling -> debugging -> correcting source files**, in association with the PM plus.

**Caution** If a load module file is created by using the Windows command prompt, the function to associate the ID850NWC with the PM plus cannot be used.

### 4.1 Setting Build Mode

To debug the load module file created by the PM plus on the ID850NWC at the source level, build to output symbol information for debugging must be performed to create a load module file. This setting can be performed by selecting [Debug Build] on the PM plus.

### 4.2 Registering Debugger to PM plus Project

The debugger to be used or the load module files to be downloaded can be specified for each project in the PM plus.

#### 4. 2. 1 Selecting debugger

The procedure for selecting the debugger is as follows:

##### Creating a new workspace

- 1 Select [File] menu -> [New Workspace...] on the PM plus.  
-> This opens the dialog box to create a new workspace using the wizard format.
- 2 Creating the necessary settings for the workspace with the wizard, the [New Workspace - Step 7/8 (Select Debugger)] dialog box will be opened. Specify [ID850NWC V850 Integrated Debugger] in this dialog box.  
-> For details of the setting, refer to the PM plus online help or user's manual.

##### Using an existing workspace

- 1 Select [Tool] menu -> [Debugger Settings...] on the PM plus.  
-> The Debugger Settings dialog box will be opened.
- 2 Specify [ID850NWC V850 Integrated Debugger] and click the <OK> button in this dialog box.

The ID850NWC is registered as the debugger of the active project.

The ID850NWC icon is displayed on the toolbar of the PM plus.

#### 4. 2. 2 Downloading multiple load module files

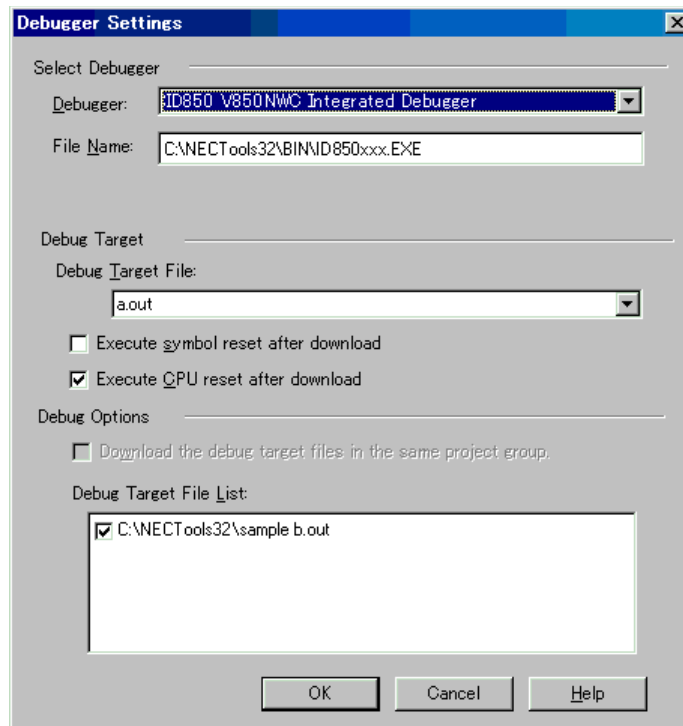
The load module files of the same project group are downloaded to the ID850NWC.

For details of the project group, refer to the PM plus online help or user's manual.

Download multiple load module files in the following steps.

- 1 Select [Tool] menu -> [Debugger Settings...] on the PM plus.  
-> The Debugger Settings dialog box will be opened.

Figure 4-1 Debugger Settings Dialog Box (PM plus)



- 2 Specify [ID850NWC V850 Integrated Debugger] in this dialog box.
- 3 Specify the load module file to be downloaded first in the [Debug Target File] when the ID850NWC is started.
- 4 Check the [Download the debug target files in the same project group] check box, and select the debug target files in the same group to be downloaded in the Debug Target File List.  
->The selected file will be downloaded right after the file specified in [Debug Target File] when the ID850NWC is started. Note that when these files are downloaded, symbol reset and CPU reset are not executed.
- 5 Click the <OK> button.

**Remark** Multiple load module files that are being downloaded can be checked in the Load Module List dialog box of the ID850NWC.

**Caution** If [execute symbol reset after download] in the [Debugger Settings...] dialog box of the product with a internal flash memory is checked, the contents of the internal flash memory are erased before downloading.

### 4.3 To Start ID850NWC from PM plus

The ID850NWC can be started from the PM plus as follows:

- Click the ID850NWC starting button on the toolbar of the PM plus.
- Select the [Build] menu -> [Debug] on the PM plus.
- Select the [Build] menu -> [Build and Debug] on the PM plus.
- Select the [Build] menu -> [Rebuild and Debug] on the PM plus.

If the debugging environment of the ID850NWC is saved to a project file currently being used by the PM plus, it will be started in the debugging environment saved in the project file.

If the debugging environment of the ID850NWC is not saved to a project file being used by the PM plus, the Configuration dialog box is opened. At this time, the device type (Chip name) cannot be changed.

#### 4. 3. 1 Restoring debugging environment

The previous debugging environment can be restored by the following procedure when the ID850NWC is started from the PM plus:

- 1 Create a new workspace (project file: e.g., sample.prj) on the PM plus.

**Caution** In the ID850NWC and PM plus, the environment information is saved to a project file and referenced. The extension of the project file that can be used by the ID850NWC and PM plus is ".prj". For the information that is saved or restored by the project file, refer to the online help or the user's manual of each product.

- 2 Start the ID850NWC from the PM plus. Because a new project file is created, set items other than the device type (Chip name) in the Configuration dialog box in the same manner as when only the ID850NWC is started.
- 3 Download the load module file to be debugged with the Download dialog box of the ID850NWC.
- 4 Debug the load module file on the ID850NWC.
- 5 Click the <Yes> button on the Exit Debugger dialog box when the ID850NWC is terminated.  
-> The debugging environment will be saved to the project file (sample.prj) for the PM plus when the ID850NWC is terminated (the debug environment can also be saved to the sample.prj file by overwriting the project file at times other than the completion of ID850NWC debugging).
- 6 When the ID850NWC is next started up after the sample.prj file is read by PM plus, the debug environment at the point when the project file was saved is automatically restored.

## 4.4 Auto Load

If a bug is found while the load module file is being debugged by the ID850NWC, correct the source file using the following procedure. Compiling and re-downloading the file can be automatically executed (Refer to "Auto load by correcting source code").

The load module is downloaded again to the ID850NWC by compiling and linking the file on the PM plus with the activated ID850NWC (refer to "Auto load by starting debugger").

### 4. 4. 1 Auto load by correcting source code

Correct the source file for auto load as follows:

- 1 Open the source file to be corrected in the Source window. Select [File] menu -> [Open] and specify the file to be corrected on the ID850NWC (if the file is already open in the Source window, that window is displayed in the forefront).  
-> The specified file will be opened in the Source window.
- 2 Select [Edit] menu -> [Correct Source] on the ID850NWC.  
-> An editor will be opened and the specified source file will be read.
- 3 Correct the source file on the editor.
- 4 Terminate the editor.

**Caution** The CPU reset is not performed when the load module file is automatically downloaded. The debug window that was opened when the editor was called, and each event setting will be restored. If the previously used line or symbol has been deleted as a result of correcting the source file, the following happens:

- A variable that was displayed is dimmed.
- The event mark of an event condition is displayed in yellow.

- 5 Select [Build] menu -> [Build and Debug], or [Build] -> [Rebuild and Debug] on the PM plus.

### 4. 4. 2 Auto load by starting debugger

If the following operation is performed on the PM plus with the ID850NWC started, the load module will be automatically downloaded to the ID850NWC.

- Selecting the [Build] menu -> [Build and Debug] on the PM plus.
- Selecting the [Build] menu -> [Rebuild and Debug] on the PM plus.

The CPU reset is not performed after the load module has been downloaded.

## CHAPTER 5 FUNCTIONS OF ID850NWC

This chapter explains the basic debugging function of the ID850NWC.

- Mapping Function
- Flash Memory Writing Function
- Setting Debugging Environment
- Program Execution Function
- Break Function
- Event Function
- Register Manipulation Function
- Memory Manipulation Function
- Watch Function
- Load/Save Function
- RAM Sampling Function

### 5.1 Mapping Function

The following types of mapping functions are available:

The mapping configuration is done by Configuration dialog box.

#### (1) User area mapping (Target)

The memory area specified for user area mapping becomes the area to access the memory in the target system or memory incorporated in the CPU.

#### (2) I/O Protect area

An I/O Protect area can be set in the area specified for the "target". This area is displayed in the same manner as an area that is not mapped (display symbol:??), on the Memory window. By mapping an area with this attribute, data cannot be read or written from/to this area by the Memory window, on the area can therefore be protected from an illegal access.

To read or write the value of the area mapped with this attribute, register the value in the IOR window or Watch window.

## 5.2 Flash Memory Writing Function

With the ID850NWC, the internal flash memory can be written and the load module can be downloaded by the same access method as an ordinary memory operation.

The data on the internal flash memory can be changed from the Memory window, Assemble window, Watch window, Memory Fill dialog box, and Memory Copy dialog box, without having to be aware that the data is that of the internal flash memory. The load module can also be downloaded to the internal flash memory by using the flash self-programming function.

**Caution** No data can be written to the internal flash memory during user program execution.

**Remark** With the ID850NWC, the remaining area contents after the load module was downloaded to the internal flash memory are erased.

## 5.3 Setting Debugging Environment

Set a debugging environment with the Configuration dialog box that is opened when the debugger is started, or the Extended Option dialog box or Debugger Option dialog box.

By creating a file (project file) to which the current debugging environment is saved during debugging, and loading this file, the debugging environment when the project file is saved can be restored.

To manipulate the project file, use the Project File Save dialog box and Project File Load dialog box.

After the project file has been loaded, the size and position of the displayed window are returned to the previous size and status (refer to Table 7-4 Contents Saved to Project File).

To load the project file when starting up the debugger, press the <Project...> button in the Configuration dialog box. It can also be specified that the project file is automatically loaded each time the debugger is started up (refer to To automatically load a project file).

## 5.4 Program Execution Function

The ID850NWC has the following execution functions. Manipulation of these functions is done with the [run] menu or the function buttons.

- [Go] (Go button)
- [Ignore break points and Go] (GoN button)
- [Return Out] (Ret button)
- [Step In] (Step button)
- [Next Over] (Over button)
- [Start From Here]
- [Come Here]
- [Go and Go]
- [Slowmotion]
- [Restart] (ReGo button)

**(1) [Go] (Go button)**

The user program is executed starting from the address indicated by the current PC register and is stopped if a set break event condition is satisfied.

**(2) [Ignore break points and Go] (GoN button)**

This command executes the user program starting from the address indicated by the current PC register. Execution of the user program continues, ignoring set breakpoints.

**(3) [Return Out] (Ret button)**

The user program is executed until execution returns to the calling function described in C language.

**(4) [Step In] (Step button)**

The operation differs depending on whether this function is executed in the Source window or the Assemble window.

**In the Source window**

Step execution of one line of the source text is performed starting from the current PC register value and the contents of each window are updated.

**In the Assemble window**

One instruction is executed from the current PC register value and the contents of each window are updated.

**(5) [Next Over] (Over button)**

The operation differs depending on whether the instruction to be executed is the jarl instruction or any other instruction.

**jarl instruction**

Next step execution is performed, assuming the function or subroutine called by the jarl instruction as one step (step execution continues until the nesting level becomes the same as when the jarl instruction was executed).

**Instruction other than jarl**

The same processing as [Step In] is performed.

**(6) [Start From Here]**

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

**(7) [Come Here]**

The user program is executed from the address indicated by the current PC register to the address selected in the line/address display area of the Source window or the Assemble window, and then a break occurs.

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



**(8) [Go and Go]**

The user program is executed starting from the address indicated by the current PC register and stopped if a set break event condition is satisfied. The contents of each window are updated, and execution of the user program is resumed from the address at which the program was stopped. This operation is repeated until the user executes [Stop].

**(9) [Slowmotion]**

Step execution of one line is performed from the address indicated by the current PC register value in the source mode. In the instruction mode, step execution of one instruction is performed. The contents of each window are updated each time step execution is performed.

This operation is repeated until the user executes [Stop].

**(10) [Restart] (ReGo button)**

The CPU is reset and the user program is executed starting from address 0.

This is the same operation as "resetting the CPU before execution of the user program and executing [Go]".

**5.5 Break Function**

The ID850NWC has the following break functions.

- Event detection break
- Break by [Come Here]
- Software break
- Break on satisfaction of condition of step execution
- Forced break ([Stop], Stop button)

**Caution** The event detection break and Software break are not performed when [Come Here], [Step In], [Return Out], [Next Over], or [Ignore break points and Go] described in "Program Execution Function" is executed.

**(1) Event detection break**

Event detection break is a function used to stop execution of the user program by detecting a set break event condition. This break is valid for [Go], [Go and Go], [Start From Here], and [Restart].

In the case of [Automatically Continue Execution], however, the contents of each window are updated and the user program is executed again after the event detection break has occurred. Set a break event as follows:

- Set as an execution event in the Source window or Assemble window (refer to Breakpoint setting/deletion function)
- Set in Break dialog box.

**Caution** Event detection break point set/overwrite/delete is possible during user program execution. In that case, The warning of a purport which makes a user program once take a break is displayed.

## (2) Break by [Come Here]

The function stops execution of a user program executed with the [Come Here] upon detection of the address specified in the Source window or Assemble window. After user program execution has been stopped, the breakpoint by this function is eliminated.

Moreover, during execution of a user program by [Come Here], break events set before the cursor position does not occur.

## (3) Software break

Software break is a function to replace the instruction at the specified address software break instruction and stop the user program executed by the [Go], [Go and Go], [Start From Here], or [Restart] command.

As many software break events as necessary can be set (however, only 2000 of them are valid).

Because the instruction at the specified address is rewritten, however, a software break event cannot be set on the external ROM or stopped by the access timing of a variable.

While one event detection break uses one hardware resource per event condition, the feature of a software break is that a breakpoint is set at two or more addresses.

Software break events are set in the Source window or Assemble window. (Refer to Breakpoint setting/deletion function.)

The Software Break Manager manages the software breaks (by validating or invalidating).

**Cautions 1** The 2000 valid addresses include the valid number of addresses when a software break is set in the internal ROM and internal flash memory. Because the ID850NWC realizes setting of software breaks in these areas by using the ROM correction function, the max number of addresses depends on the product type (0, 4, or 8).

**Caution 2** Software breaks can be set or deleted while the user program is being executed.

In that case, The warning of a purport which makes a user program once take a break is displayed.

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

Break on satisfaction of condition of step execution is a function used to stop execution of the user program when the end condition of each command ([Step In], [Next Over], [Return Out], and [Slowmotion]) is satisfied.

Because one instruction is executed, stopped, and the condition is confirmed at one time, the processing time is longer than that of real-time execution.

## (5) Forced break ([Stop], Stop button)

Forced break is a function used to forcibly stop execution of the user program. It is valid for all the execution commands.

## 5.6 Event Function

An event is a specific status of the target system during user program execution, such as "address 0x1000 is fetched" or "data is written to address 0x2000".

The ID850NWC uses an event to trigger an action of functions such as break functions. An event function is used to set, delete, and reference a specific status of the target system that triggers an action (debugging action) as an event condition.

### 5. 6. 1 Using event function

So that an action that is triggered by an event function in accordance with the user's debugging aim, the following various event conditions must be set.

Table 5-1 Various Event Conditions

Set Condition	Setting Dialog Box	Contents
Break event condition	Break dialog box	Condition in which the execution of the user program is stopped.

These various event conditions are set independently as an Event condition or Event link condition, or as a combination of these. To use the event function of the ID850NWC, therefore, first create the necessary Event condition and Event link condition, and set the various event conditions shown in the Table 5-1 by using these conditions.

### 5. 6. 2 Event condition

Create an event condition in the Event dialog box.

Set an address condition, status condition, and data condition in this dialog box. Specify a combination of these as one event condition and name and register this event condition. The registered event condition can be used to set various event conditions. However, the number of event conditions that can be set is limited.

The event condition that is created when a breakpoint is set in the Source window or the Assemble window can also be used (refer to "Breakpoint setting/deletion function").

### 5. 6. 3 Event link condition

An event link condition specifies sequential constraints for each event, so that two or more events are treated as a single event.

To create an event link condition, use the Event Link dialog box.

By arranging already registered event conditions in this dialog box, one event link condition can be named and registered. The registered event link condition can be used to set various event conditions in the same manner as the event condition. However, the number of event conditions that can be set is limited.

#### **5. 6. 4 Setting of various event conditions**

Various event conditions listed in Table 5-1 are individually created in the corresponding dialog box.

To create various event condition, drag and drop the event icon of an event condition or event link condition registered on the Event Manager to the condition area of each setting dialog box.

The created various event condition is set by clicking the <Set> or <OK> button in the setting dialog box. The mark of the event that has been set is displayed in red. After the event has been set, a debugging action occurs as various event conditions.

Up to 256 various event conditions can be registered. The number of various event conditions that can be "set" differs depending on the various event condition (refer to "Table 7-18 Maximum Number of Valid Events for Each Event Condition").

#### **5.7 Register Manipulation Function**

The register manipulation function is used to display or change the contents of the program registers, system registers, and peripheral I/O registers.

This function can be used in the Register window or IOR window. To change the register to be displayed in each window, use the Register Select dialog box or IOR Select dialog box.

A user-defined I/O port can be displayed or changed by registering it to the Add I/O Port dialog box. With a product that supports programmable I/O registers, the programmable I/O registers can be displayed or changed by specifying the use of the programmable I/O area in the Configuration dialog box.

#### **5.8 Memory Manipulation Function**

As a memory manipulation function, the memory contents can be displayed or changed by using mnemonic codes, hexadecimal codes, and ASCII codes.

This function can be used in the Memory window or Assemble window. The contents of memory can be copied, initialized, and compared.

Note that the memory contents can also be rewritten in the DMM dialog box.

#### **5.9 Watch Function**

Data (variable) values can be checked or changed in the Watch window.

Local variables can be checked or changed in the Local Variable window.

The contents of a variable value can be easily referenced in the Source window or Assemble window by placing the mouse cursor on the selected variable (balloon watch function).

## 5.10 Load/Save Function

The load/save function is used to load/save each file. The ID850NWC loads/saves each file individually. These three types of files can be handled:

<b>view files</b>	File recording the screen information when it is saved. By loading this file, a reference file is opened in the Source window.
<b>information file</b>	File recording the set data and debugging environment as information on the user program and ID850NWC.
<b>setting files</b>	File recording the window information.

### 5. 10. 1 view files

The list of the view files to be handled is shown below:

These files can be manipulated in the View File Load dialog box and View File Save dialog box.

Table 5-2 List of View Files

File Type	Contents
Source Text (*.SVW)	Source window view file Records the source text in the Source window.
Assemble (*.DIS)	Assemble window view file Records the assembly information in the Assemble window.
Memory (*.MEM)	Memory window view file Records the memory information in the Memory window.
Register (*.RGW)	Register window view file Records the register information in the Register window.
Local Variable (*.LOC)	Local Variable window view file Records the local variable information in the Local Variable window.
Stack Trace (*.STK)	Stack Trace window view file Records the stack trace information in the Stack Trace window.
Console (*.LOG)	Console window view file Records the information in the Console window.

### 5. 10. 2 Information files

The list of the information files to be handled is shown below.

Table 5-3 List of Information Files

File Name (Extension)	Contents	Dialog Box
Object file (*.HEX)	Stores the object codes of the user program.	Download dialog box, Upload dialog box
Load module file (*.OUT)	Stores the object codes, symbols, and source information of the user program.	Download dialog box
project file (*.PRJ)	Stores the debugging environment	Project File Load dialog box, Project File Save dialog box
Binary file (*.BIN)	Records data in binary format.	Download dialog box, Upload dialog box

A project file sets the display information, size, and display position of a window or dialog box, and restores the debugging environment to the previous status (when the project file was saved) when it is loaded.

### 5. 10. 3 Setting files

The list of the setting files to be handled is shown below:

These files can be manipulated in the View File Load dialog box and View File Save dialog box.

Table 5-4 List of Setting Files

File Name (Extension)	Contents
Watch setting file (*.WCH)	Stores the watch setting contents of the Watch window.
Peripheral I/O register setting file (*.IOR)	Stores the peripheral I/O register setting contents of the IOR window.
Event setting file (*.EVN)	Stores the event setting contents of the Event Manager.

## 5.11 RAM Sampling Function

The RAM sampling functions are used to read and display the memory contents, even during user program execution, if a variable allocated to a range in which the memory contents can be read, or the memory contents are displayed.

This function can be checked in the Watch window, Memory window or IOR window.

Settings related to this function are performed in the Extended Option dialog box.

## CHAPTER 6 OPERATION

This chapter explains the flow of basic debugging operations using the ID850NWC and the following functions:

- Flow of Debugging Operations
- Active Status and Static Status
- Jump Function

### 6.1 Flow of Debugging Operations

#### (1) Setting debugging environment

Prepare for debugging.

-> Configuration dialog box

Download a load module file to be debugged.

-> Download dialog box

Checking the loaded file.

-> Load Module List dialog box

#### (2) Displaying source program

By downloading a load module file with debug information, the corresponding source file is automatically displayed.

-> Source window

#### (3) Displaying disassemble result

The disassembled result of the downloaded user program is displayed.

-> Assemble window

#### (4) Setting breakpoint

Breakpoints can be set easily in the Source window, Assemble window, and Watch window.

-> Breakpoint setting/deletion function

-> To set a breakpoint for read/write access to a variable

#### (5) Setting event

Register event conditions and event link conditions.

-> Event dialog box, Event Link dialog box

Setting of various event conditions

Set a various event condition by using the Event Manager or an event icon in the event manager area on each various event setting dialog box.

Each various event condition corresponds to the following various event setting dialog box.

Break event condition -> Break dialog box

Checking setting status of various event conditions

-> Event dialog box or the event manager area in each various event setting dialog box

Checking setting status of software breakpoint

-> Software Break Manager

**(6) Executing user program**

By executing the user program, the PC advances to the set breakpoint, break event, or forced breakpoint (refer to "Program Execution Function").

Use the execution control button or [Run] menu to execute the user program.

**(7) Editing, checking, and changing memory contents**

-> Memory window, Memory Copy dialog box, Memory Fill dialog box, Memory Compare dialog box, Memory Compare Result dialog box, DMM dialog box

**(8) Registering, checking, and changing variable value**

-> Watch window, Quick Watch dialog box, Add Watch dialog box, Change Watch dialog box, Local Variable window

**(9) Checking stack contents**

-> Stack Trace window

**(10) Checking and changing of registered contents of register value**

-> Register window, Register Select dialog box

The value of the PC register can also be checked in the Source window and Assemble window.

**(11) Checking and changing peripheral I/O register value**

-> IOR window, IOR Select dialog box

**(12) Registering, checking, and changing user-defined I/O port value**

-> Add I/O Port dialog box, IOR window

**(13) Changing mnemonic and executing online assembly**

-> Assemble window, Source window

**(14) Searching character string in each window**

By clicking the < Search...> button in the window in which searching is to be executed, the following types of searching dialog boxes are opened:

Source Search dialog box, Assemble Search dialog box, Memory Search dialog box

**(15) Manipulating view file**

-> View File Save dialog box, View File Load dialog box



**(16) Uploading memory contents and coverage result**

-> Upload dialog box

**(17) Saving debugging environment to project file**

-> Project File Save dialog box

**(18) Terminating ID850NWC**

-> Exit Debugger dialog box

**6.2 Active Status and Static Status**

The Source window, Assemble window, and Memory window have the following two window statuses:

<b>Active status</b>	The display position and contents of the window in the active status are automatically updated in association with the current PC value. This window is also the jump destination of the jump function. Only one window can be opened in the active status. Select this status by the [Window] menu.
<b>Static status</b>	The display position of the window in the static status does not move in association with the current PC value, but the displayed contents are updated. The static window is not used as the jump destination of the jump function. In addition. If an active window is already open, the next window is opened in the static status. Two or more static windows can be opened at the same time. Select this status by the [Window] menu.

Because two or more static windows can be opened, the current status of the windows can be temporarily held.

**6.3 Jump Function**

The jump function is used to jump to Source window, Assemble window, or Memory window from a jump pointer that is the line or address on which the cursor is placed, and display the contents of the jump destination window from the jump pointer.

You can jump among the above windows, or from Stack Trace window, Event Manager, and Register window to the above windows.

**Jump method**

- 1 Move the cursor to the line or address that is to be used as the jump pointer, on the window from which jumping is possible (select an event icon on the Event Manager).
- 2 Select the following menu item to which execution is to jump from the [Jump] menu.  
Source window->Select [Source].  
Assemble window->Select [Assemble].  
Memory window->Select [Memory].

**Caution** If a program code does not exist on the line at the cursor position, the first address of the line with a program code above or below that line is used as the jump pointer.

#### **Details of jump source address**

To jump from the following windows, the jump destination is as follows:

- To jump from the Register window, move the cursor to a register value.
- To jump from the Stack Trace window, the jump destination is as follows:

##### **[With current function]**

- If the jump destination is the Source window  
-> Jumps to the current PC line.
- If the jump destination is a window other than the Source window  
-> Jumps to the current PC address.

##### **[With function other than current function]**

- If the jump destination is the Source window  
-> Jumps to the line that calls a nested function.
- If the jump destination is a window other than the Source window  
-> Jumps to the address next to the instruction that calls a nested function.

## CHAPTER 7 WINDOW REFERENCE

This chapter explains the details of the window/dialog box functions of the ID850NWC.

### 7.1 Window List

Table 7-1 lists the windows of the ID850NWC.

Table 7-1 Window List

Window Name	Contents
Main window	This window is displayed first, when the ID850NWC is started. It controls execution of the user program. Various windows are opened from this window.
Configuration dialog box	Displays and sets the ID850NWC operation environment.
Extended Option dialog box	This dialog box is used to display and set the extended options of the ID850NWC.
Debugger Option dialog box	Displays and sets options.
Font dialog box	Displays the types of fonts displayed in the Source window and sets the size of the font.
Project File Save dialog box	Saves the debug environment.
Project File Load dialog box	Loads the debug environment.
View File Save dialog box	Saves the contents of the current window to a view file or event setting file.
View File Load dialog box	Reads the view file or source event setting file of each window.
Download dialog box	Loads an object file and binary file.
Upload dialog box	Saves the memory contents to a file.
Load Module List dialog box	Lists the names and file path names of the downloaded load module files.
Browse dialog box	Debugger Option dialog box, Select the file to be set in Source Search dialog box.
Source Text move dialog box	This dialog box is used to specify a file to be displayed in Source window and the position from which displaying the file is to be started.
Address move dialog box	Specifies the start address to display the contents of the Memory window, or Assemble window.
Symbol To Address dialog box	Displays the addresses and symbol values of variables and functions.
Source window	Displays a source file or text file.
Source Search dialog box	Searches for a character string in a source file.
Assemble window	Disassembles the program and executes online assembly.
Assemble Search dialog box	Searches the contents of the Assemble window.
Memory window	Displays and modifies the contents of memory.
Memory Search dialog box	Searches the contents of the Memory window.
Memory Fill dialog box	Initializes the memory contents with specified data.
Memory Copy dialog box	Copies the memory.

Table 7-1 Window List

Window Name	Contents
Memory Compare dialog box	Compares the memory.
Memory Compare Result dialog box	Displays the result of comparing the memory.
DMM dialog box	Sets addresses and data subject to DMM.
Watch window	Displays and modifies variables.
Quick Watch dialog box	Temporarily displays and changes the value of a variable.
Add Watch dialog box	Registers a variable to be displayed in the Watch window.
Change Watch dialog box	Changes a variable to be displayed in the Watch window.
Local Variable window	Displays and changes the local variable in the current function.
Stack Trace window	Displays and changes the stack contents of the current user program.
Register window	Displays and changes the contents of the registers (program registers and system registers).
Register Select dialog box	Selects a register to be displayed in the Register window.
IOR window	Displays and changes the contents of the peripheral I/O registers.
IOR Select dialog box	Selects a peripheral I/O register or I/O port to be displayed in the IOR window.
Add I/O Port dialog box	Registers an I/O port to be displayed in the IOR window.
Event Manager	Displays, enables/disables, and deletes each event condition.
Event dialog box	Registers and displays event conditions.
Event Link dialog box	Registers and displays event link conditions.
Break dialog box	Registers, sets, and displays break event conditions.
Software Break Manager	Display, enable or disable, and delete software breaks.
Reset Debugger dialog box	Initializes the ID850NWC, target CPU, and symbol information.
About dialog box	Displays the version of the ID850NWC.
Exit Debugger dialog box	Terminate the ID850NWC.
Console window	Inputs commands.
Error/Warning dialog box	Displays error/warning messages.

## 7.2 This section explains each debug window as follows:

**Window name**

Briefly explains the function of the window or dialog box and points to be noted.

In addition, the display image of the window or dialog box is also illustrated.

Items of related operation are also explained.

**Opening**

---

Explains how to open the window or dialog box.

If two or more methods of opening are listed, the window or dialog box can be opened by any of them.

**Explanation of each area**

---

Explains items to be set to or displayed in each area of the window or dialog box.

**Functions often used (right-click menu)**

---

Explains the context menu that is displayed in the window when the right mouse button is clicked.

From the context menu, convenient functions often used in this window can be selected with a single action (window only).

**Function buttons**

---

Explains the operation of each button in the window or dialog box.

**Related operations**

---

Explains the operation of a window or dialog box related to this window or dialog box.

## Main window

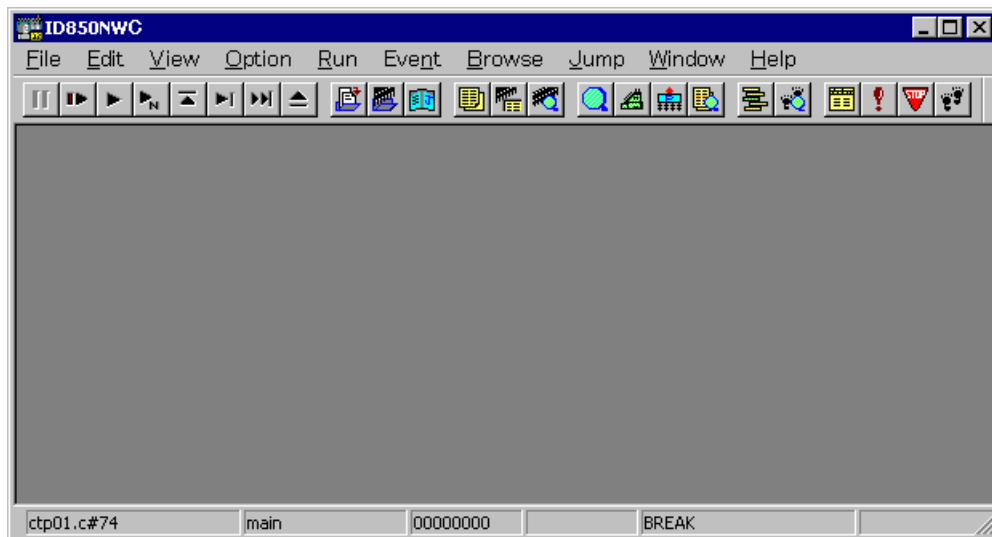
This window is automatically opened when the ID850NWC is started up and initialized. It exists on the screen until the ID850NWC is terminated. Other windows are manipulated from this window. Execution of the user program is controlled in this window.

Execution of the user program is controlled in the following three modes:

Source mode	Debugs the user program at the source level.
Instruction mode	Debugs the user program at the instruction level.
Auto mode	Automatically selects the source mode or instruction mode.

The auto mode is set when the debugger is started up.

Figure 7-1 Main Window



This section explains the following items:

- Toolbar
- Menu bar
- Window display area
- Status bar

### Toolbar

The toolbar is a collection of buttons that can execute often-used commands with a single action. Each button is graphically displayed to indicate the image of its function. This toolbar can be displayed in the following two modes. The modes are selected in Debugger Option dialog box.

Figure 7-2 Toolbar with Only Graphics (default)














Figure 7-3 Toolbar with Graphics and Characters



Whether the toolbar is displayed or not can be specified by selecting [Option] -> [Tool Bar] from the menu bar. This toolbar can be moved inside or outside the main window by dragging and dropping with the mouse.

The meaning of each button on the toolbar is as follows. When the mouse cursor is placed on a button of the toolbar, a tool hint pops up several seconds later.

 <b>Stop</b>	Stops execution of the user program. Same function as [Run] -> [Stop] on the menu bar.
 <b>ReGo</b>	Resets the CPU and executes the user program. Same function as [Run] -> [Restart] on the menu bar.
 <b>Go</b>	Executes the user program from the current PC without resetting the CPU. Same function as [Run] -> [Go] on the menu bar.
 <b>Go</b>	Executes the user program without a break occurrence. Same function as [Run] -> [Ignore break points and Go] on the menu bar.
 <b>Ret</b>	The user program is real-time executed until execution returns. Same function as [Run] -> [Return Out] on the menu bar * This command is used for a function described in C language.
 <b>Step</b>	Step execution (executes instructions in the program one by one.) If a function or subroutine is called, its instructions are executed one by one. Same function as [Run] -> [Step In] on the menu bar.
 <b>Over</b>	Next step execution (executes the program, assuming a function/call statement as one step.) If a function or subroutine is called, its instructions are not executed on a step-by-step basis. Same function as [Run] -> [Next Over] on the menu bar.
 <b>Res</b>	Resets the CPU. Same function as [Run] -> [CPU Reset] on the menu bar.
 <b>Open</b>	Opens View File Load dialog box. Same function as [File] -> [Open...] on the menu bar.
 <b>Load</b>	Opens Download dialog box. Same function as [File] -> [Download...] on the menu bar.
 <b>Proj</b>	Opens Project File Load dialog box. Same function as [File] -> [Project] -> [Open...] on the menu bar.

 <b>Src</b>	Displays the source text. Opens Source window. Same function as [Browse] -> [Source Text] on the menu bar.
 <b>Asm</b>	Displays the disassemble results. Opens Assemble window. Same function as [Browse] -> [Assemble] on the menu bar.
 <b>Mem</b>	Displays the contents of the memory. Opens Memory window. Same function as [Browse] -> [Memory] on the menu bar.
 <b>Wch</b>	Displays the watch contents. Opens Watch window. Same function as [Browse] -> [Watch] on the menu bar.
 <b>Reg</b>	Displays the register contents. Opens Register window. Same function as [Browse] -> [Register] on the menu bar.
 <b>IOR</b>	Displays the contents of the Peripheral I/O registers. Opens IOR window. Same function as [Browse] -> [I/O Register] on the menu bar.
 <b>Loc</b>	Displays the local variable contents. Opens Local Variable window. Same function as [Browse] -> [Local Variable] on the menu bar.
 <b>Stk</b>	Displays the stack contents. Opens Stack Trace window. Same function as [Browse] -> [Stack Trace] on the menu bar.
 <b>TrW</b>	Cannot be selected.
 <b>Mgr</b>	Opens Event Manager. Same function as [Event] -> [Event Manager] on the menu bar.
 <b>Evn</b>	Registers and sets events. Opens Event dialog box. Same function as [Event] -> [Event...] on the menu bar.
 <b>Brk</b>	Registers and sets break events. Opens Break dialog box. Same function as [Event] -> [Break...] on the menu bar.
 <b>Trc</b>	Cannot be selected.

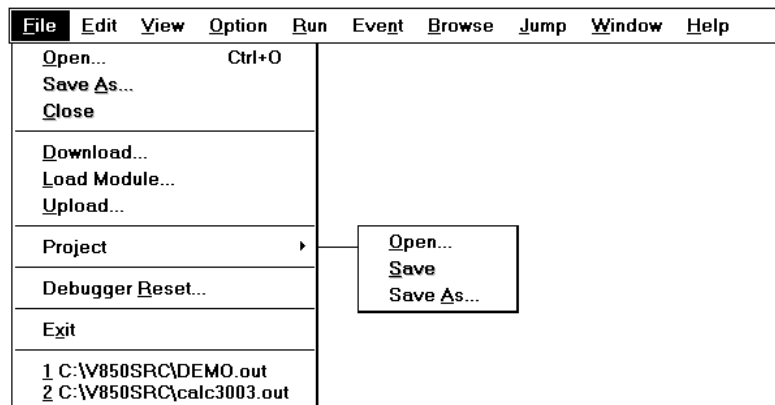


## Menu bar

The menu bar consists of the following menus:

- (1) [File] menu
- (2) [Edit] menu
- (3) [View] menu
- (4) [Option] menu
- (5) [Run] menu
- (6) [Event] menu
- (7) [Browse] menu
- (8) [Jump] menu
- (9) [Window] menu
- (10) [Help] menu

### (1) [File] menu



<a href="#">Open...</a>	Loads a view file, source file, or text file. Opens View File Load dialog box. The operation differs depending on the extension of the file selected in the dialog box.
<a href="#">Save As...</a>	Saves the contents displayed on the current window to the file whose name is specified. Opens View File Save dialog box.
<a href="#">Close</a>	Closes the current window.
<a href="#">Download...</a>	Downloads a program. Opens Download dialog box.
<a href="#">Load Module...</a>	Lists the names of the files that have been downloaded. Opens Load Module List dialog box.
<a href="#">Upload...</a>	Uploads a program. Opens Upload dialog box.
<a href="#">Project</a>	Manipulates a project file.

<a href="#">Open...</a>	Opens a project file. Opens Project File Load dialog box.
<a href="#">Save</a>	Overwrites the current status to the project file currently being read to the ID850NWC.
<a href="#">Save As...</a>	Saves the current status to a specified project file. Opens Project File Save dialog box.
<a href="#">Debugger Reset...</a>	Initializes the CPU, symbols, and ID850NWC. Opens Reset Debugger dialog box.
<a href="#">Exit</a>	Terminates the ID850NWC. Opens Exit Debugger dialog box.
<a href="#">(Open file)</a>	Lists the names of the files opened.

(2) **[Edit]** menu

<b>File</b>	<b>Edit</b>	<b>View</b>	<b>Option</b>	<b>Run</b>	<b>Event</b>	<b>Browse</b>	<b>Jump</b>	<b>Window</b>	<b>Help</b>
	Cut		Ctrl+X						
	Copy		Ctrl+C						
	Paste		Ctrl+V						
	Write in		Enter						
	Restore		Esc						
	Memory								
	Edit Source								

Fill...
Copy...
Compare...
Modify...

<a href="#">Cut</a>	Cuts a selected character string and saves it to the clipboard buffer.
<a href="#">Copy</a>	Copies a selected character string and saves it to the clipboard buffer.
<a href="#">Paste</a>	Pastes the contents of the clipboard buffer to the text cursor position.
<a href="#">Write in</a>	Writes the modified contents to the target.
<a href="#">Restore</a>	Cancels the modification.
<a href="#">Memory</a>	Manipulates the memory contents.
<a href="#">Fill...</a>	Initializes the memory. Opens Memory Fill dialog box.
<a href="#">Copy...</a>	Copies the memory. Opens Memory Copy dialog box.
<a href="#">Compare...</a>	Compares the memory. Opens Memory Compare dialog box.
<a href="#">Modify...</a>	Rewrites the memory contents during user program execution. Opens DMM dialog box.
<a href="#">Edit Source</a>	Opens the source file displayed in the active Source window with the editor specified by the PM plus when the PM plus runs.

**(3) [View] menu**

This menu displays different items depending on the window that is active. The following items may be displayed:

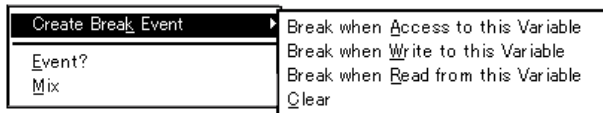
- Common items
- Source window-dedicated items
- Assemble window-dedicated items
- Memory window-dedicated items
- Watch window-dedicated items
- Register window-dedicated items
- IOR window-dedicated items
- Local Variable window-dedicated items, Stack Trace window-dedicated items
- Event Manager-dedicated items

**Common items**

File	Edit	View	Option	Run	Event	Browse	Jump	Window	Help
		Search...			Ctrl+G				
		Moye...			Ctrl+J				
		Quick Watch...			Ctrl+W				
		Add Watch...							
		View Watch							
		Change Watch...							
		Delete Watch			Del				
		Symbol...							

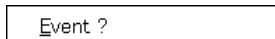
Search...	Performs a search. Opens the search dialog box corresponding to the current window. Same operation as the <Search> button.
Moye...	Moves the display position. Opens the specification dialog box corresponding to the current window.
Quick Watch...	Temporarily displays the contents of the specified data. Opens Quick Watch dialog box.
Add Watch...	Registers the specified data to the Watch window. Opens Add Watch dialog box.
View Watch	Adds the selected data to the Watch window. If the data is a symbol, it is added in accordance with the setting of Debugger Option dialog box.
Change Watch...	Changes the data on the line selected by the Watch window. Opens Change Watch dialog box. This menu is valid only when a variable is selected in Watch window.
Delete Watch	Deletes the selected watch point from the Watch window. This menu is valid only when a variable is selected in Watch window.
Symbol...	Displays the address of the specified variable or function, or the value of the specified symbol. Opens Symbol To Address dialog box.

## Source window-dedicated items



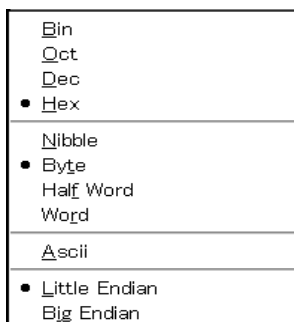
Create Break Event	Creates a break event by using the selected variable.
Break when Access to this Variable	Creates a break event that can be accessed for read/write.
Break when Write to this Variable	Creates a break event that can be accessed for write.
Break when Read from this Variable	Creates a break event that can be accessed for read.
Clear	Deletes a break event corresponding to the selected variable.
Event?	Displays the event information of a line at the cursor position or a selected variable name. If an event is set, Event dialog box is opened.
Mix	Specifies whether the results of assembly are displayed in mixed display mode, or not displayed. Checked: Mixed display Not checked: No display (default)

## Assemble window-dedicated items



Event?	Displays the event information of the address at the cursor position. If an event is set, Event dialog box is opened.
--------	--

## Memory window-dedicated items

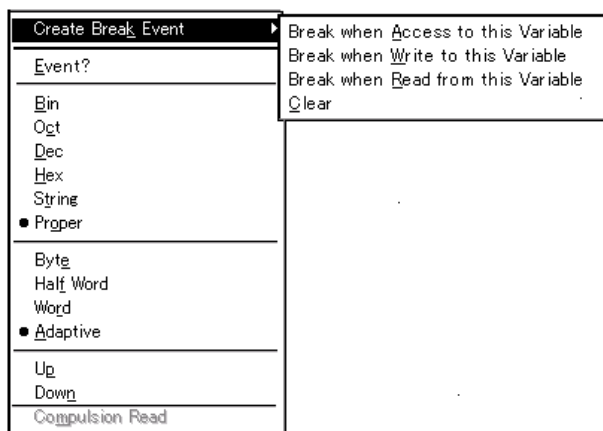


Bin	Displays binary numbers.
Oct	Displays octal numbers.
Dec	Displays decimal numbers.
Hex	Displays hexadecimal numbers (default).
Nibble	Displays in 4-bit units.

<a href="#">Byte</a>	Displays in 8-bit units (default).
<a href="#">Half Word</a>	Displays in 16-bit units.
<a href="#">Word</a>	Displays in 32-bit units.
<a href="#">Ascii</a>	Selects whether ASCII characters are displayed or not. Checked: Displayed Not checked: No display (default)
<a href="#">Little Endian</a>	Displays in little endian (default).
<a href="#">Big Endian</a>	Displays in big endian.

#### Watch window-dedicated items

Only "selected items" can be manipulated in the Watch window.



<a href="#">Create Break Event</a>	Creates a break event by using the selected item.
<a href="#">Break when Access to this Variable</a>	Creates a break event that can be accessed for read/write.
<a href="#">Break when Write to this Variable</a>	Creates a break event that can be accessed for write.
<a href="#">Break when Read from this Variable</a>	Creates a break event that can be accessed for read.
<a href="#">Clear</a>	Deletes a break event corresponding to the selected item.
<a href="#">Event?</a>	Displays the event information of the variable selected. If an event is set, Event dialog box is opened.
<a href="#">Bin</a>	Displays binary numbers.
<a href="#">Oct</a>	Displays octal numbers.
<a href="#">Dec</a>	Displays decimal numbers.
<a href="#">Hex</a>	Displays octal numbers.
<a href="#">String</a>	Displays character strings.
<a href="#">Proper</a>	Displays the default value of each variable. Symbols are displayed in accordance with the setting of Debugger Option dialog box (default).
<a href="#">Byte</a>	Displays in 8-bit units.

<a href="#">Half Word</a>	Displays in 16-bit units.
<a href="#">Word</a>	Displays in 32-bit units.
<a href="#">Adaptive</a>	Displays the default value of each variable. Only this item is valid for a symbol in C language. Symbols are displayed in accordance with the setting of Debugger Option dialog box (default).
<a href="#">Up</a>	Moves up one line.
<a href="#">Down</a>	Moves down one line.
<a href="#">Compulsion Read</a>	Forcibly reads the peripheral I/O registers that are disabled from being read because their values will be changed, or the data of the I/O ports and I/O protect area added in Add I/O Port dialog box.

#### Register window-dedicated items

<a href="#">Bin</a> <a href="#">Oct</a> <a href="#">Dec</a> <input checked="" type="radio"/> <a href="#">Hex</a>
<a href="#">Pick Up</a> <a href="#">Select...</a>

<a href="#">Bin</a>	Displays binary numbers.
<a href="#">Oct</a>	Displays octal numbers.
<a href="#">Dec</a>	Displays decimal numbers.
<a href="#">Hex</a>	Displays hexadecimal numbers (default).
<a href="#">Pick Up</a>	Displays only the registers selected in Register Select dialog box. Checked: Displays only the selected register. Not checked: Displays all the registers (default).
<a href="#">Select...</a>	Opens Register Select dialog box.

#### IOR window-dedicated items

<a href="#">Bin</a> <a href="#">Oct</a> <a href="#">Dec</a> <input checked="" type="radio"/> <a href="#">Hex</a>
Sort By <a href="#">Name</a> <input checked="" type="radio"/> Sort By <a href="#">Address</a> <a href="#">UnSort</a>
<input checked="" type="checkbox"/> <a href="#">Attribute</a>
<a href="#">Pick Up</a> <a href="#">Select...</a>
<a href="#">Compulsion Read</a>

<a href="#">Bin</a>	Displays binary numbers.
<a href="#">Oct</a>	Displays octal numbers.
<a href="#">Dec</a>	Displays decimal numbers.
<a href="#">Hex</a>	Displays hexadecimal numbers (default).

<a href="#">Sort By Name</a>	Displays in alphabetical order.
<a href="#">Sort By Address</a>	Displays in address order (default).
<a href="#">Unsort</a>	Does not sort.
<a href="#">Attribute</a>	Displays or does not display an attribute field (Attribute). Checked: Displayed (default) Not checked: Not displayed
<a href="#">Pick Up</a>	Displays only the registers selected in IOR Select dialog box. Checked: Displays only the selected register. Not checked: Displays all the registers (default).
<a href="#">Select...</a>	Opens IOR Select dialog box.
<a href="#">Compulsion Read</a>	Forcibly reads the peripheral I/O registers that are disabled from being read because their values will be changed, or the data of the I/O ports and I/O protect area added in Add I/O Port dialog box.

#### Local Variable window-dedicated items, Stack Trace window-dedicated items

<a href="#">Bin</a>
<a href="#">Oct</a>
<a href="#">Dec</a>
<a href="#">Hex</a>
<a href="#">String</a>
<a href="#">Proper</a>

<a href="#">Bin</a>	Displays binary numbers.
<a href="#">Oct</a>	Displays octal numbers.
<a href="#">Dec</a>	Displays decimal numbers.
<a href="#">Hex</a>	Displays octal numbers.
<a href="#">String</a>	Displays character strings.
<a href="#">Proper</a>	Displays the default value of each variable (default).

#### Event Manager-dedicated items

<a href="#">Select All Event</a>	Ctrl+A
<a href="#">Delete Event</a>	Del
<a href="#">Sort By Name</a>	
<a href="#">Sort By Kind</a>	
<a href="#">Unsort</a>	
<a href="#">Detail</a>	
<a href="#">Overview</a>	

<a href="#">Select All Event</a>	Selects all the events registered to the Event Manager.
<a href="#">Delete Event</a>	Deletes a selected event.
<a href="#">Sort By Name</a>	Displays icons in the order of event names.
<a href="#">Sort By Kind</a>	Displays icons in the order of event types.
<a href="#">Unsort</a>	Does not sort icons (default).
<a href="#">Detail</a>	Detailed display

<a href="#">Overview</a>	List display (default)
--------------------------	------------------------

**(4) [Option] menu**

<a href="#">File</a>	<a href="#">Edit</a>	<a href="#">View</a>	<b><a href="#">Option</a></b>	<a href="#">Run</a>	<a href="#">Event</a>	<a href="#">Browse</a>	<a href="#">Jump</a>	<a href="#">Window</a>	<a href="#">Help</a>
			<ul style="list-style-type: none"> <li>✓ <a href="#">Tool Bar</a></li> <li>✓ <a href="#">Status Bar</a></li> <li>✓ <a href="#">Button</a></li> </ul>						
			<ul style="list-style-type: none"> <li><a href="#">Source Mode</a></li> <li><a href="#">Instruction Mode</a></li> <li>• <a href="#">Auto Mode</a></li> </ul>						
			<ul style="list-style-type: none"> <li><a href="#">Configuration...</a></li> <li><a href="#">Extended Option...</a></li> <li><a href="#">Debugger Option...</a></li> <li><a href="#">Add I/O Port...</a></li> </ul>						
			<ul style="list-style-type: none"> <li><a href="#">Trace</a></li> </ul>						
			<a href="#">Tracer ON</a>						

<a href="#">Tool Bar</a>	Selects whether the tool bar is displayed (default) or not.
<a href="#">Status Bar</a>	Selects whether the status bar is displayed (default) or not.
<a href="#">Button</a>	Selects whether the buttons on each window are displayed (default) or not.
<a href="#">Source Mode</a>	Executes step execution at the source level (in line units).
<a href="#">Instruction Mode</a>	Executes step execution at the instruction level (in instruction units).
<a href="#">Auto Mode</a>	Automatically selects step execution at the source level or step execution at the instruction level (default). Step execution is performed at the source level (in a mode other than mixed display mode) if Source window is active. It is performed at the instruction level if Assemble window is active. If neither window is active, step execution is performed at the source level.
<a href="#">Configuration...</a>	Sets the environment. Opens Configuration dialog box.
<a href="#">Extended Option...</a>	Sets extended functions. Opens Extended Option dialog box.
<a href="#">Debugger Option...</a>	Sets debugger options. Opens Debugger Option dialog box.
<a href="#">Add I/O Port...</a>	Adds user-defined I/O ports. Opens Add I/O Port dialog box.
<a href="#">Trace</a>	Cannot be selected.
<a href="#">Tracer ON</a>	Cannot be selected.



## (5) [Run] menu

File	Edit	View	Option	<b>Run</b>	Event	Browse	Jump	Window	Help
					Restart		F4		
					Stop		F2		
					Go		F5		
					Ignore break points and Go		Ctrl+F5		
					Return Out		F7		
					Step In		F8		
					Next Over		F10		
					Start From Here		Shift+F6		
					Come Here		F6		
					Go & Go				
					Slowmotion				
					CPU Reset		F3		
					Change PC		Ctrl+F9		
					Break Point		F9		
					Software Break Point		F11		
					Delete All Breakpoints				
					• Uncond. Trace ON				
					Cond. Trace ON				
					Trace Control Mode				
					Tracer Start				

<a href="#">Restart</a>	Resets the CPU and executes the program. Same operation as the <b>ReGo</b> button
<a href="#">Stop</a>	Forcibly stops program execution. Same operation as the <b>Stop</b> button
<a href="#">Go</a>	Executes the program from the current PC. Same operation as the <b>Go</b> button
<a href="#">Ignore break points and Go</a>	Executes the program, ignoring break points being set. Same operation as the <b>GoN</b> button
<a href="#">Return Out</a>	The user program is real-time executed until execution returns. Same operation as the <b>Ret</b> button <b>Note:</b> This menu item is available only for a function described in C language.
<a href="#">Step In</a>	Executes the instructions in the program one by one (step execution). If a function or subroutine is called, its instructions are executed one by one. Same operation as the <b>Step</b> button
<a href="#">Next Over</a>	Executes the instructions in the program one by one (Next step execution). If a function or subroutine is called, its instructions are not executed on a step-by-step basis. Same operation as the <b>Over</b> button
<a href="#">Start From Here</a>	Executes the program from the cursor position on Source window or Assemble window.
<a href="#">Come Here</a>	Executes the program from the current PC to the cursor position in the Source window or Assemble window.
<a href="#">Go &amp; Go</a>	Continues executing the program. If a break occurs because a break condition is satisfied, the window is updated and the program is executed again. Each time a break has occurred, the same operation as clicking the <b>Go</b> button is performed.

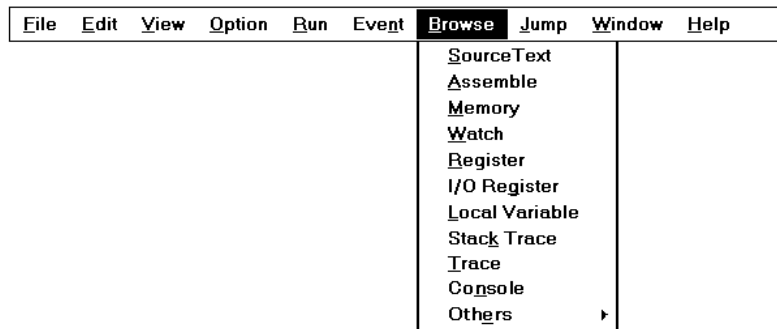
<a href="#">Slowmotion</a>	Continues step execution. Each time step execution has been performed, the window is updated and then step execution is performed again. Same operation as clicking the <b>Step</b> each time a break has occurred.
<a href="#">CPU Reset</a>	Resets the CPU. Same operation as the <b>Res</b> button
<a href="#">Change PC</a>	Sets the address at the cursor position in the Source window or Assemble window to the PC.
<a href="#">Break Point</a>	Sets or cancels a breakpoint at the cursor position in the Source window or Assemble window.
<a href="#">Software BreakPoint</a>	Sets or cancels a software breakpoint at the cursor position in the Source window or Assemble window.
<a href="#">Delete All Breakpoints</a>	Deletes all software breakpoints that are set.
<a href="#">Uncond. Trace ON</a>	Cannot be selected.
<a href="#">Cond. Trace ON</a>	Cannot be selected.
<a href="#">Tracer Control Mode</a>	Cannot be selected.
<a href="#">Tracer Start/Tracer Stop</a>	Cannot be selected.

## (6) [Event] menu

File	Edit	View	Option	Run	Event	Browse	Jump	Window	Help
					Event Manager				
					Software Break Manager				
					Event...				
					Event Link...				
					Break...				
					Trace...				
					Delay Count...				

<a href="#">Event Manager</a>	Manages various event conditions. Opens Event Manager. Same operation as the <b>Mgr</b> button
<a href="#">Software Break Manager</a>	Manages software break event conditions. Opens the Software Break Manager.
<a href="#">Event...</a>	Registers an event condition. Opens Event dialog box. Same operation as the <b>Ev</b> button
<a href="#">Event Link...</a>	Registers an event link condition. Opens Event Link dialog box.
<a href="#">Break...</a>	Registers and sets a break condition. Opens Break dialog box. Same operation as the <b>Brk</b> button
<a href="#">Trace...</a>	Cannot be selected.
<a href="#">Delay Count...</a>	Cannot be selected.

## (7) [Browse] menu



<a href="#">Source Text</a>	Displays a source text. Opens Source window. Same operation as the <b>Src</b> button If there is a Source Text window already open in the active status, it is opened in the static status.
<a href="#">Assemble</a>	Displays the disassemble results. Opens Assemble window. Same operation as the <b>Asm</b> button If there is a Assemble window already open in the active status, it is opened in the static status.
<a href="#">Memory</a>	Displays the contents of the memory. Opens Memory window. Same operation as the <b>Mem</b> button If there is a Memory window already open in the active status, it is opened in the static status.
<a href="#">Watch</a>	Displays the watch contents. Opens Watch window. Same operation as the <b>Wch</b> button
<a href="#">Register</a>	Displays the register contents. Opens Register window. Same operation as the <b>Reg</b> button
<a href="#">I/O Register</a>	Displays the contents of the Peripheral I/O registers. Opens IOR window. Same operation as the <b>IOR</b> button
<a href="#">Local Variable</a>	Displays the local variable. Opens Local Variable window. Same operation as the <b>Loc</b> button
<a href="#">Stack Trace</a>	Displays the stack trace results. Opens Stack Trace window. Same operation as the <b>Stk</b> button
<a href="#">Trace</a>	Cannot be selected.
<a href="#">Console</a>	Opens Console window.
<a href="#">Others</a>	Displays other windows. Displays a user-defined window list.

## (8) [J]ump] menu

File	Edit	View	Option	Run	Event	Browse	<b>Jump</b>	Window	Help
							SourceText	Ctrl+U	
							Assemble	Ctrl+D	
							Memory	Ctrl+M	

<a href="#">Source Text</a>	Displays the corresponding source text and source line, using the data value selected in the current window as the jump destination address. If no line information is at the jump destination address, however, you cannot jump. Opens Source window. If an active Source window is open, that window is displayed in the forefront (so that it can be manipulated).
<a href="#">Assemble</a>	Disassembles and displays the results from the jump destination address specified by the data value selected in the current window. Opens Assemble window. If an active Assemble window is open, that window is displayed in the forefront (so that it can be manipulated).
<a href="#">Memory</a>	Displays the memory contents from the jump destination address specified by the data value selected in the current window. Opens Memory window. If an active Memory window is open, that window is displayed in the forefront (so that it can be manipulated).

## (9) [W]indow] menu

File	Edit	View	Option	Run	Event	Browse	Jump	<b>Window</b>	Help
								New Window	
								Cascade	
								Tile	
								Arrange Icons	
								Close All	
								Refresh	
								• Active	
								Static	
								1 Trace View	
								✓ 2 Assemble	

<a href="#">New Window</a>	Opens a new window displaying the same contents as those of the current window. This menu is valid only when the current window is Source window, Assemble window, or Memory window.
<a href="#">Cascade</a>	Cascade display of the windows in the Main window.
<a href="#">Tile</a>	Tile display of the windows in the Main window.
<a href="#">Arrange Icons</a>	Rearranges the icons in the Main window.
<a href="#">Close All</a>	Closes all windows, except the Main window.
<a href="#">Refresh</a>	Updates the contents of the window with the latest data.
<a href="#">Active</a>	Sets the window in the active status. If the window is already in the active status, a check mark is shown on the left of this item.

<a href="#">Static</a>	Sets the window in the static status. If the window is already in the static status, a check mark is shown on the left of this item.
<a href="#">(Open Window)</a>	Lists the windows that are open. The window with the check mark shown on the side of the figure is the current window. By selecting a window name, the selected window is used as the current window.

#### (10) [Help] menu

File	Edit	View	Option	Run	Event	Browse	Jump	Window	<b>Help</b>
ID850NWC Help Command Reference									
Main Window Help Current Window Help F1									
About...									

<a href="#">ID850NWC Help</a>	Displays the help of ID850NWC.
<a href="#">Command Reference</a>	Opens the Help window of COMMAND REFERENCE.
<a href="#">Main Window Help</a>	Displays the help of the Main window.
<a href="#">Current Window Help</a>	Displays the help of the current window.
<a href="#">About...</a>	Displays the version of the ID850NWC. Opens About dialog box.

#### Window display area

This area (at the center) displays various debug windows.

The displayed window can be changed in size or an icon can be created in this area.

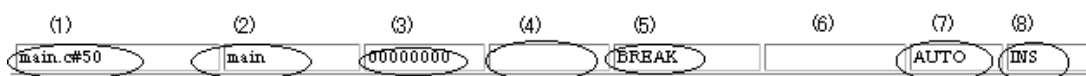
#### Status bar

The status bar is displayed at the bottom of the window and indicates the status of the ID850NWC and in-circuit emulator.

While the user program is being executed, the status bar is displayed in red.

Whether the status bar is displayed or not can be selected by selecting [Option] -> [Status Bar] from the menu bar.

Figure 7-4 Status Bar



(1) Program name	Displays the program file name indicated by the PC value.
Source name	Displays the source file name indicated by the PC value.
Line number	Displays the line number indicated by the PC value.
(2) Function name	Displays the function name indicated by the PC value.
(3) PC value	Displays the current PC value.
(4) CPU status	Displays the status of the CPU (target device). <b>RESET</b> ...Reset status <b>POW OFF</b> ...Power is not supplied to the target. <b>HOLD</b> ...Bus hold <b>HALT</b> ...Halt mode <b>WAIT</b> ...Wait mode <b>STOP/IDLE</b> ...Software stop mode, hardware stop mode, or idle
(5) IE status	Displays the status of the in-circuit emulator. (If there are two or more the statuses, they delimited with ' ' and displayed.) <b>RUN</b> ...User program execution in progress (the color of the status bar changes). <b>STEP</b> ...Step execution in progress <b>BREAK</b> ...Break occurring. <b>Time</b> ...Displays the result of measuring the time from the start of user program execution to the occurrence of break (Run-Break time) (measurable range: 100 ns to 214 seconds). <b>TIMER OVERFLOW</b> ...Measurement result overflowed. <b>Caution:</b> <b>Time</b> and <b>TIMER OVERFLOW</b> are displayed only with a CPU core that includes RCU1.
(6) Break Cause	Displays Table 7-2 Break Cause.
(7) STEP mode	Displays the step execution mode. Displays that the following modes are selected from the [Option] menu: <b>SRC</b> ...Source mode <b>INST</b> ...Instruction mode <b>AUTO</b> ...Automatic mode
(8) Key input mode	Displays the key input mode. <b>INS</b> ... Insertion mode <b>OVR</b> ...Overwrite mode The Memory window is fixed to OVR mode.

Table 7-2 Break Cause

Display	Meaning
Manual Break	Forced break
Temporary Break	Temporary break
Event Break	Break by event
Software Break	Software break
Flash Macro Service	Under flash macro service

Table 7-2 Break Cause

Display	Meaning
Ffa01: The line information on PC position not found.	<p>The source file corresponding to program counter (PC) value when the program was stopped could not be found. The following reasons are possible.</p> <ol style="list-style-type: none"><li>1) The source file exists in a location that the source path does not connect to.</li><li>2) The program stopped where the source files, such as library or RX, do not exist.</li><li>3) The program looped, jumped to an address that is not used by the program, and stopped there.</li></ol>

## Configuration dialog box

Displays and sets the ID850NWC operation environment.

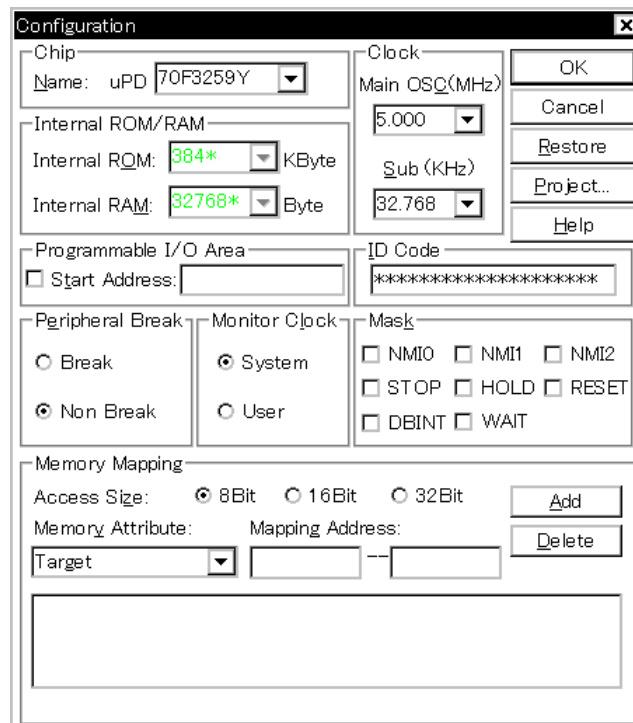
This dialog box is automatically displayed after the ID850NWC is started up.

To use the ID850NWC, the operating environment of the ID850NWC and in-circuit emulator must be first set in this dialog box.

However, no setting is required to read a project as the results of reading the project file are reflected in this dialog box.

For details of the mapping function, refer to "Mapping Function".

Figure 7-5 Configuration Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
  - Setting of memory mapping
- Function buttons
- Related operations
  - To change the value of a register required for access of an external memory



## Opening

This dialog box can be opened as follows:

- Automatically when the ID850NWC is started up
- Select [Option] -> [Configuration...] from the menu bar.  
(Or press Alt, O, and C in that order)

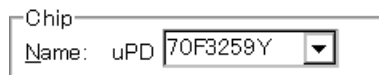
## Explanation of each area

The Configuration dialog box consists of the following areas:

- (1) Chip (CPU selection area)
- (2) Internal ROM/RAM (internal ROM/RAM setting area)
- (3) Programmable I/O Area (programmable I/O area address setting area)
- (4) Clock (CPU clock source selection area)
- (5) ID Code (ID code input area)
- (6) Peripheral Break (break mode selection area)
- (7) Monitor Clock (monitor operation clock selection area)
- (8) Mask (mask setting area)
- (9) Memory Mapping (mapping setting area)

### (1) Chip (CPU selection area)

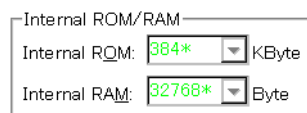
This area is used to select the chip name.



A chip name is selected from the drop-down list. On the drop-down list, only the chip names registered to the registry from the DF installer are displayed.

This area can be specified only when the debugger is started up. The chip name selected at the previous start-up is displayed by default.

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



This area is used to display the size of the internal RAM and internal ROM of the CPU.

The internal ROM and RAM sizes cannot be changed.

The default size is automatically displayed with the asterisk (\*) attached at the end when a chip is selected.

**(3) Programmable I/O Area (programmable I/O area address setting area)**

This area is used to specify usage of the programmable I/O area and the start address.

The start address of the programmable I/O area can be input by checking the check box when the programmable I/O area is used, only if the device selected by Chip supports the programmable I/O area. The address is aligned to 16 KB.

If the device selected by **Chip** has an expansion I/O area of fixed addresses, this area is automatically set.

**(4) Clock (CPU clock source selection area)**

This area is used to select the frequencies of the CPU operation main clock (fX) and subclock (fTx).

The frequencies set in this area are used for WAIT calculation for JTAG communication.

<b>Main OSC</b>	Specifies a frequency before the main clock is multiplied. A frequency can be selected from the drop-down list (5.000, 8.000, 13.500, or 18.000) or input from the keyboard.
<b>Sub</b>	Specifies the frequency of the subclock (this cannot be set if the device selected by <b>Chip</b> does not have a subclock). A frequency can be selected from the drop-down list or input from the keyboard.

**(5) ID Code (ID code input area)**

This area is used to input the ID code to be used when the code on the internal ROM or internal flash memory is read (ID code authentication).

Input a hexadecimal number of 20 digits (10 bytes) as the ID code (all 'F' by default). The ID code is saved to the registry.

If inputting the ID code fails three times, the ID850NWC is forcibly terminated.

This area does not have to be set with a ROMless product or a product without a security unit.

For the details of ID code authentication, refer to "IE-V850E1-CD-NWUser's Manual".

**(6) Peripheral Break (break mode selection area)**

Peripheral Break

☐ Break

☒ Non Break

This area is used to specify whether the peripheral I/O function is stopped during a break.

Break	Stopped
Non Break	Not stopped (default)

**(7) Monitor Clock (monitor operation clock selection area)**

Monitor Clock

☒ System

☐ User

This area is used to specify whether the operation clock of the monitor program is switched from the subclock to the main clock during a break.

This area does not have to be set with a product without a subclock.

System	The operation clock is switched to the main clock and the monitor program is executed (default). <b>Caution :</b> In the ID850NWC, the clock is changed by manipulating PCC, but not while the main clock is stopped. If the operation clock is switched to the main clock during a break, the clock is returned to the previous setting when execution returns to the user program.
User	The monitor program is executed with the clock selected by the user program.

**(8) Mask (mask setting area)**

Mask

☐ NMIO ☐ NMI1 ☐ NMI2

☐ STOP ☐ HOLD ☐ RESET

☐ DBINT ☐ WAIT

This area is used to mask the signal sent from the target .

The signal of a masked pin is not input to the in-circuit emulator and CPU.

Mask a pin only when the operation of the target is not stable at the debugging stage.

By checking RESET, the external reset or internal reset generated by the watchdog timer can be masked. At this time, whether internal reset can be masked or not depends on the device.

**(9) Memory Mapping (mapping setting area)**

This area is used to set mapping by specifying the access size, memory attribute, and address.

**Access Size (selection of memory access size)**

Selects memory access size.

This setting is used to specify the access size on the ID850NWC software; the operation of the external bus hardware is set in accordance with the settings of the MODE pin and I/O register.

8Bit	Accesses memory with ld.b instruction/st.b instruction (default).
16Bit	Accesses memory with ld.h instruction/st.h instruction.
32Bit	Accesses memory with ld.w instruction/st.w instruction.

**Memory Attribute (mapping attribute specification)**

The following mapping attributes can be selected. Select a mapping attribute according to the usage(refer to "Mapping Function").

Target	Selects the target memory.
I/O Protect	Selects the I/O protect area. This area can be set only in the area set as Target.

**Caution** The area set as **I/O Protect** is not read unless it is registered to the IOR window a or Watch window as an I/O port. To read this area, forcibly read it on these windows.

**Mapping Address (mapping address specification)**

Specify the address to be mapped.

Input the higher and lower addresses from the keyboard.

**Setting of memory mapping**

Specify **Memory Attribute** and **Mapping Address** and then click the <Add> button.

-The attribute of >**Memory Attribute**, address range of **Memory Mapping**, and memory mapping corresponding to the data of the bus width are set and displayed on a list.

A mapping attribute other than **I/O Protect**, a mapping attribute is adjusted in mapping units when the <Add> button is clicked. If the mapping unit does not match, the minimum settable range including the specified address is mapped.

The mapping unit is as follows:

Table 7-3 Mapping Unit in Configuration Dialog Box

Mapping Area	Mapping Unit
I/O Protect	1-byte unit
Target	1Mbyte units

**Caution** If the external memory is mapped, change the value of a register required for access of an external memory.(refer to "To change the value of a register required for access of an external memory").

#### Deleting memory mapping

Select mapping to be deleted from the list and click the <Delete> button.

-> This deletes the selected mapping.

#### Function buttons

OK	Validates the current environment. Sets the environment and closes this dialog box.
Cancel	Cancels the changes and closes this dialog box.
Restore	Restores the previous environment settings status before this dialog box was opened.
Project	Opens the Project File Load dialog box. If an error occurs while a project file is being opened or read, the ID850NWC can no longer continue and is terminated.
Help	Opens the Help window.

#### Related operations

##### To change the value of a register required for access of an external memory

When mapping has been performed for external memory, change the values of the registers required for accessing external memory prior to downloading, using IOR window or the hook procedure.

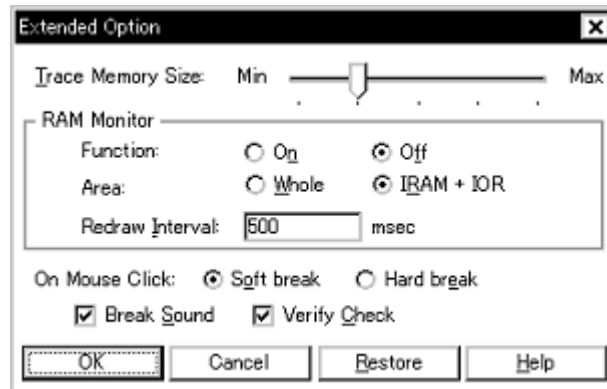
For how to change register values using the hook procedure, refer to "Hook Procedure" in COMMAND REFERENCE.

For the registers to be changed, refer to the hardware manual of the CPU that is used.

## Extended Option dialog box

This dialog box is used to display and set the extended options of the ID850NWC.

Figure 7-6 Extended Option Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

- Select [Option] -> [Extended Option...] from the menu bar.  
(Or press Alt, O, and X in that order)

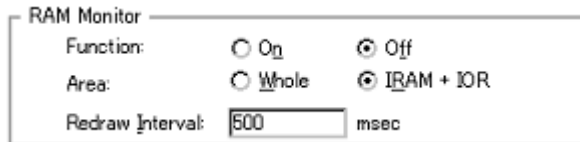
### Explanation of each area

The Extended Option dialog box consists of the following areas:

- (1) Trace Memory Size (trace memory size setting area)
- (2) RAM Monitor (RAM monitor setting area)
- (3) On Mouse Click (default break selection area)
- (4) Break Sound (beep sound specification area)
- (5) Verify Check (verify specification area)

**(1) Trace Memory Size (trace memory size setting area)**

This area does not have to be set.

**(2) RAM Monitor (RAM monitor setting area)****Function (Select RAM Sampling function)**

This area selects the RAM sampling function as valid/invalid.

Since this function is realized by software emulation, the user program execution is temporarily stopped.

<b>On</b>	Enables this function. All the break events already set become invalid.
<b>Off</b>	Disables this function (default).

**Area (Whole RAM sampling range specification)**

Specify the address range in RAM sampling is to be performed.

The variables and data allocated to this area can be displayed in Watch window , Memory window and IOR window.

<b>Whole</b>	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 wid.
<b>IRAM+IOR</b>	Only the internal RAM area and the peripheral I/O register area are the target (default).

**Redraw Interval ( RAM sampling time specification)**

Specify the sampling time (ms) of the RAM sampling.

The RAM area can be sampled during emulation.

Variables, data, etc., allocated to the RAM area can be displayed in the Watch window, Memory window, or IOR window.

The sampling time can be specified in 100-ms units from 0 to 65500. Display is not performed if 0 or nothing is specified.

**(3) On Mouse Click (default break selection area)**

On Mouse Click: ☒ Soft break ☐ Hard break

This area is used to select whether a software breakpoint or hardware breakpoint is set as the default breakpoint if a breakpoint is set in the point mark area by clicking the mouse button in the Source window or Assemble window.

Soft break	Sets a software breakpoint (default). The mark of breakpoint is displayed in blue.
Hard break	Sets a hardware breakpoint. The mark of breakpoint is displayed in red.

**(4) Break Sound (beep sound specification area)**

☒ Break Sound

This area is used to specify whether a beep sound is issued upon the occurrence of a break.

If the check box is checked, a beep sound is issued when a break occurs (default).

**(5) Verify Check (verify specification area)**

☒ Verify Check

This area is used to specify whether a verify check is performed when data has been written to memory.

If the check box is checked, a verify check is performed when download, memory fill, or memory copy is executed (default). A verify check is also performed when a variable or data is changed in Watch window or Memory window and is written to memory.

When they are written to the internal flash memory(including downloading), an internal verify is performed regardless of check in this area(but read verify is not performed).

**Function buttons**

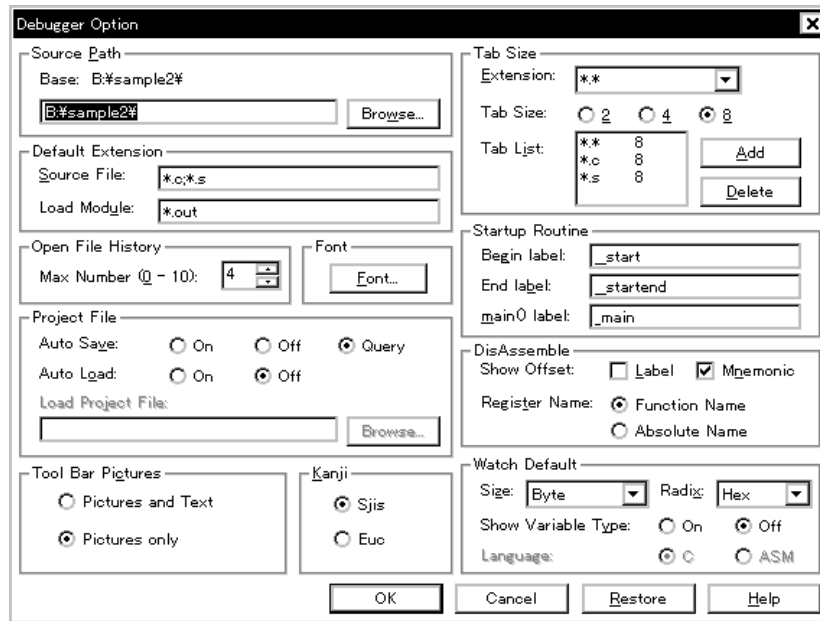
OK	Validates the settings and closes this dialog box.
Cancel	Cancels the settings and closes this dialog box.
Restore	Restores the previous settings before this dialog box was opened.
Help	Opens the Help window.



## Debugger Option dialog box

This dialog box is used to display and set the various options of the ID850NWC.

Figure 7-7 Debugger Option Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

- Select [Option] -> [Debugger Option...] from the menu bar (or press Alt+O, and G in that order).

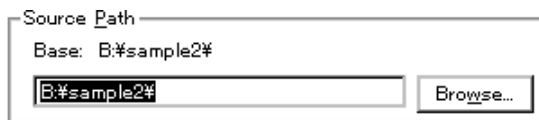
### Explanation of each area

The Debugger Option dialog box consists of the following areas:

- (1) Source Path (source path specification area)
- (2) Default Extension (default extension specification area)
- (3) Open File History (open file history setting area)

- (4) Font (display font setting area)
- (5) Project File (project file setting area)
- (6) Tool Bar Pictures (toolbar display button setting area)
- (7) Kanji (Kanji code setting area)
- (8) Tab Size (tab size setting area)
- (9) Startup Routine (start-up symbol setting area)
- (10) DisAssemble (offset display setting area)
- (11) Watch Default (watch symbol setting area)

(1) **Source Path** (source path specification area)



This area is used to specify the directory in which a source file or text file is searched.

A path can be input from the keyboard or set by the <Browse...> button. A relative path can also be specified.

The directory that is the basis of a relative path is displayed on the right of **Base:**. The base directory is determined in the following sequence:

- Directory to which the project file has been loaded
- Directory to which a load module or hex file has been loaded last
- Current directory of Windows

To delimit paths, use ";" (semicolon) or "," (comma).

Example: If a source is in the following directory

C:\src

C:\asm

The source path is specified as follows:

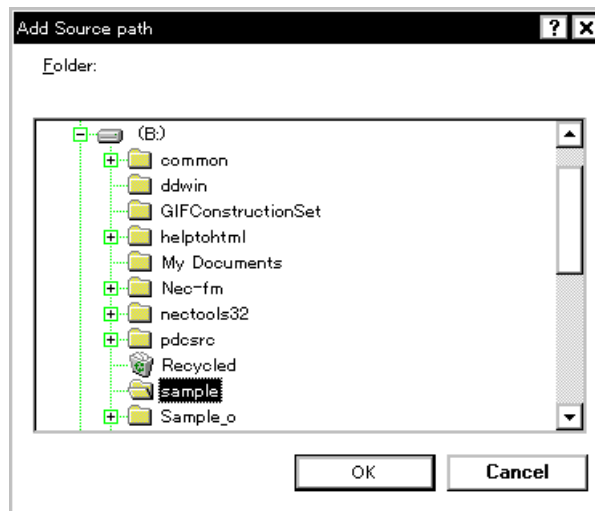
C:\src; C:\asm

Clicking the <Browse...> button displays the Add Source path Dialog Box and a source path can be added.

**Reference** Immediately after the Debugger Option dialog box has been opened, the base directory is selected and opened

If the selected directory has already been set for the source path, a source path is not added.

Figure 7-8 Add Source path Dialog Box



## (2) Default Extension (default extension specification area)

Default Extension

Source File: \*.c;\*.s

Load Module: \*.out

Delimit extensions with " " (blank), ";" (semicolon) or "," (comma).

Source File	Set the extension of a source file that is displayed when the Browse dialog box is opened by selecting [File] -> [Open...] from the menu bar. The default extension is "*.C, *.S".
Load Module	Set the extension of a load module that is displayed when Download dialog box is opened. The default extension is "*.OUT".

## (3) Open File History (open file history setting area)

Open File History

Max Number (0 - 10): 4

This area is used to set the number of histories (0 to 10) of the open file displayed in the bottom field of the [File] menu. If 0 is set, no history is displayed on the menu.

The default value is 4.

## (4) Font (display font setting area)

Font

Font...

This area is used to specify the font displayed on the Source window. Clicking the <Font...> button opens the Font dialog box in which the font to be displayed and its size can be set.

**(5) Project File (project file setting area)**

This area is used to set automatic saving and loading of the project file.

**Auto save**

Sets whether the project file is automatically saved at the ID850NWC termination.

<a href="#">On</a>	Automatically saves the project file at the ID850NWC termination.
<a href="#">Off</a>	Does not automatically save the project file at the ID850NWC termination.
<a href="#">Query</a>	Displays the Exit Debugger dialog box at the ID850NWC termination (default).

**Auto Load**

Sets whether the project file is automatically loaded at the ID850NWC start up.

<a href="#">On</a>	Automatically loads the project file at the ID850NWC start up. Sets whether the <a href="#">Load Project File</a> is automatically loaded
<a href="#">Off</a>	Does not automatically loads the project file at the ID850NWC start up(default).

**Load Project File**

Specify the project file to be loaded automatically.

This item can be specified only when [On](#) is specified for [Auto Load](#).

Set a project file name by inputting from the keyboard or clicking the <Browse...> button. Clicking the <Browse...> button displays Browse dialog box.

**(6) Tool Bar Pictures (toolbar display button setting area)**

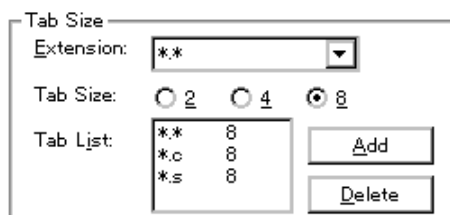
This area sets the buttons to be displayed on the toolbar.

<a href="#">Pictures and Text</a>	Displays a button on which a graphic and character are displayed.
<a href="#">Pictures only</a>	Displays a button with only graphic (default).

**(7) Kanji (Kanji code setting area)**

This area is used to specify Kanji codes to be displayed in the Source window. Kanji codes can be specified only on the Japanese Windows.

<a href="#">Sjis</a>	Uses SJIS as Kanji codes (default).
<a href="#">Euc</a>	Uses EUC as Kanji codes.

**(8) Tab Size (tab size setting area)**

This area is used to set the tab size for each extension when files are displayed.

**Extension**

Set an extension. Input an extension from the keyboard, or select one from the drop-down list.

**Tab Size**

Select the tab size. Select how many spaces are displayed as a tab code.

<a href="#">2</a>	Displays two spaces.
<a href="#">4</a>	Displays four spaces.
<a href="#">8</a>	Displays eight spaces.

**Tab List**

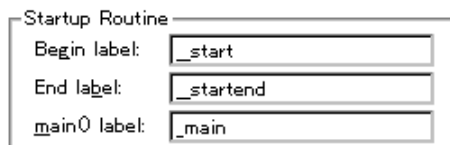
Displays the tab size set for each extension.

**To change the tab size setting,**

select [Extension](#) and [Tab Size](#) and click the <Add> button.

**To delete the tab size setting,**

select the setting to be deleted from [Tab List](#) and click <Delete> button.

**(9) Startup Routine (start-up symbol setting area)**

This area is used to specify the first address and end address of the text area (code area) of the start-up routine by symbols.

Specify the symbol of the first address as [Begin label](#), and the symbol of the last address as [End label](#).

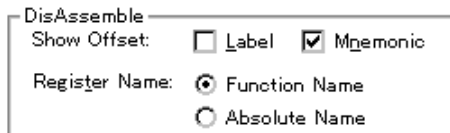
In this way, the source file can be automatically opened if an object file in the load module format is downloaded in Download dialog box. Specify the source file to be displayed at this time as `main()` label, as a display start symbol.

If the specified symbol is not correct, the source file cannot be opened until the PC reaches the address range of the corresponding source file. In addition, the start-up routine cannot be skipped by step execution.

<code>Begin label</code>	Symbol of first address (default: <code>_start</code> )
<code>End label</code>	Symbol of end address (default: <code>_startend</code> )
<code>main() label</code>	Display start symbol (default: <code>_main</code> )

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

#### (10) **DisAssemble** (offset display setting area)



Performs the setting for disassemble display.

##### **Show Offset**

Specifies whether an offset (symbol + offset) is displayed during disassemble display.

When the offset is not displayed, only a symbol that matches a numeric value is displayed, if any. If no matching symbol is found, the numeric value is displayed as a hexadecimal number unchanged.

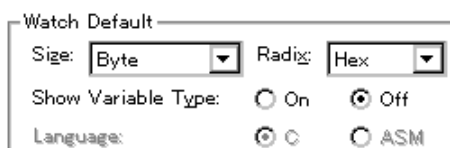
<code>Label</code>	Specifies whether the offset is displayed in the Label field. This check box is not checked in the default condition, and the offset is not displayed.
<code>Mnemonic</code>	Specifies whether the offset is displayed in the Mnemonic field. This check box is checked in the default condition, and the offset is displayed.

##### **Register Name** (register name display selection area)

This area is used to select the method of displaying register names in mnemonics during disassemble display.

<code>Function Name</code>	Displays register names as function names or nicknames (default).
<code>Absolute Name</code>	Displays register names as absolute names.

#### (11) **Watch Default** (watch symbol setting area)



This area is used to specify a symbol to be watched.

**Size (default size specification)**

Sets the display size of data if [Adaptive] is selected in Watch window etc.

Byte	8-bit display(default)
Half Word	16-bit display
Word	32-bit display

**Radix (default radix specification)**

Sets the default radix in which data is to be displayed if [Proper] is specified in Watch window etc.

Hex	Displays data in hexadecimal numbers (default).
Dec	Displays data in decimal numbers.
Oct	Displays data in octal numbers.
Bin	Displays data in binary numbers.
String	Character string

**Show Variable Type (display/non-display of variable type)**

Select the display/non-display of variable type is specified in Watch window etc.

On	Displays the type of a variable.
Off	Does not display the type of a variable (default).

**Language (display/non-display of type of variable)**

Select the display/non-display of type of variable is specified in Watch window etc.

With the ID850NWC, the display format is fixed to C-like and cannot be selected.

C	Displays a C-like base number (default).
ASM	Cannot be selected.

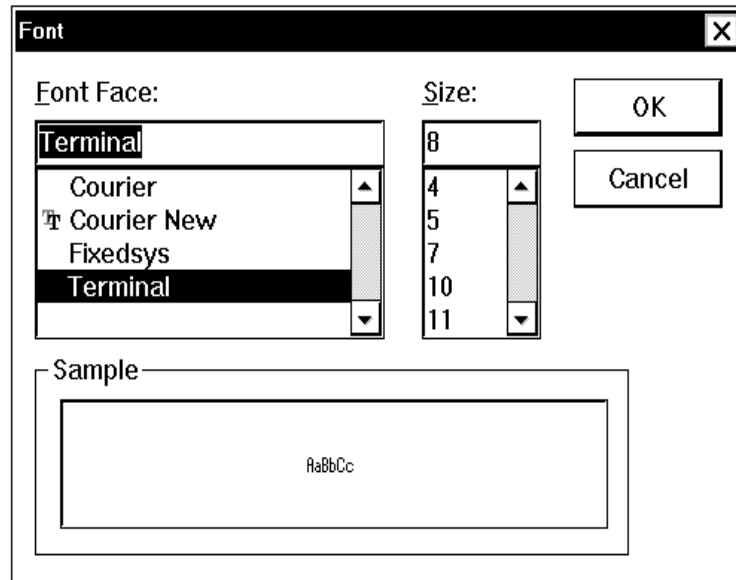
**Function buttons**

OK	Validates the settings and closes this dialog box.
Cancel	Cancels the settings and closes this dialog box.
Restore	Restores the previous settings before this dialog box was opened.
Help	Opens the Help window.

## Font dialog box

This dialog box is used to select the font to be displayed in the Source window.

Figure 7-9 Font Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

- Click the <Font> button in Debugger Option dialog box.

### Explanation of each area

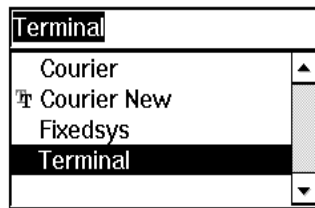
The Font dialog box consists of the following areas:

- (1) Font Face: (font name specification area)
- (2) Size: (font size specification area)
- (3) Sample (sample display area)



**(1) Font Face: (font name specification area)**

Font Face:



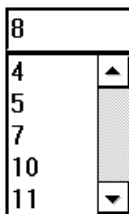
This area is used to select a font from the fonts currently usable.

Only fonts with equal width (fonts with a constant stroke width and a fixed pitch) are enumerated.

When a font name is selected from the list, the font name is displayed in the text box, and the font size that can be used with that font is displayed under [Size](#).

**(2) Size: (font size specification area)**

Size:

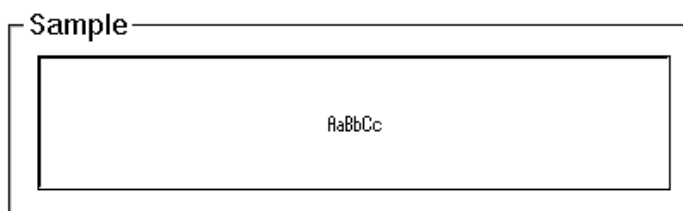


This area is used to specify the font size (unit : pt).

On the drop-down list, the font size usable for the font specified in [Font Face](#) is displayed in point units.

When the font size is selected from the drop-down list, the selected font size is displayed in this area.

The font size can also be directly input to the text box from the keyboard.

**(3) Sample (sample display area)**

This area displays a sample character string of the specified font and size.

**Function buttons**

OK	Changes the font and its size in the Source window as specified, and closes this dialog box.
Cancel	Closes the dialog box without changing the font.

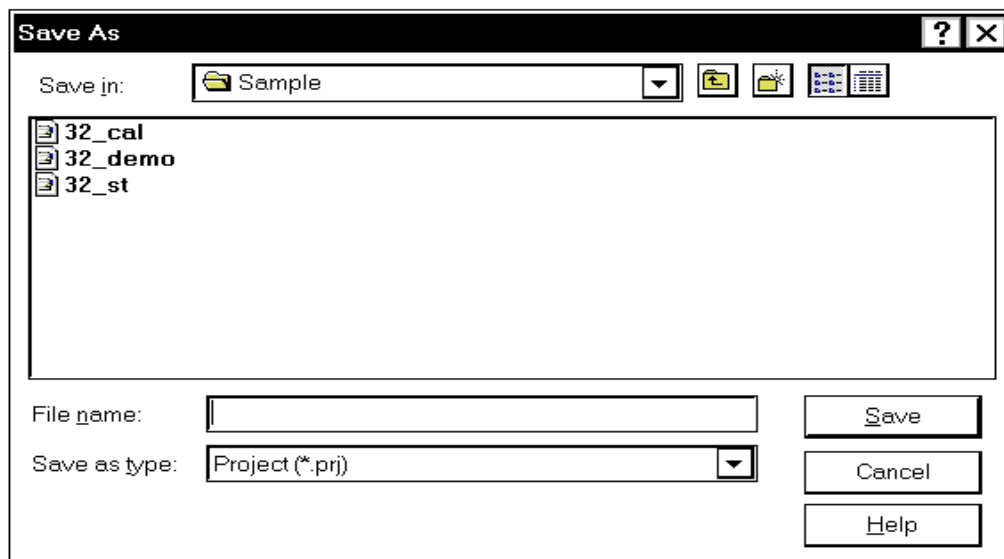
## Project File Save dialog box

Saves the debug environment.

It is used to save a new project file or an existing file with the file name changed (refer to "Setting Debugging Environment").

When a file is saved, the size and position of the window that has been displayed are also saved.

Figure 7-10 Project File Save Dialog Box



The following contents are saved to the project file:

Table 7-4 Contents Saved to Project File

Window Name	Set Data
Configuration dialog box	All items (target device, clock setting, pin mask setting, mapping information)
Main window	Display position, tool bar/status bar/button display information, execution mode information
Download dialog box	File information to be downloaded
Extended Option dialog box, Debugger Option dialog box	Set information
Source window	Display information, font information of window
Assemble window, Memory window	Display information of window, display start address

Table 7-4 Contents Saved to Project File

Window Name	Set Data
Stack Trace window, IOR window, Local Variable window, Event Manager, Console window, EXPANSION WINDOW	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
Register window	Display information of window, display bank information
Watch window	Display information of window, watch registration information
Add I/O Port dialog box	Added I/O port information
DMM dialog box	DMM information
Software Break Manager	Display information of window, software break information

This section explains the following items:

- Opening
- Explanation of each area
- Function buttons
- Related operations
  - To automatically save a project file

## Opening

This dialog box can be opened as follows:

- Select [File] -> [Project] -> [Save As...] from the menu bar  
(or press Alt+F, J and A in that order).

### To save a file of same name as a project file previously loaded or saved

- Select [File] -> [Project] -> [Save] on the menu bar  
(or press Alt+F, J and S in that order).

## Explanation of each area

The Project File Save dialog box consists of the following areas:

- (1) Save in:, File name: (file selection area)
- (2) Save as type: (extension selection area)

**(1) Save in:, File name: (file selection area)**

This area is used to specify a file name. A file name can be directly input from the keyboard, or selected from the list at the upper part of this area.

Up to 64 character string with a extension can be specified.

**(2) Save as type: (extension selection area)**

This area is used to specify the extension (**.PRJ**) of the project file to be loaded.

If the extension is omitted, **.PRJ** is appended as the default extension.

---

**Function buttons**

Save	Saves the project information to the selected file. After saving, the dialog box is closed.
Cancel	Closes this dialog box without saving the file.
Help	Opens the Help window.

---

**Related operations****To automatically save a project file**

Use Project File (project file setting area) in Debugger Option dialog box that is opened when [Option] -> [Debugger Option...] is selected from the menu bar.

Select On for Auto Save in this area. The current environment will be automatically saved to the project file when the dialog box is closed.

## Project File Load dialog box

This dialog box is used to return the debug environment to the previous environment.

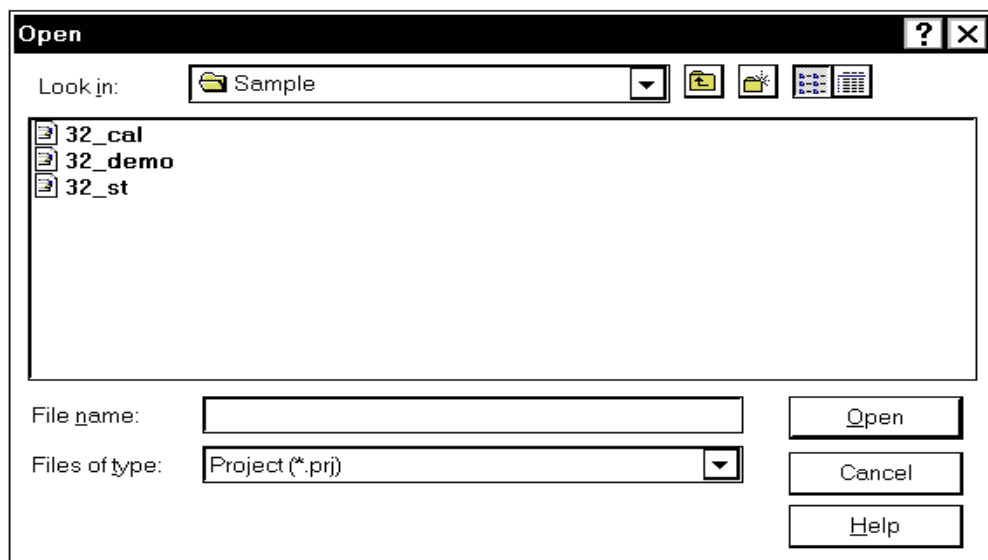
After a file has been loaded, the size and position of the window is returned to the previous status (status saved to a project file) (refer to "Setting Debugging Environment").

After a project file has been loaded, it is displayed on the top of an active Source window if any.

By loading the project file, the items to be saved in Table 7-4 Contents Saved to Project File are set.

**Caution** If a project file different from the target device after ID850NWC has been loaded is loaded, ID850NWC loads the project file after it has been reset (ID850NWC is then used as the target device specified by the project file).

Figure 7-11 Project File Load Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons
- Related operations
  - To automatically load a project file

## Opening

This dialog box can be opened as follows:

- Click the **Proj** button.
- Select [File] -> [Project] -> [Open...] on the menu bar (or press Alt+F, J and O in that order).

## Explanation of each area

The Project File Load dialog box consists of the following areas:

- (1) Look In:, File name: (file selection area)
- (2) Files of type: (extension selection area)

### (1) Look In:, File name: (file selection area)

This area is used to specify a file name. A file name can be directly input from the keyboard, or selected from the list at the upper part of this area.

Up to 64 character string with a extension can be specified.

### (2) Files of type: (extension selection area)

This area is used to specify the extension (**.PRJ**) of the project file to be loaded.

## Function buttons

Open	Loads the selected file. After loading the file, this dialog box is closed.
Cancel	Closes this dialog box without loading the file.
Help	Opens the Help window.

## Related operations

### To automatically load a project file

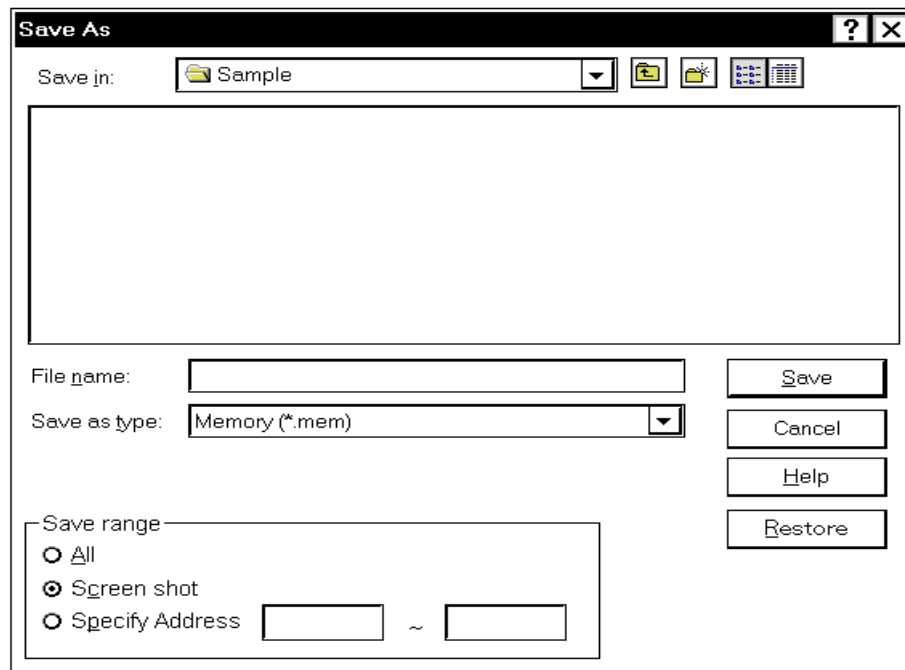
Use Project File (project file setting area) in Debugger Option dialog box that is opened when [Option] -> [Debugger Option...] is selected from the menu bar.

Select On for Auto Load in this area and specify the name of the project file to be loaded automatically in the Load Project File area. The specified project file will be automatically loaded when the system is started next time.

## View File Save dialog box

This dialog box is used to save the contents of the current window to a view file or event setting file when a dialog box is opened.

Figure 7-12 View File Save Dialog Box (Example: to Save Memory Window)



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

- When the window to be saved is the current window  
Select [File] -> [Save As...] from the menu bar  
(or press Alt+F and A in that order).

### Explanation of each area

This dialog box consists of the following areas:

- (1) Save in:, File name: (file selection area)
- (2) Save as type: (extension selection area)
- (3) Save range (save range setting area)

#### (1) Save in:, File name: (file selection area)

This area is used to specify a file name. A file name can be directly input from the keyboard, or selected from the list at the upper part of this area.

#### (2) Save as type: (extension selection area)

This area is used to specify the type (extension) of the file.

The extension of the file corresponding to the current window is displayed as follows:

Table 7-5 Extension Corresponding to Current Window

File Type (Extension)	Current Window Name
Source Text (*.SVW)	Source window
Assemble (*.DIS)	Assemble window
Memory (*.MEM)	Memory window
Watch(*.WCH)	Watch window
Register (*RGW)	Register window
I/O Register (*.IOR)	IOR window
Local Variable (*.LOC)	Local Variable window
Stack Trace (*.STK)	Stack Trace window
Event(*.EVN)	Event Manager
Console (*.LOG)	Console window
All (*.*)	All files

#### (3) Save range (save range setting area)

Save range

☐ All

☒ Screen shot

☐ Specify Address  ~

This area is displayed if the current window to be saved is the Assemble window, Memory window, Source window.

Specify the range of data to be saved by selecting any of the following:



All	Saves the entire range, from the first line to the last line.
Screen shot	Saves the area visible on the screen, from the top line on the screen to the bottom line. If the Source window is in the mixed display mode, however, the window contents are saved from the source line that includes the area visible on the screen.
Specify Line Specify Address above any	Specify the start line and end line of the area to be saved. If the start line and end line are omitted, the first line and last line are assumed. For details, it is as follows.

#### Specify Line (if current window is Source window)

☒ Specify Line     ~

Specify the range of the line numbers to be saved. The default radix for inputting a numeric value is decimal.

If the Source window is in the mixed display mode, the mixed displayed part on the specified line is also saved.

If a range of 100 lines or more is specified, a message dialog box is displayed to indicate the progress of saving. To stop saving midway, click the <Stop> button in the message dialog box.

#### Specify Address (if current window is Assemble window, or Memory window)

☒ Specify Address     ~

This area is used to specify the address range in which data is to be saved to the file. An address can be also specified by a symbol or expression (refer to "Table 7-10 Specifying Symbols"). The default radix for inputting a numeric value is hexadecimal.

If a range of 256 bytes or more is specified, a message dialog box is displayed to indicate the progress of saving. To stop saving midway, click the <Stop> button in the message dialog box.

#### Function buttons

Save	Saves the information of the current dialog box to the selected file. After saving, the dialog box is closed.
Cancel	Closes this dialog box without saving the file.
Help	Opens the Help window.
Restore	Restores the status before this dialog box was opened.

## View File Load dialog box

This dialog box is used to read the view file, source file, text file, event setting file, watch setting file, or peripheral I/O register setting file saved in the View File Save dialog box, and open a window.

The window to be opened and its status differ as follows, depending on the file to be loaded.

### Loading source file to which symbol information has been read

If there is a Source window in the active status, it is opened in the static status; otherwise, the Source window is opened in the active status.

### Loading source file to which symbol information has not been read, or view file

A window of text-format files is opened in the Source window in the static status.

### Loading event setting file

The Event Manager is opened, and the event setting contents will return to the previous environment.

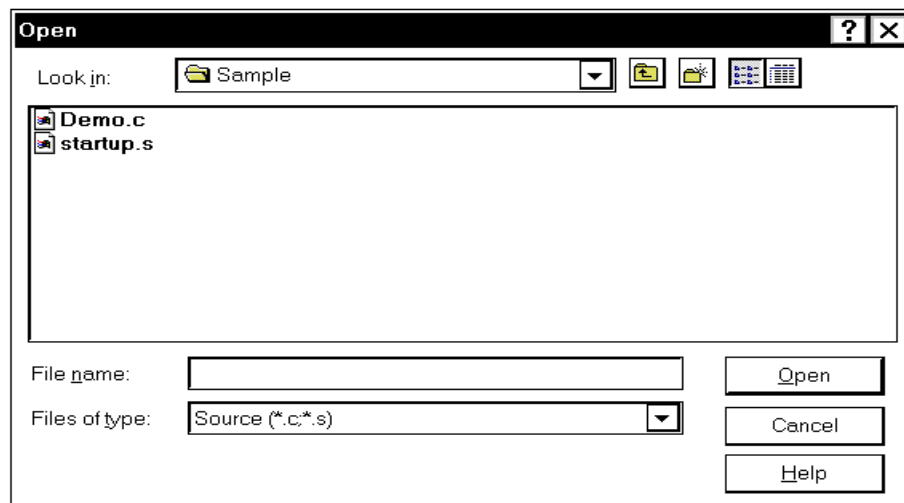
### Loading watch setting file

The Watch window is opened, and the watch setting contents will return to the previous environment.

### Loading peripheral I/O register setting file

The IOR window is opened, and the peripheral I/O register setting contents will return to the previous environment.

Figure 7-13 View File Load Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

## Opening

This dialog box can be opened as follows:

- Click the Open button.
- Select [File] -> [Open...] on the menu bar  
(or press Alt+F and O in that order, or press Ctrl+O).

## Explanation of each area

This dialog box consists of the following areas:

- (1) Look In:, File name: (file selection area)
- (2) Files of type: (extension selection area)

### (1) Look In:, File name: (file selection area)

This area is used to specify a file name. A file name can be directly input from the keyboard, or selected from the list at the upper part of this area.

Up to 64 character string with a extension can be specified.

### (2) Files of type: (extension selection area)

This area is used to specify the type (extension) of the file to be loaded.

Table 7-6 Type of File That Can Be Loaded on View File Load Dialog Box

File Type (Extension)	Meaning
Source (*.C, *.S)	Source file The extension of the source (c, s) can be changed in Debugger Option dialog box.
Text (*.TXT)	Text file
Source Text (*.SVW)	Source windowView files
Assemble (*.DIS)	Assemble window view files
Memory (*.MEM)	Memory window view files
Watch (*.WCH)	Watch window setting files <b>Caution</b> : Can not load the variable value.
Register (*.RGW)	Register window view files
I/O Register (*.IOR)	IOR window setting files <b>Caution</b> : Can not load the variable value.
Local Variable (*.LOC)	Local Variable window view files
Stack Trace (*.STK)	Stack Trace window view files
Event (*.EVN)	Event setting file
Console (*.LOG)	Console window
All (*.*)	All files

**Function buttons**

---

Open	Loads the selected file. After loading the file, this dialog box is closed.
Cancel	Closes this dialog box without loading the file.
Help	Opens the Help window.

## Download dialog box

This dialog box is used to select the name and format of a file to be downloaded, and downloads memory contents to the in-circuit emulator and the target system.

If a load module file has been downloaded, the corresponding source file is searched, and Source window is automatically opened.

Up to 20 load module files can be downloaded.

To check the loaded files, use Load Module List dialog box that is opened when [File] -> [Load Module] is selected from the menu bar.

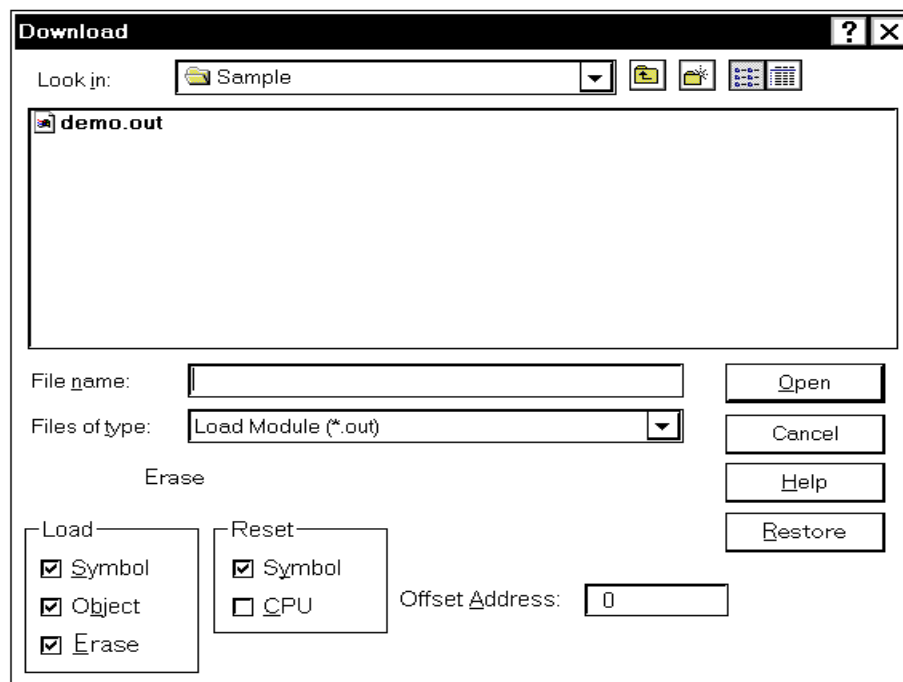
In this dialog box, a file can be downloaded to the internal flash memory (refer to "Flash Memory Writing Function").

### Format of file that can be downloaded

- Load module (ELF (.OUT))
- Intel Hex format (standard, extended)
- Motorola Hex format S type (S0, S2, S3, S7, S8)
- Extended Tektronix Hex format
- Binary data

**Caution** If a file other than a load module file is loaded, source debugging cannot be executed.

Figure 7-14 Download Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons
- Cancel processing

## Opening

This dialog box can be opened as follows:

- Click the **Load** button.
- Select [File] -> [Download...] on the menu bar (or press Alt+F, and D in that order).

## Explanation of each area

- (1) Look In:, File name: (file selection area)
- (2) Files of type: (extension selection area)
- (3) Load (load condition specification area)
- (4) Reset (reset condition specification area)
- (5) Offset Address: (offset setting area)

### (1) Look In:, File name: (file selection area)

This area is used to specify a file name. A file name can be directly input from the keyboard, or selected from the list at the upper part of this area.

Two or more files can be specified in this dialog box. To specify two or more files, delimit each file name with " " (double quotation mark). Files can also be specified by clicking the mouse button while holding down the Shift or Ctrl key.

Up to 64 character string with a extension can be specified.

### (2) Files of type: (extension selection area)

This area is used to specify the type (extension) of the file to be downloaded.

The contents displayed on the drop-down list are as follows:

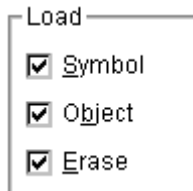
Table 7-7 Type of File That Can Be Downloaded

Contents (extension)	Format
Load Module(*.OUT)	Load module
Hex Format(*.HEX)	Hex format (automatic format identification)
Binary Data(*.BIN)	Binary data

**Reference** These are default extensions; other extensions can also be used.

The default extension of the displayed load module can also be specified under Default Extension (default extension specification area) in Debugger Option dialog box.

### (3) **Load** (load condition specification area)

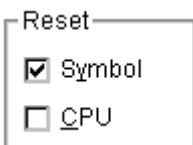


This area is used to set a load condition.

This setting is valid only if a file in the load module format is specified.

Symbol	Specifies whether symbol information is read (when checked, default) or not. The memory capacity can be saved by not reading symbol information when a program consisting of two or more load module files is to be debugged and if the symbol information of some modules does not have to be read.
Object	Specifies whether object information is read (when checked, default) or not.
Erase	Specifies whether the contents of the internal flash memory are erased all before download (when checked, default) or not (only a product with internal flash memory).

### (4) **Reset** (reset condition specification area)

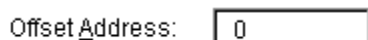


This area is used to set a reset condition.

This setting is valid only if a file in the load module format is specified.

Symbol	Specifies whether symbol information is reset (when checked, default) or not. When debugging a program consisting of two or more load module files, load each one of the load module files without resetting the symbol information. When downloading two or more load module files, take care that location addresses do not overlap.
CPU	Specifies whether the CPU is reset or not (not checked, default).

### (5) **Offset Address:** (offset setting area)



This area is used to specify the offset address that is used when a file is loaded (for binary data, specify the start address). An address can be also specified by a symbol or expression (refer to "Table 7-10 Specifying Symbols"). The default radix for inputting a numeric value is hexadecimal.

---

**Function buttons**

---

Open	Loads the selected file. After loading the file, this dialog box is closed.
Cancel	Closes this dialog box without loading the file.
Help	Opens the Help window.
Restore	Restores the input data to the original status.

---

**Cancel processing**

---

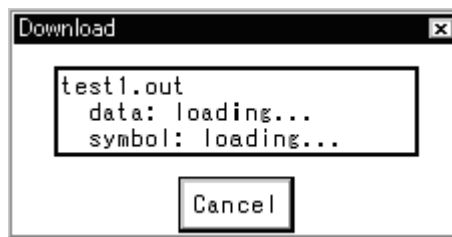
The following dialog box appears while downloading and the downloading can be cancelled at any time.

The progress of the download is displayed in the dialog box.

This dialog box is closed automatically after completing downloading.

If an error occurs during the download, an error message will be displayed in a new window.

Figure 7-15 The Progress of Download





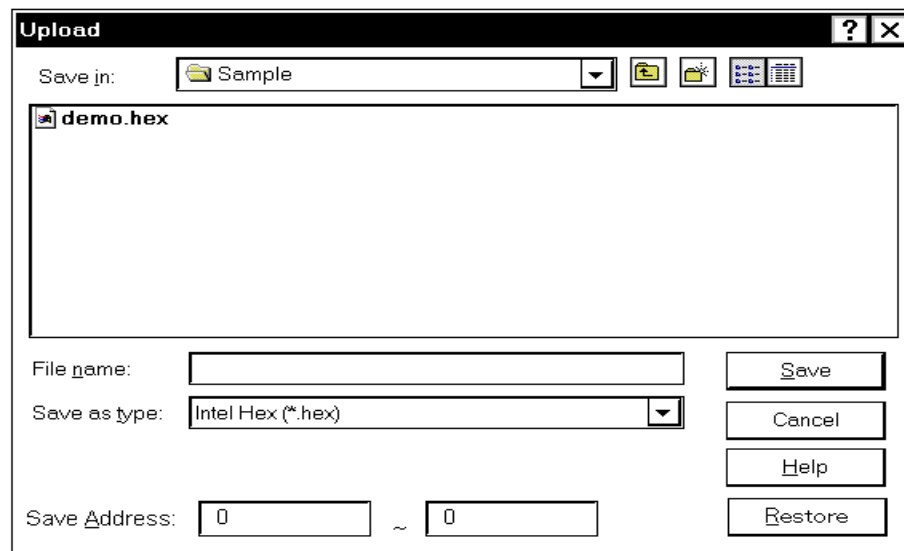
## Upload dialog box

This dialog box is used to set the name and format of the file to which memory contents are to be saved.

### Format of specifiable file

- Intel Hex format (extended ... 20-bit address)
- Motorola Hex format (S0, S3, S7 ... 32-bit address)
- Extended Tektronix Hex format
- Binary data format

Figure 7-16 Upload Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

- Select [File] -> [Upload...] on the menu bar (or press Alt+F, and U in that order).

## Explanation of each area

The Upload dialog box consists of the following areas:

- (1) Save in:, File name: (saved file name specification area)
- (2) Save as type: (extension selection area)
- (3) Save Address: (save range setting area)

### (1) Save in:, File name: (saved file name specification area)

This area is used to specify a file name. A file name can be directly input from the keyboard, or selected from the list at the upper part of this area.

Up to 64 characters can be specified including the extension.

### (2) Save as type: (extension selection area)

This area is used to specify the type (extension) of the file. The format of the data to be saved to the file is determined by the extension. If "All (\*.\*)" is selected, however, the default Intel extended Hex format is selected. The contents displayed on the drop-down list are as follows:

Table 7-8 Type of File That Can Be Uploaded

Contents (extension)	Format
Intel Hex (*.HEX)	Intel extended Hex format
Motorola Hex (*.HEX)	Motorola Hex format
Tektro Hex (*.HEX)	Extended Tektronix Hex format
Binary Data (*.BIN)	Binary data format
All (*.*)	All files

**Reference** These are default extensions; other extensions can also be used.

### (3) Save Address: (save range setting area)

Save Address:  ~

This area is used to specify the address range in which data is to be saved to the file. An address can be also specified by a symbol or expression (refer to "Table 7-10 Specifying Symbols").

The default radix for inputting a numeric value is hexadecimal.

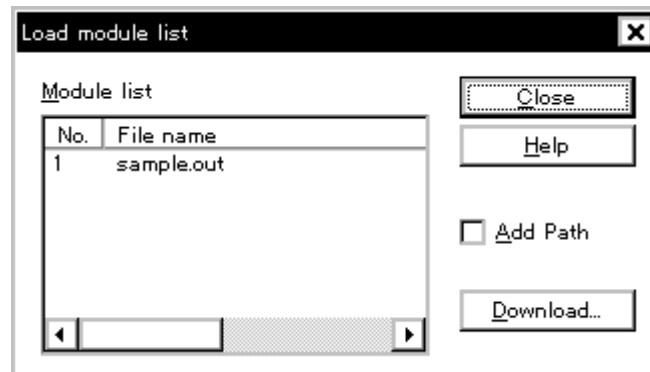
## Function buttons

Save	Saves data in the specified setting.
Cancel	Closes this dialog box without saving the data to the file.
Help	Opens the Help window.
Restore	Restores the input data to the original status.

## Load Module List dialog box

This dialog box can be used to check the file name and file path name of a load module that has already been downloaded. By using the <Download> button, The Download dialog box can be opened and a load module can be downloaded.

Figure 7-17 Load Module List Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

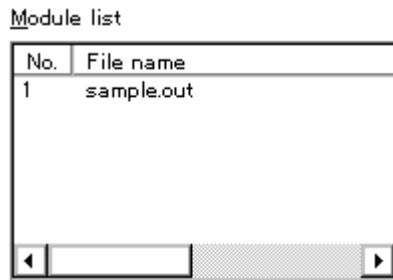
This dialog box can be opened as follows:

- Select [Eile] -> [Load Module...] on the menu bar (or press Alt+F, and L in that order).

### Explanation of each area

The Load module list dialog box consists of the following items:

- (1) Module list (file name display area)
- (2) Add Path (path name addition check box)

**(1) Module list (file name display area)**

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

The numbers displayed in the "No." field indicate the sequence in which the load module file names were read. The file names are displayed with the full path if the path name addition check box is checked; otherwise, only the file names will be displayed.

If the file names do not fit in the screen, the screen can be scrolled by using the mouse or the arrow keys of the keyboard. If symbol information has been reset in the Reset Debugger dialog box box, or if symbol information has been reset in the Download dialog box, the load module file names downloaded before that are cleared.

**(2) Add Path (path name addition check box)**

☐ Add Path

This area is used to specify whether load module file names are displayed with the path (checked) or only file names are displayed (not checked, default) in the file name display area.

**Function buttons**

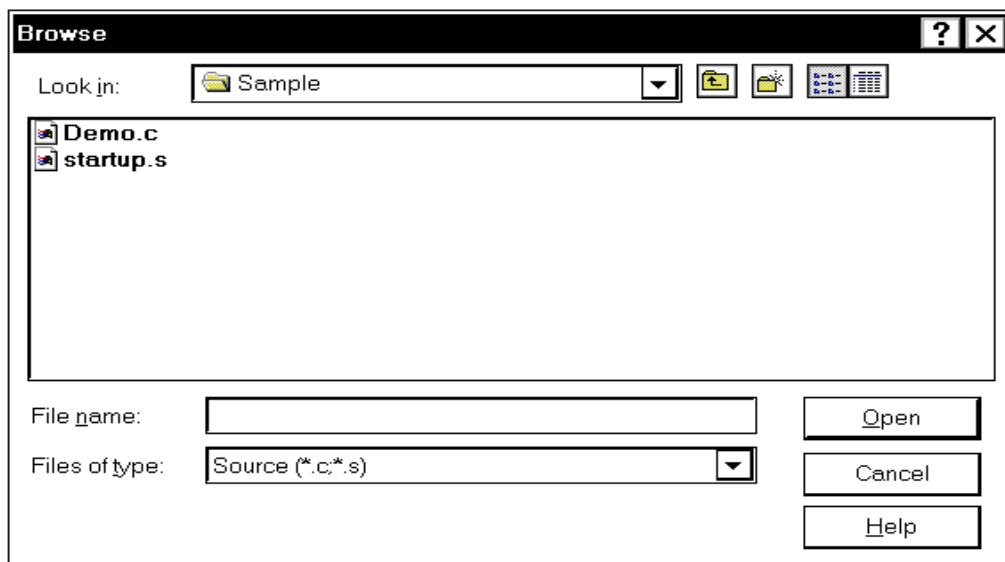
Close	Closes this dialog box.
Help	Opens the Help window.
Download...	Opens Download dialog box. A new load module can be downloaded. The file name of the newly downloaded module will be added to the file name display area when the Download dialog box is closed.

## Browse dialog box

This dialog box is used to select the file to be set in the Source Search dialog box or the Source Text move dialog box.

**Reference** If this dialog box is opened for the first time after the system has been started up, the directory first specified by the source path is displayed. When the dialog box is opened the second and subsequent times, the previously displayed directory is recorded and displayed again. If the <Cancel> button is pressed, however, the previously displayed directory is not recorded.

Figure 7-18 Browse Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

- Click the <Browse...> button in the Source Search dialog box or the Source Text move dialog box.

### Explanation of each area

The Browse dialog box consists of the following areas:

- (1) Look In:, File name: (file selection area)
- (2) Files of type: (extension selection area)

#### (1) Look In:, File name: (file selection area)

This area is used to specify a file name. A file name can be directly input from the keyboard, or selected from the list at the upper part of this area.

Up to 64 character string with a extension can be specified.

#### (2) Files of type: (extension selection area)

This area is used to specify the type (extension) of the file to be opened.

Table 7-9 Type of File That Can Be Displayed in Browse Dialog Box

File Type (Extension)	Meaning
Source (*.C, *.S)	Source file The extension of the source (c, s) can be changed in Debugger Option dialog box.
Text(*.TXT)	Text file
All(*.*)	All files

### Function buttons

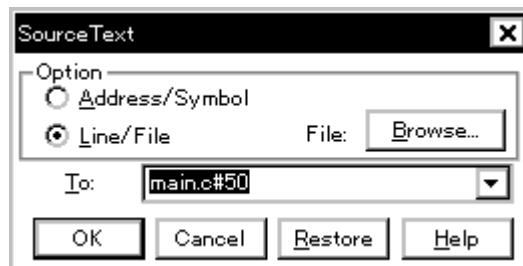
Open	Sets the selected file. After setting the file, closes this dialog box.
Cancel	Does not set the file and closes this dialog box.
Help	Opens the Help window.

## Source Text move dialog box

This dialog box is used to specify a file to be displayed in the Source window and the position from which displaying the file is to be started.

This dialog box is selected if [View] -> [Move...] is selected from the menu bar while the Source window is open, and enables the display start position of a file to be changed (a new window is not opened).

Figure 7-19 Source Text Move Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Specifying
  - Specifying a display start position by an address (or symbol).
  - Specifying a file to be displayed in the Source window
- Function buttons

### Opening

This dialog box can be opened as follows:

When the Source window is active

- Select [View] -> [Move...]  
(or press Alt+V and V in that order, or press Ctrl+J).

### Explanation of each area

The Source Move dialog box consists of the following areas:

- (1) Option (input mode selection area)
- (2) To (file/address specification area)

**(1) Option (input mode selection area)**

The image shows a dialog box titled 'Option'. It contains two radio buttons: 'Address/Symbol' and 'Line/File'. The 'Line/File' radio button is selected. To the right of the radio buttons is a text field labeled 'File:' followed by a 'Browse...' button.

This area is used to select the input mode when the display start position is specified.

Address/Symbol	If the display start position is specified by an address (or symbol)
Line/File	If the display start position is specified by a line number (or file name)

**(2) To (file/address specification area)**

The image shows a text field labeled 'To:' containing the text 'main.c#50'. There is a small downward arrow icon on the right side of the field.

This area is used to specify the file name or address to be displayed. In the default condition, the character string selected in the window that called this dialog box is displayed in this area. If no character string has been selected, the first character string of the input history is displayed. If no input history is available, the current PC is displayed. As necessary, the character string displayed in this area can be changed by directly inputting a character string from the keyboard.

By displaying the input history, the contents previously input can be reused. Up to 16 input histories can be recorded.

**Specifying**

On this dialog box, the following two items can be specified.

- Specifying a display start position by an address (or symbol).
- Specifying a file to be displayed in the Source window

**Specifying a display start position by an address (or symbol).**

Select **Address/Symbol** under **Option** and specify the address from which display is to be started for **To**.

An address can be also specified by a symbol or expression (refer to "Table 7-10 Specifying Symbols"). The default radix for inputting a numeric value is hexadecimal.

Clicking the <OK> button displays the source text so that the source line corresponding to the specified address value can be viewed.

**Specifying a file to be displayed in the Source window**

Select **Line/File** under **Option** and specify the name of the file to be displayed in the Source window for **To**. Specification can be made using only the file name, or using an absolute path or relative path.

If only the file name is specified, or if a relative path is specified, the files in the directory of the source path specified in Debugger Option dialog box are sequentially searched. The file that is found first as a result of searching is displayed.

In addition to a file name, a line number can be also specified as follows:

**<<path name>file name#>line number**



Delimit the file name from the line number with "#". Specify the line number in decimal. The file name may be omitted. The default radix for inputting a numeric value is hexadecimal.

Clicking the <OK> button displays the specified file with the specified line number as the first line. If the file name omitted, the file being displayed is displayed starting from the specified line. If the line number is omitted, the file is displayed starting from its first line.

---

**Function buttons**

---

Browse...	Opens the Browse dialog box.
OK	Starts displaying the source text from the specified position.
Cancel	Closes this dialog box.
Restore	Restores the input data to the original status.
Help	Opens the Help window.

## Address move dialog box

This dialog box is used to specify the address from which displaying memory contents, results of disassembly, or I/O registers are to be displayed. This dialog box is displayed if [View] -> [Move...] is selected from the menu bar while the Memory window, the Assemble window, or the IOR window is opened, and enables the display start address of data to be changed.

Figure 7-20 Address Move Dialog Box  
(Example: When Memory Window Is Open)



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

When the Memory window, Assemble window, or IOR window is opened

- Select [View] -> [Move...] from the menu bar  
(or press Alt+V and V in that order, or press Ctrl+J).

### Explanation of each area

The Address Move dialog box consists of the following areas:

- (1) To (address specification area)

**(1) To (address specification area)**

This area is used to specify an address. In the default condition, the string selected in the window that called this dialog box, the current PC value (Memory window, Assemble window), or blank (IOR window) is displayed.

An address can be also specified by a symbol or expression (refer to "Table 7-10 Specifying Symbols"). The default radix for inputting a numeric value is hexadecimal.

By displaying the input history, the contents previously input can be reused.

Up to 16 input histories can be recorded.

**Function buttons**

OK	Displays the memory contents, results of disassembly, or I/O registers.
Cancel	Closes this dialog box.
Restore	Restores the input data to the original status.
Help	Opens the Help window.

## Symbol To Address dialog box

Displays the address of the specified variable or function, or the value of the specified symbol.

Figure 7-21 Symbol To Address Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

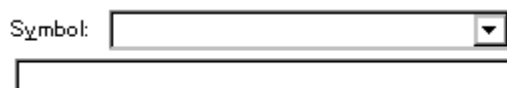
- Select [View] -> [Symbol...] from the menu bar (or press Alt+V and Y in that order).

### Explanation of each area

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

- (1) Symbol: (symbol specification and conversion result display areas)
- (2) Radix (display radix specification area)

#### (1) **Symbol:** (symbol specification and conversion result display areas)



The upper area (symbol specification area) is used to specify the variable, function name, symbol name, or line number to be converted.

In addition, an I/O port name, peripheral I/O register name, register name, PSW flag name, or expression can be also specified.

In the lower area (conversion result display area), the variable, address of the function, value of the symbol, address of the line number, or value of the expression specified under **Symbol** is displayed. The address value of an I/O port name or peripheral I/O register name, the register contents of a register name, or flag value of a PSW flag name is displayed. The specification method is indicated below.

Table 7-10 Specifying Symbols

Conversion Target	Specification Method
Variable	var file#var (to specify a static function with file name) func#var (to specify a static function with function name) file#var (to specify a static function with file name and function name)
Function	func file#func (to specify a static function with file name)
Label	label file#label(to specify a local label with file name)
Line number of source file	prog\$file#var file#no
I/O Port	portname
I/O Register	I/O regname
Register	regname
PSW Flag	pswname

**Remark** var: Variable name, func: Function name, 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: Line number of source file

"#" is used as a separator for file names, variables, function names, and line numbers.

If a specified symbol is not found in the scope, all symbols (static variables, static functions, local labels) are searched.

In the default status, a symbol name takes precedence. To temporarily change the priority, prefixing "\$" to a symbol gives the priority to a register name.

To specify a load module name when two or more load modules are read, use "\$" as a separator to delimit the load module name from a file name, variable, function name, or symbol name.

In the default condition, the character string selected in the window that called this dialog box is displayed. As necessary, input data from the keyboard to change the character string. The default radix for inputting a numeric value is decimal.

To change the contents of the symbol specification area, click the <OK> button. The conversion result will be displayed in the conversion result display area.

By displaying the input history, the contents previously input can be reused. Up to 16 input histories can be recorded.

**(2) Radix (display radix specification area)**

Radix: ☒ Hex ☐ Dec ☐ Oct ☐ Bin

This area is used to specify the radix of the data to be displayed in the conversion result display area.

Hex	Hexadecimal number (default).
Dec	Decimal number
Oct	Octal number
Bin	Binary number

**Function buttons**

OK	If the contents of the symbol specification area have been changed, converts the symbol. After conversion, closes the dialog box if the contents of the symbol specification area have not been changed.
Cancel	Closes this dialog box.
Restore	Restores the input data to the original status. If the <OK> button has already been clicked, the data is restored to the status immediately after the <OK> button was clicked.
Help	Opens the Help window.

## Source window

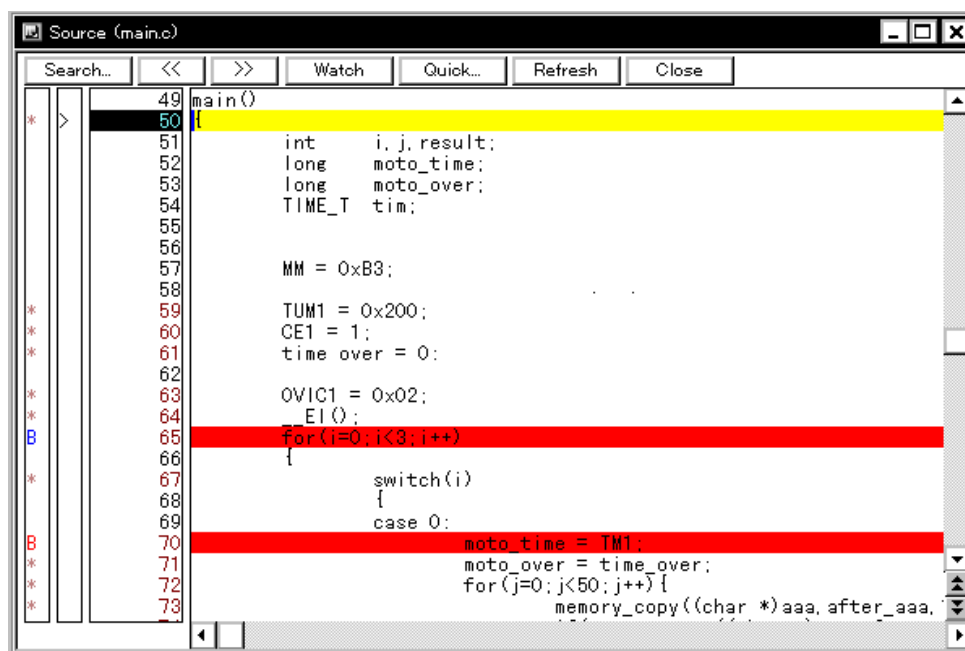
This window displays source files or text files.

In Mixed display mode, the program can be disassembled and displayed and online assembly can be executed while source files are being displayed.

The Source window that is displaying the source file to which symbol information is read may be in two statuses: Active Status and Static Status.

**Caution** If program codes is described in an include file and these codes are included in multiple files, the line numbers and addresses do not correspond on a one-to-one bases. In such an include file, function that indicates the correspondence relationship between line numbers and addresses dose not correctly operate.

Figure 7-22 Source Window



This section explains the following items:

- Opening
- Explanation of each area
  - Breakpoint setting/deletion function
- Functions often used (right-click menu)
- Function buttons
- Related operations

**Concerning display**

- Mixed display mode
- To set a source path
- To change the display start position
- To display another source file
- To specify the default extension of a source file
- To convert a symbol
- To specify the symbol of a start-up routine
- To set an opened file history
- To set a beep sound to be issued upon break
- To change the displayed font
- To change the displayed tab size

**Concerning operation**

- [Start From Here] function
- [Come Here] function
- Program counter setting function
- Jump Function
- To change disassembled contents
- Watch function (referencing data value)
- To search in a source program
- To set a default breakpoint
- Drag & drop function (line/address)
- Drag & drop function (character string)

---

**Opening**

---

This window can be opened as follows:

- This window is automatically opened if the corresponding source file exists after the download module file has been downloaded.
- Click the **Src** button.
- Select [**B**rowse] -> [**S**ource] from the menu bar (or press Alt+B, and S in that order).
- Select [**J**ump] -> [**S**ource] from the menu bar (or press Alt+J and S in that order, or press Ctrl+U).
- Click the <Open> button with selecting a source file or text file in the View File Load dialog box.



## Explanation of each area

The Source window consists of the following areas:

- (1) Point mark area
- (2) Current PC mark area
- (3) Line number/address display area
- (4) Source text display area

### (1) Point mark area



This area displays the Breakpoint setting/deletion function, Event setting status, and Program code.

The 'B' mark is displayed in the following colors, depending on the type and status of the breakpoint.

A breakpoint becomes valid as soon as it has been set in the point mark area.

Table 7-11 Display Color of Breakpoint and Its Meaning

Display Color	Meaning
B (blue)	Software breakpoint is set.
B (red)	Valid hardware breakpoint (after execution) is set.
B (green)	Valid hardware breakpoint (before execution) is set. <b>Remark:</b> Breaks before execution (B) are set with priority.
B (black)	Invalid hardware breakpoint is set. This hardware breakpoint can be validated on the Event Manager or in the Break dialog box.

### Breakpoint setting/deletion function

By clicking this area with the mouse, a breakpoint can be set or deleted. Whether a hardware breakpoint or software breakpoint is to be set or deleted can be selected by the right-click menu of this area (or by On Mouse Click in the Extended Option dialog box).

#### Setting a breakpoint

A breakpoint can be set on a line for which "" is displayed in the point mark area (if "" is not displayed for the line, the breakpoint is set on the line above or below the line, whichever has "" displayed). If a breakpoint is set on a line on which an event breakpoint has already been set, "A" indicating that multiple events have been set is displayed.

#### Deleting a breakpoint

Click the position at which the breakpoint to be deleted is set.

If right-click menu -> [Software Breakpoint] is selected, 'B (blue: software breakpoint)' is deleted. If [Hardware Breakpoint] is selected, 'B (red: hardware break)' is deleted. If another event remains,

however, the mark of that event is displayed.

Table 7-12 Details of Setting/Deleting Breakpoint

Display Status of Line	On Mouse Click [right-click menu]	Operation on clicking
Blank or with mark other than B	Soft Break [Software Breakpoint]	Sets a software breakpoint.
	Hard Break [Breakpoint]	Sets a hardware breakpoint
With B(blue) mark (Software breakpoint setting status)	Soft Break [Software Breakpoint]	Deletes a software breakpoint
	Hard Break [Breakpoint]	Sets a hardware breakpoint
With B(red, green, black) mark (Hardware breakpoint setting status)	Soft Break [Software Breakpoint]	Sets a software breakpoint.
	Hard Break [Breakpoint]	Deletes a hardware breakpoint

### Event setting status

The setting status of various events can be displayed.

If an execution event or access event is set on the corresponding line, a mark corresponding to the type of event is displayed.

If an address range is specified as the address condition of the event, the lower addresses of the range are displayed.

The mask specification of the address condition is not reflected.

Table 7-13 Meaning of Event Display Mark

Mark	Meaning
<b>E</b>	Setting of event condition
<b>L</b>	Setting of event for event link condition
<b>B</b>	Setting of break event
<b>A</b>	Setting of multiple events

### Program code

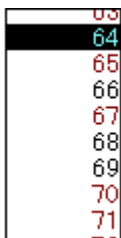
The line marked '\*' in the point mark area indicates that a program code corresponding to the line number exists. This program code is displayed only if a load module has been downloaded and symbol information has been read.

**(2) Current PC mark area**

The mark '>', which indicates the current PC value (PC register value), is displayed in this area.

Clicking this mark with the mouse displays a pop-up window that shows the PC register value.

By double-clicking the current PC mark area, the program can be executed up to a specified line.

**(3) Line number/address display area**

This area displays the line numbers of a source file or text file.

If a Program code corresponding to a line number of the source file exists, the character of the line number is displayed in **red**. If a program code does not exist, the character of the line number is displayed in black. In the mixed display mode, disassembled addresses are displayed. The characters of an address are displayed in gray.

In this area, any position can be selected (highlighted) by clicking the mouse button. The selected line number of an address can be dragged to another window or area, so that its variable value can be displayed or the line number of the address can be specified as an event condition (refer to "Drag & drop function (line/address)").

**(4) Source text display area**

This area displays source files or text files. In the mixed display mode, disassemble display and online assemble can also be executed (refer to "To change disassembled contents").

A data value can be temporarily displayed by selecting (highlighting) a character string and placing the mouse cursor over it (Balloon watch function).

The current PC line and the line on which a breakpoint is set are emphasized for display, as follows:

Current PC line (Yellow)	This line is the source line or disassemble display line corresponding to the current PC value (indicated by '>').
Breakpoint setting line (red)	This line is the source line or a disassemble display line on which a valid breakpoint is set (indicated by a 'B' mark in red, green or blue).

In the mixed display mode, the color of only the disassemble display line is changed (the source line is displayed in the normal color).

In this area, the following functions can be used on the line or address on which the cursor is placed (start address of program code):

- > [Start From Here] function
- > [Come Here] function
- > Program counter setting function
- > Jump Function

**Caution** If a Program code does not exist on the source line, the top address of the line above or below the line on which a program code exists is manipulated by these functions.

These functions cannot be performed in the following cases. The corresponding menu will be dimmed and cannot be selected.

- If a file other than a source file is displayed
- While the user program is being executed

A character string selected in this area can be dragged and dropped to another window or dialog box (refer to "Drag & drop function (character string)").

#### Functions often used (right-click menu)

In this window, functions that are often used can be selected from the menu that is displayed by clicking the right mouse button. In this way, the same function can be used more easily than with the normal procedure.

Move...	Moves the display position. Opens the Source Text move dialog box.
Mix	Turns on/off mixed display of disassembly. Checked: Mixed display Not checked: No display (default)
Add Watch...	Adds the specified data to the Watch window. Opens the Add Watch dialog box.
Symbol...	Displays the address of the specified variable or function, or the value of the specified symbol. Opens the Symbol To Address dialog box.
Break when Access to this Variable	Creates a break event for a read/write access.
Break when Write to this Variable	Creates a break event for a write access.
Break when Read from this Variable	Creates a break event for a read access.
Clear	Deletes the selected break event condition.
Event?	Displays the event information of the line at the cursor position or the selected variable. If an event is set, Event dialog box is opened.
Come Here	Executes the program from the current PC to the cursor position.
Change PC	Sets the address at the cursor position to the PC.

Break Point	Sets or deletes a breakpoint at the cursor position. <b>Remark:</b> Breaks before execution (B) are set with priority.
Software Break Point	Sets or deletes a software break point at the cursor position.
Assemble	Disassembles and displays starting from the jump destination address specified by the data value at the cursor position. Opens the Assemble window. If an active Assemble window is open, that window is displayed in the forefront (so that it can be manipulated).
Memory	Displays the memory contents starting from the jump destination address specified by the data value at the cursor position. Opens the Memory window. If an active Memory window is open, that window is displayed in the forefront (so that it can be manipulated).

### Function buttons

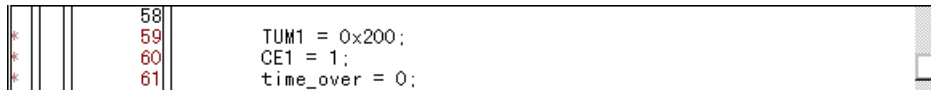
Search...	Opens the Source Search dialog box and searches a character string of the source text. If a character string is selected in the source text display area, the Source Search dialog box is opened to search the character string. If no character string is selected, the Source Search dialog box is opened with nothing specified to be searched. Specify a search method in the Source Search dialog box. The result of search is highlighted in the Source window. This is the same operation as selecting [View] -> [Search...] from the menu bar.
<<	Searches forward (upward on screen) for the text that satisfies the search condition set in the Source Search dialog box, starting from the address at the cursor position. This button is displayed as the <Stop> button during a search.
>>	Searches backward (downward on screen) for the text that satisfies the search condition set in the Source Search dialog box, starting from the address at the cursor position. This button is displayed as the <Stop> button during a search.
Stop (during search)	Stops searching.
Watch	Adds the variables selected in the source text display area to the Watch window. If the Watch window is not opened, it is opened. If no text is selected in the source text display area, the Watch window is only opened. This is the same operation as selecting [View] -> [View Watch] from the menu bar.
Quick...	Temporarily displays the contents, such as a variable, selected in the source text display area in the Quick Watch dialog box. Opens the Quick Watch dialog box. If no text is selected in the source text display area, the Quick Watch dialog box is only opened. This is the same operation as selecting [View] -> [Quick Watch] from the menu bar.
Refresh	Updates the contents of the window with the latest data.
Close	Closes this window.

## Related operations

### Mixed display mode

The normal display mode and mixed display mode can be alternately selected by selecting [View] -> [Mix] from the menu bar.

#### Normal display mode



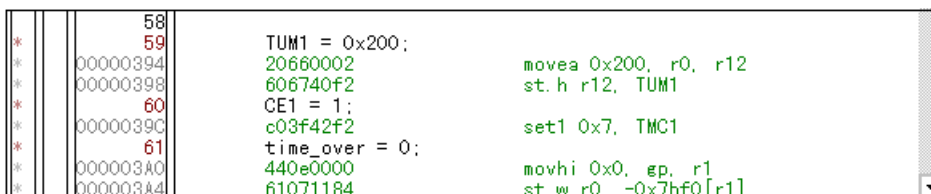
```

58
* 59 TUM1 = 0x200;
* 60 CE1 = 1;
* 61 time_over = 0;

```

In the normal display mode, general text files can be displayed as well as source files.

#### Mixed display mode



```

58
* 59 TUM1 = 0x200;
* 00000394 20660002 movea 0x200, r0, r12
* 00000398 606740f2 st.h r12, TUM1
* 60 CE1 = 1;
* 0000039C c03f42f2 setl 0x7, TMC1
* 61 time_over = 0;
* 000003A0 440e0000 movhi 0x0, gp, r1
* 000003A4 61071184 st.w r0, -0x7bf0[r1]

```

In the mixed display mode, disassembly of programs combined with source files is displayed and online assembly is carried out. (This mode is valid only when a load module is being downloaded and symbol information is being read, and the source file that corresponds to the downloaded load module is displayed).

If a program code corresponds to the line of the displayed source file, the disassembly line is displayed next to the source line. In the disassembly line, an address label, a code data, and a disassembled mnemonic are displayed (the starting position for display of the mnemonic can be adjusted by the tab size setting value).

To execute online assembly, move the cursor to a mnemonic. The results of online assembly are also reflected in the Memory window.

The contents displayed in the mixed display mode can be saved as a view file.

### To set a source path

Set a source path under Source Path (source path specification area) in the Debugger Option dialog box that is opened by selecting [Option] -> [Debugger Option] from the menu bar.

### To change the display start position

Change the display start position in the Source Text move dialog box that is opened when right-click menu -> [Move...] is selected in the Source window.

### To display another source file

Select [File] -> [Open] from the menu bar and specify a file.

### To specify the default extension of a source file

Set a source path under Default Extension (default extension specification area) in the Debugger Option dialog box that is opened by selecting [Option] -> [Debugger Option] from the menu bar.

**To convert a symbol**

Convert symbols in the Symbol To Address dialog box that is opened when right-click menu -> [Symbol...] is selected in the Source window.

**To specify the symbol of a start-up routine**

Specify the symbol of a start-up routine under Startup Routine (start-up symbol setting area) in the Debugger Option dialog box that is opened by selecting [Option] -> [Debugger Option...] from the menu bar. If the setting in this area is wrong, the source file cannot be opened.

**To set an opened file history**

Set an opened file history under Open File History (open file history setting area) in the Debugger Option dialog box that is opened by selecting [Option] -> [Debugger Option...] from the menu bar. The number of a history display of the opened files can be specified in this area.

**To set a beep sound to be issued upon break**

Set the beep sound under Break Sound (beep sound specification area) in the Extended Option dialog box that is opened by selecting [Option] -> [Extended Option...] from the menu bar.

**To change the displayed font**

Change the displayed font in the Font dialog box that can be opened by clicking the <Font...> button in the Debugger Option dialog box that is opened when [Option] -> [Debugger Option...] is selected from the menu bar.

**To change the displayed tab size**

The displayed tab size can be specified for each file extension under Tab Size (tab size setting area) in the Debugger Option dialog box that is opened when [Option] -> [Debugger Option...] is selected from the menu bar.

**[Start From Here] function**

This function is used to execute the user program from the line on which the cursor is placed.

**Method**

Move the cursor to the line from which the program is to be executed, and select right-click menu -> [Start From Here].

**[Come Here] function**

This function is used to execute the user program to the line on which the cursor is placed (while the user program is being executed in this mode, the break event currently set does not occur).

**Method**

Move the cursor to the line on which the break is to occur, and select right-click menu -> [Come Here].

**Program counter setting function**

This function is used to set the address of the line on which the cursor is placed to the program counter (PC).

**Method**

Move the cursor to the line that is to be set to the PC, and select [Run] -> [Change PC] from the menu

bar.

### **To change disassembled contents**

To change the disassembled contents, move the cursor to the mnemonic field (the overwrite and insertion modes are alternately selected by pressing the Insert key).

If an attempt is made to move the cursor to another line after the disassembled contents have been changed in the mnemonic field, the new contents are checked. If the new contents are illegal, the code data on the line where the contents have been changed is indicated as "\*\*". In this case, the new contents are not written to memory even if the cursor is moved to another line.

The contents changed in the mnemonic field are written into the memory by pressing the Enter key. When the Enter key is pressed, the new contents are checked and, if they are illegal, the code data on the illegal line is indicated by "\*\*". If even one line is illegal, the new contents are not written into the memory.

In this case, correct the contents. To discard the contents, press the ESC key. If the contents are correct and if the Enter key is pressed, the contents are written into the memory, and then the cursor moves to the next line in the mnemonic field, so that the data on the next line can be changed.

### **Watch function (referencing data value)**

Data values can be displayed as follows:

#### **To temporarily display a value**

The following two methods can be used:

- Select (highlight) a character string and put the mouse cursor on it then the data value is displayed (Balloon watch function).
- Select (highlight) a character string and click the <Quick...> button. The Quick Watch dialog box will be opened to display the data value.

#### **To trace the change of a data value**

Register a variable or symbol to the Watch window in either of the following ways:

- Select (highlight) the character string whose data value is to be displayed, click the <Watch> button, or drag the character string to the Watch window and register it to the window (Drag & drop function (character string)).
- Select (highlight) the character string whose a data value is to be displayed and register the data value from the Add Watch dialog box that is opened when right-click menu -> [Add Watch] is selected.

### **To search in a source program**

Use the Source Search dialog box that is opened by clicking the <Search...> button in the Source window.

If the character string to be searched exists in the window, select that character string and click the <Search...> button. The selected character string will be opened in the specified format.

### **To set a default breakpoint**

Set a default breakpoint under On Mouse Click (default break selection area) in the Extended Option dialog box that is opened when [Option] -> [Extended Option...] is selected from the menu bar. In this



way, a default breakpoint that is assumed when the point mark area is clicked can be set.

#### Drag & drop function (line/address)

The line number or address selected and highlighted can be dragged to another window or area.

If the position of the line number or address selected in this area is dragged with the left mouse button, the shape of the mouse cursor changes from an arrow to '-'. When the cursor is moved to the window or area in which they can be dropped, the mouse cursor changes to 'OK'.

In the window in which the line number or address has been dropped, an operation is performed on the dropped address or the address that is obtained from the dropped line number. The operation to be performed after dropping the line number or address differs, depending on the window or area in which the line number or address has been dropped.

Table 7-14 Details of Drag & Drop Function (Line/Address)

Window/Area to Drop to	Operation After Drop
The Event Manager or the event manager area in each various event setting dialog box	Automatically creates an execution event condition by using the dropped line number or address as an address condition. Event condition names are automatically created as Evt00001, Evt00002, and so on. The address condition is set for the closest symbol in the format of symbol name + offset value.
Condition setting area in each various event setting dialog box (other than address and data setting areas)	Automatically creates an execution event condition by using the dropped line number or address as an address condition. The automatically created event condition is set in each condition setting area in which the line number or address has been dropped. Event condition names are automatically created as Evt00001, Evt00002, and so on. The address condition is set for the closest symbol in the format of symbol name + offset value.
Condition setting area in each various event setting dialog box (address and data setting areas)	The text of the dropped line number or address is set in the area in which the line number or address has been dropped. The address condition is set for the closest symbol in the format of symbol name + offset value.

**Caution** The various event setting dialog boxes includes the Event dialog box, Event Link dialog box, and Break dialog box.

#### Drag & drop function (character string)

A selected text can be moved to another window or area by dragging and dropping.

- 1 If the selected text is dragged by using the left mouse button, the shape of the mouse cursor changes from an arrow to '-'.
- 2 When the cursor is moved to the window or area in which the text can be dropped, the mouse cursor changes to 'OK'.

The operation to be performed after dropping the text differs, depending on the window or area in which the text has been dropped.

Table 7-15 Details of Drag &amp; Drop Function (Character String)

Window/Area to Drop to	Operation After Drop												
The Event Manager or the event manager area in each various event setting dialog box	<p>If the dropped text can be converted as a symbol into an address value, an event condition in the Access status (all access statuses) or Execute status is automatically created, using the converted address value as an address condition. Event condition names are automatically created as Evt00001, Evt00002, and so on.</p> <p>The address condition is set by the dropped text.</p> <p>The relationship between the event condition to be created and the symbol is as follows:</p> <table> <tr> <th>Symbol</th><th>Status</th></tr> <tr> <td>Variable</td><td>Access(R/W)</td></tr> <tr> <td>Function</td><td>Execute</td></tr> <tr> <td>Symbol in the data section</td><td>Access(R/W)</td></tr> <tr> <td>Symbol in the code section</td><td>Execute</td></tr> <tr> <td>Other</td><td>Access(R/W)</td></tr> </table>	Symbol	Status	Variable	Access(R/W)	Function	Execute	Symbol in the data section	Access(R/W)	Symbol in the code section	Execute	Other	Access(R/W)
Symbol	Status												
Variable	Access(R/W)												
Function	Execute												
Symbol in the data section	Access(R/W)												
Symbol in the code section	Execute												
Other	Access(R/W)												
Condition setting area in each various event setting dialog box (other than address and data setting areas)	<p>If the dropped text can be converted as a symbol into an address value, an event condition in the Access status (all access statuses) or Execute status is automatically created, using the converted address value as an address condition.</p> <p>The automatically created event condition is set in each condition setting area in which the line number or address has been dropped.</p> <p>Event condition names are automatically created as Evt00001, Evt00002, and so on.</p> <p>The address condition is set by the dropped text.</p> <p>The relationship between the event condition to be created and the symbol is as follows:</p> <table> <tr> <th>Symbol</th><th>Status</th></tr> <tr> <td>Variable</td><td>Access(R/W)</td></tr> <tr> <td>Function</td><td>Execute</td></tr> <tr> <td>Symbol in the data section</td><td>Access(R/W)</td></tr> <tr> <td>Symbol in the code section</td><td>Execute</td></tr> <tr> <td>Other</td><td>Access(R/W)</td></tr> </table>	Symbol	Status	Variable	Access(R/W)	Function	Execute	Symbol in the data section	Access(R/W)	Symbol in the code section	Execute	Other	Access(R/W)
Symbol	Status												
Variable	Access(R/W)												
Function	Execute												
Symbol in the data section	Access(R/W)												
Symbol in the code section	Execute												
Other	Access(R/W)												
Condition setting area in each various event setting dialog box (address and data setting areas)	The dropped text is set in the area.												
Watch window	If the dropped text is recognizable as a symbol, the contents of the symbol are displayed.												

**Caution** The various event setting dialog boxes includes the Event dialog box, Event Link dialog box, and Break dialog box.

## Source Search dialog box

This dialog box is used to search for the character string in the Source window.

By setting each item and then clicking the <Find Next> button, searching can be started.

By clicking the <Set Find> button, the direction buttons ("<<" and ">>") in the Source window can be used for the search.

Figure 7-23 Source Search Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

When Source window is displayed

- Click the <Search> button.  
(or press Ctrl and G in that order).
- Select [View] -> [Search...]  
(or press Alt+V, and S in that order).

### Explanation of each area

The Source Search dialog box consists of the following areas:

- (1) Find What (search data specification area)
- (2) Match Case (search condition specification area)
- (3) Direction (search direction specification area)
- (4) File (search file specification area)

**(1) Find What (search data specification area)**

Find What:

This area is used to specify the data to be searched.

In the default condition, the character string selected in the window that called this dialog box is displayed. As necessary, input data from the keyboard to change the character string.

Up to 256 character string can be specified.

By displaying the input history, the contents previously input can be reused. Up to 16 input histories can be recorded.

**(2) Match Case (search condition specification area)**

☒ Match Case

This area is used to specify whether uppercase and lowercase characters are distinguished, before searching is started.

Checked	Distinguished (default)
Not checked	Not distinguished

**(3) Direction (search direction specification area)**

Direction  
☐ Up ☒ Down

This area is used to specify the direction of the search.

Up	Forward search. Searches data forward (upward on screen) from the current position of the cursor.
Down	Backward search (default). Searches data backward (downward on screen) from the current position of the cursor.

**(4) File (search file specification area)**

File:

This area is used to specify the file to be searched.

If no file is specified, searching is executed in the window that called this dialog box.

**Function buttons**

---

Find Next	Searches the specified data in accordance with a given condition. If the specified character string is found as a result of a search, it is highlighted. To continue searching, click this button again.
Set Find	Sets the specified condition as the search condition and closes this dialog box.
Stop (during search)	Stops searching.
Cancel	Closes this dialog box. This button is displayed as the <Stop> button while data is being searched.
Browse...	Opens Browse dialog box.
Help	Opens the Help window.

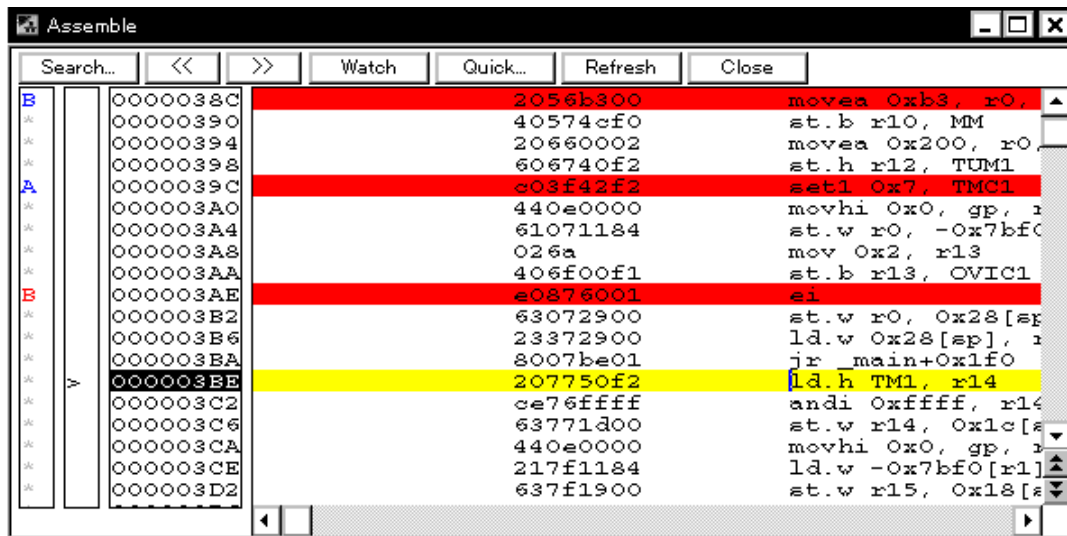
## Assemble window

This window is used to disassemble and display programs. It is also used to execute online assembly.

To execute online assembly, move the cursor to a mnemonic. The results of online assembly are also reflected in the Memory window.

The Assemble window has two statuses: Active Status and Static Status.

Figure 7-24 Assemble Window



This section explains the following items:

- Opening
- Explanation of each area
- Functions often used (right-click menu)
- Function buttons
- Related operations

### Concerning display

- To change the display start position
- To change the displayed offset and register name

### Concerning operation

- [Start From Here] function
- [Come Here] function
- Program counter setting function
- Jump Function
- To change disassembled contents

- Watch function (referencing data value)
- To search in the Assemble window
- Drag & drop function (line/address)
- Drag & drop function (character string)

## Opening

---

This window can be opened as follows:

- Click the **Asm** button.
- Select [**B**rowse] -> [**A**ssemble] from the menu bar (or press Alt+B, and A in that order).
- Select [**J**ump] -> [**A**ssemble] from the menu bar (or press Alt+J and A in that order, or press Ctrl+D).

## Explanation of each area

---

The Assemble window consists of the following areas:

- (1) Point mark area
- (2) Current PC mark area
- (3) Address specification area
- (4) Disassemble display area

### (1) Point mark area



This area is displayed the Breakpoint setting/deletion function and the Event setting status.

### (2) Current PC mark area



The mark '>', which indicates the current PC value (PC register value), is displayed in this area.

By double-clicking the current PC mark area, the program can be executed up to a specified line.

**(3) Address specification area**

```

000003AA
000003AE
000003B2
000003B6
000003BA
000003BE
000003C2

```

This area displays the disassembly start address. When this area is clicked, an address highlighted and selected.

This area has the Drag & drop function (line/address).

**(4) Disassemble display area**

028a	mov 0x2, r13
406f00f1	st.b r13, OVIC1
e0878001	ai
63072900	st.w r0, 0x28[sp]
23372900	ld.w 0x28[sp], r1
8007be01	jr main+0x1f0
207750f2	ld.h TM1, r14
ce76ffff	andi 0xffff, r14
63771d00	st.w r14, 0x1c[sp]
440e0000	movhi 0x0, gp, r1
217f1184	ld.w -0x7bf0[r1]
637f1900	st.w r15, 0x18[sp]

This area displays the labels and code data of addresses, and disassembled mnemonics.

The current PC line and breakpoint setting line are displayed in different colors for emphasis.

The disassembled mnemonics displayed in the mnemonic field can be changed (refer to "To change disassembled contents").

If the number of new instruction bytes is less than the number of previous instruction bytes as a result of changing, as many 'NOP' instructions as necessary are inserted. If the number of new instruction bytes is more than the number of previous instruction bytes, the next instruction is overwritten. In this case also, as many 'NOP' instructions as necessary are inserted. The same applies to instructions that straddle over source lines.

In this area, the following functions can be used on the line or address on which the cursor is placed (start address of program code):

- > [Start From Here] function
- > [Come Here] function
- > Program counter setting function
- > Jump Function

A character string selected in this area can be dragged and dropped in another window or dialog box (refer to "Drag & drop function (character string)").

**Functions often used (right-click menu)**

In this window, functions that are often used can be selected from the menu that is displayed by clicking the right mouse button. In this way, the same function can be used more easily than with the normal procedure.



Move...	Moves the display position. Opens the Address move dialog box.
Add Watch...	Adds the specified data to the Watch window. Opens the Add Watch dialog box.
Symbol...	Displays the address of the specified variable or function, or the value of the specified symbol. Opens the Symbol To Address dialog box.
Come Here	Executes the program from the current PC to the cursor position.
Change PC	Sets the address at the cursor position to the PC.
Break Point	Sets or deletes a breakpoint at the cursor position. <b>Remark:</b> Breaks before execution (B) are set with priority.
Software Break Point	Sets or cancels a software break point at the cursor position.
Source Text	Displays the corresponding source text and source line, using the data value at the cursor position as the jump destination address. If no line information exists at the jump destination address, however, you cannot jump. Opens the Source window. If an active Source window is open, that window is displayed in the forefront (so that it can be manipulated).
Memory	Displays the memory contents starting from the jump destination address specified by the data value at the cursor position. Opens the Memory window. If an active Memory window is open, that window is displayed in the forefront (so that it can be manipulated).

### Function buttons

Search...	Opens the Assemble Search dialog box and searches for a character string of mnemonics. If a character string is selected in the disassemble display area, the Assemble Search dialog box is opened to search for the character string. If no character string is selected, the Assemble Search dialog box is opened with nothing specified to be searched. Specify a search method in the Assemble Search dialog box. The result of search is highlighted in the Assemble window. This is the same operation as selecting [View] -> [Search...] from the menu bar.
<<	Searches forward (upward on screen) for the contents that satisfy the search condition set in the Assemble Search dialog box, starting from the address at the cursor position. This button is displayed as the <Stop> button during a search.
>>	Searches backward (downward on screen) for the contents that satisfy the search condition set in the Assemble Search dialog box, starting from the address at the cursor position. This button is displayed as the <Stop> button during a search.
Stop (during search)	Stops searching.

Watch	Adds the symbols selected in the disassemble display area to the Watch window. If the Watch window is not opened, it is opened. If no text is selected in the disassemble display area, the Watch window is only opened. This is the same operation as selecting [View] -> [View Watch] from the menu bar.
Quick...	Temporarily displays the contents, such as symbols, selected in the disassemble display area on the Quick Watch dialog box. Opens the Quick Watch dialog box. If no text is selected in the disassemble display area, the Quick Watch dialog box is only opened. This is the same operation as selecting [View] -> [Quick Watch...] from the menu bar.
Refresh	Updates the contents of the window with the latest data.
Close	Closes this window.

### Related operations

#### To change the display start position

Change the display start position in the Address move dialog box, which is opened when right-click menu -> [Move...] is selected in the Assemble window.

#### To change the displayed offset and register name

Change the display offset or register name under DisAssemble (offset display setting area) in the Debugger Option dialog box that is opened when [Option] -> [Debugger Option...] is selected from the menu bar. A function name or an absolute name can be selected as a register name.

#### To search in the Assemble window

Use the Assemble Search dialog box, which can be opened by clicking the <Search...> button in the Assemble window. If the character string to be searched is existed in the window, select that character string and click the <Search...> button. The selected character string is opened in the specified format.

## Assemble Search dialog box

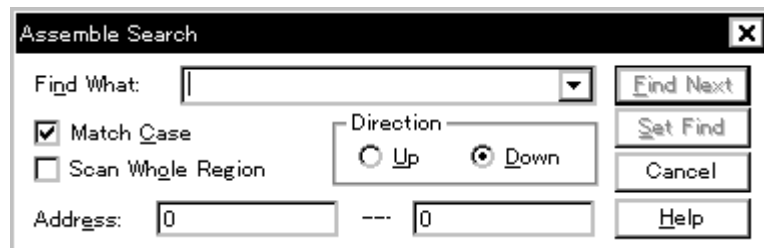
This dialog box is used to search the contents of Assemble window.

Successive character strings included in an input character string and disassembler character string are compared as one blank character.

By setting each item and then clicking the <Find Next> button, searching can be started.

By clicking the <Set Find> button, the direction buttons ("<<" and ">>") in the Assemble window can be used for the search.

Figure 7-25 Assemble Search Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

When Assemble window is the current window

- Click the <Search> button.  
(or press Ctrl+G).
- Select [View] -> [Search...]  
(or press Alt+V, and S in that order).

### Explanation of each area

The Assemble Search dialog box consists of the following areas:

- (1) Find What (search data specification area)
- (2) Match Case (search condition specification area)
- (3) Scan Whole Region (search condition specification area)
- (4) Direction (search direction specification area)

(5) Address (search range specification area)

(1) **Find What** (search data specification area)

Find What:

This area is used to specify the data to be searched.

In the default condition, the character string selected in the window that called this dialog box is displayed. As necessary, input data from the keyboard to change the character string.

Up to 256 character string can be specified.

By displaying the input history, the contents previously input can be reused. Up to 16 input histories can be recorded.

(2) **Match Case** (search condition specification area)

☒ Match Case

This area is used to specify whether uppercase and lowercase characters are distinguished, before searching is started.

Checked	Distinguished (default)
Not checked	Not distinguished

(3) **Scan Whole Region** (search condition specification area)

☐ Scan Whole Region

This area is used to specify whether the specified range is fully searched or not.

Checked	Searches the entire range.
Not checked	Searches the remaining part of the range (default)

(4) **Direction** (search direction specification area)

Direction  
☐ Up ☒ Down

This area is used to specify the direction of the search.

Up	Forward search. Searches data forward (upward on screen) from the current position of the cursor.
Down	Backward search (default). Searches data backward (downward on screen) from the current position of the cursor.

**(5) Address (search range specification area)**

Address:  ---

This area is used to specify the address to be searched.

An address can be also specified by a symbol or expression (refer to "Table 7-10 Specifying Symbols"). The default radix for inputting a numeric value is hexadecimal.

**Function buttons**

Find Next	Searches the specified data in accordance with a given condition. If the specified character string is found as a result of a search, it is highlighted. To continue searching, click this button again.
Set Find	Sets the specified condition as the search condition and closes this dialog box.
Stop (during search)	Stops searching.
Cancel	Closes this dialog box. This button is displayed as the <Stop> button while data is being searched.
Help	Opens the Help window.

## Memory window

This window is used to display and change the memory contents.

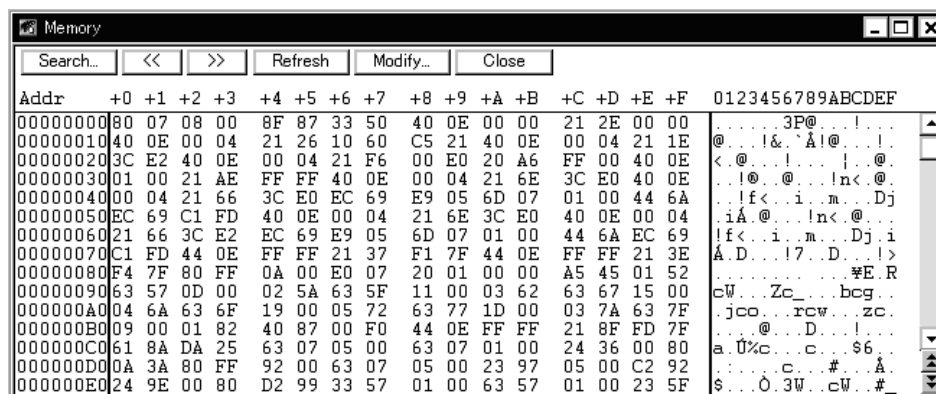
The display start position when the Memory window is opened is as follows:

First time	Display starts from the first address of the RAM sampling area, if the RAM sampling function is enabled. If it is disable, display starts from the first address of internal RAM area.
Second and subsequent times	Display starts from the address at which an active status window was closed (if an active status window has never been closed, display starts from the first display start position)

Even while the program is being executed, any 1 KB in the internal RAM area (the display range set in the Extended Option dialog box) can be displayed.

The Memory window has two statuses: Active Status and Static Status.

Figure 7-26 Memory Window



This section explains the following items:

- Opening
- Explanation of each area
- Functions often used (right-click menu)
- Function buttons
- Related operations
  - To change the memory contents
  - To change the display start position
  - To search in the Memory window
  - To display ASCII characters
  - To initialize the memory contents with a specified code

- To copy the memory contents
- To compare the memory contents
- To verify and check a memory write
- To change the memory contents during program execution (DMM function)

## Opening

This window can be opened as follows:

- Click the **Mem** button.
- Select [Browse] -> [Memory...] from the menu bar (or press Alt+B, and M in that order).
- Select [Jump] -> [Memory...] from the menu bar (or press Alt+J, and M in that order).

## Explanation of each area

The Memory window consists of the following areas:

- (1) Addr (address display area)
- (2) +0 +1 +2....(memory display area)
- (3) 0 1 2 3.... (ASCII display area)

### (1) Addr (address display area)

```
Addr
00FFE000
00FFE010
00FFE020
00FFE030
00FFE040
00FFE050
00FFE060
00FFE070
```

This area displays memory addresses.

### (2) +0 +1 +2....(memory display area)

```
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +
00 20 00 00 00 00 08 00 00 00 0
00 00 04 00 00 00 00 00 0C 08 0
02 00 00 00 40 00 00 80 00 00 0
02 02 00 00 60 00 02 01 00 00 0
00 00 00 00 00 00 00 00 00 08 0
00 00 00 00 00 00 00 00 00 00 0
```

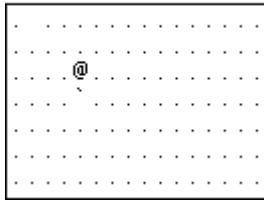
This area is used to display and change the memory contents.

Data into which the current memory contents are to be changed is displayed in red and is actually written to the target memory when the Enter key is pressed. The contents (displayed in red) can be canceled by the ESC key. Up to 256 bytes can be specified at one time.

This area is also used as a jump pointer of the jump function.

**(3) 0 1 2 3.... (ASCII display area)**

0123456789ABCDEF



This area is used to display and change the memory contents in ASCII characters.

This area is displayed when right-click menu -> [Ascii] is selected. Display of this area can be turned on/off by this menu item.

Data can be changed in this area in the same manner as in the memory display area.

This area is also used as a jump pointer of the jump function.

**Functions often used (right-click menu)**

In this window, functions that are often used can be selected from the menu that is displayed by clicking the right mouse button. In this way, the same function can be used more easily than with the normal procedure.

Move...	Moves the display position. Opens the Address move dialog box.
Bin	Displays binary numbers.
Oct	Displays octal numbers.
Dec	Displays decimal numbers.
Hex	Displays hexadecimal numbers (default).
Nibble	Displays in 4-bit units.
Byte	Displays in 8-bit units (default).
Half word	Displays in 16-bit units.
Word	Displays in 32-bit units.
Ascii	Turns on/off display of ASCII characters. Checked: Displayed Not checked: No display (default)

**Function buttons**

Search...	Opens the Memory Search dialog box and searches for character strings from the displayed memory contents, or memory contents. Selected data (a memory value) is displayed in the Memory Search dialog box as data to be searched. If the Memory Search dialog box is opened without data specified, specify data from the keyboard. The result of the search is highlighted in the Memory window.
-----------	---



<<	Searches the memory contents satisfying the search condition set in the Memory Search dialog box, forward (upward on screen) from the address at the cursor position. This button is displayed as the <Stop> button during a search.
>>	Searches the memory contents satisfying the search condition set in the Memory Search dialog box, backward (downward on screen) from the address at the cursor position. This button is displayed as the <Stop> button during a search.
Stop (during search)	Stops searching.
Refresh	Updates the contents of the window with the latest data.
Modify...	Opens the DMM dialog box.
Close	Closes this window.

## Related operations

### To change the memory contents

To change the memory contents, move the cursor to the memory display area or ASCII display area in the window, and write data directly from the keyboard. The amount of memory contents that can be changed at one time is 256 bytes or less. To set the change, press the return key. To cancel the change, press the ESC key.

### To change the display start position

Use the Address move dialog box, which can be opened by selecting right-click menu -> [Move...] in the Memory window.

### To search in the Memory window

Use the Memory Search dialog box, which can be opened by clicking the <Search...> button in the Memory window. If the character string to be searched exists in the window, select that character string and click the <Search...> button. The selected character string is opened in the specified format. Non-mapped, peripheral I/O register, and I/O protect areas are not searched.

### To display ASCII characters

The memory contents are displayed in ASCII characters when right-click menu -> [Ascii] is selected in the Memory window.

To turn off display of ASCII characters, select [Ascii] again to remove the check mark.

### To initialize the memory contents with a specified code

Initialize the memory contents with a specified code in the Memory Fill dialog box, which can be opened by selecting [Edit] -> [Memory] -> [Initialize] from the menu bar.

If verify check is enabled in the Extended Option dialog box, verify check is executed after initialization.

**To copy the memory contents**

Copy the memory contents in the Memory Copy dialog box, which can be opened by selecting [Edit] -> [Memory] -> [Copy] from the menu bar.

If verify check is enabled in the Extended Option dialog box, verify check is executed after copy.

**To compare the memory contents**

Compare the memory contents in the Memory Compare dialog box, which can be opened by selecting [Edit] -> [Memory] -> [Compare] from the menu bar.

If a difference is found as a result of comparison, the Memory Compare Result dialog box is opened and the details are displayed in this dialog box.

**To verify and check a memory write**

Use the Extended Option dialog box, which is opened by selecting [Option] -> [Extended Option] from the menu bar. To execute a verify check, check the [Verify] check box.

**To change the memory contents during program execution (DMM function)**

Set addresses and data for DMM (Dynamic Memory Modification) in the DMM dialog box, which can be opened by selecting [Edit] -> [Memory] -> [Modify] from the menu bar.

Nothing can be written to the internal ROM and internal flash memory.

## Memory Search dialog box

This dialog box is used to search the memory contents of the part of Memory window at which the cursor is located.

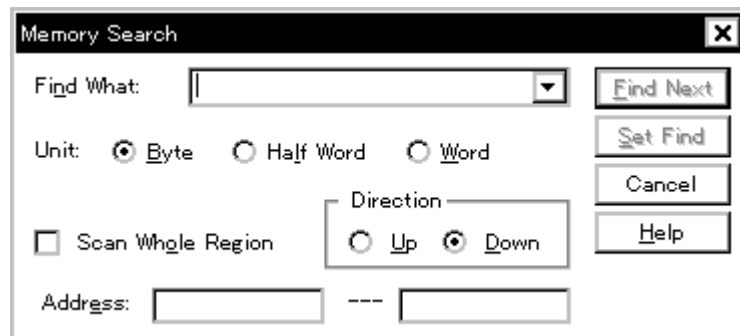
If the cursor is in the memory display area, the memory display area is searched for specified data, which is assumed to be a binary data string. If the cursor is in the ASCII display area, the ASCII display area is searched for specified data, which is assumed to be an ASCII character string.

By setting each item and then clicking the <Find Next> button, searching can be started.

By clicking the <Set Find> button, the direction buttons ("<<" and ">>") in the Memory window can be used for the search.

Non-mapped, peripheral I/O register, and I/O protect areas are not searched.

Figure 7-27 Memory Search Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

When Memory window is the current window

- Click the <Search> button.  
(or press Ctrl+G).
- Select [View] -> [Search...]  
(or press Alt+V, and S in that order).

## Explanation of each area

The Memory Search dialog box consists of the following areas:

- (1) Find What (search data specification area)
- (2) Unit (search condition specification area)
- (3) Scan Whole Region (search condition specification area)
- (4) Direction (search direction specification area)
- (5) Address (search range specification area)

### (1) Find What (search data specification area)

Find What:

This area is used to specify the data to be searched.

In the default condition, the character string selected in the window that called this dialog box is displayed. As necessary, input data from the keyboard to change the character string.

When searching the memory display area, up to 16 data items can be specified. Delimit each data with a "blank character". When searching the ASCII display area, data of up to 256 characters can be specified. A "blank character" in the data is treated as a blank character.

By displaying the input history, the contents previously input can be reused. Up to 16 input histories can be recorded.

### (2) Unit (search condition specification area)

Unit: ☒ Byte ☐ Half Word ☐ Word

This area is used to specify the number of bits of the data to be searched in the memory display area.

Byte	Searches the data as 8-bit data (default).
Half Word	Searches the data as 16-bit data.
Word	Searches the data as 32-bit data.

### (3) Scan Whole Region (search condition specification area)

☐ Scan Whole Region

This area is used to specify whether the specified range is fully searched or not.

Checked	Searches the entire range.
Not checked	Searches the remaining part of the range (default)

**(4) Direction (search direction specification area)**

Direction

☐ Up ☒ Down

This area is used to specify the direction of the search.

Up	Forward search. Searches data forward (upward on screen) from the current position of the cursor.
Down	Backward search (default). Searches data backward (downward on screen) from the current position of the cursor.

**(5) Address (search range specification area)**

Address:  ---

This area is used to specify the address to be searched.

An address can be also specified by a symbol or expression (refer to "Table 7-10 Specifying Symbols"). The default radix for inputting a numeric value is hexadecimal.

**Function buttons**

Find Next	Searches the specified data in accordance with a given condition. If the specified character string is found as a result of a search, it is highlighted. To continue searching, click this button again.
Set Find	Sets the specified condition as the search condition and closes this dialog box.
Stop (during search)	Stops searching.
Cancel	Closes this dialog box. This button is displayed as the <Stop> button while data is being searched.
Help	Opens the Help window.

## Memory Fill dialog box

This dialog box is used to initialize the memory contents in the Memory window with specified codes.

Figure 7-28 Memory Fill Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

- Select [Edit] -> [Memory] -> [Fill] from the menu bar.  
(or press Alt+E, M and F in that order).

### Explanation of each area

The Memory Fill dialog box consists of the following areas:

- (1) Address (address range specification area)
- (2) fill code (data specification area)

#### (1) Address (address range specification area)

Address  
From: 0 -- 0

This area is used to specify the memory address range whose contents are to be initialized.

An address can be also specified by a symbol or expression (refer to "Table 7-10 Specifying Symbols"). The default radix for inputting a numeric value is hexadecimal.

Input the first address to be initialized and then the last address to be initialized.

**(2) fill code (data specification area)**

fill code =>

This area is used to specify the data with which the memory contents are to be initialized.

Up to 16 binary data strings (byte data strings) can be specified.

Delimit each data with a "blank character".

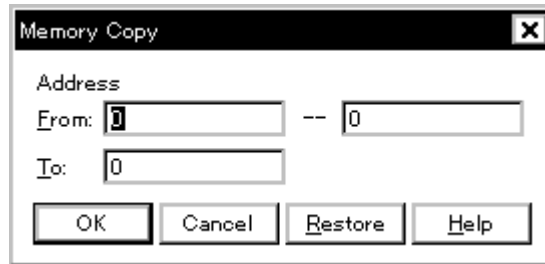
**Function buttons**

OK	Initializes the memory.
Stop (during initialization)	Stops initialization. While initialization is in progress, the <Cancel> button is replaced by the <Stop> button.
Cancel	Closes this dialog box. While memory contents are being initialized, this button is replaced by the <Stop> button.
Restore	Restores the input data to the original status.
Help	Opens the Help window.

## Memory Copy dialog box

This dialog box is used to copy the memory contents in the Memory window.

Figure 7-29 Memory Copy Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

- Select [E]dit -> [M]emory -> [C]opy from the menu bar.  
(or press Alt+E, M and C in that order).

### Explanation of each area

The Memory Copy dialog box consists of the following area:

- (1) Address (address range specification area)



**(1) Address (address range specification area)**

Address

From:  --   
 To:

This area is used to specify the copy source and copy destination addresses.

An address can be also specified by a symbol or expression (refer to "Table 7-10 Specifying Symbols"). The default radix for inputting a numeric value is hexadecimal.

From	Specify the address range of the copy source. Input the first address of the copy source and then the end address.
To	Specify the first address of the copy destination.

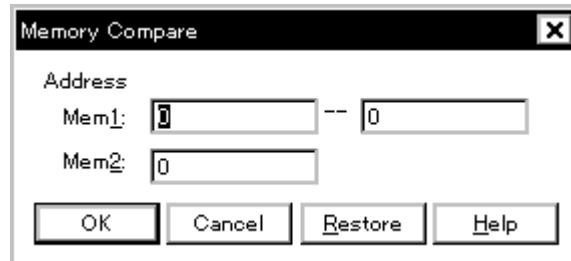
**Function buttons**

OK	Copies the memory.
Stop (during copying)	Stops copying. While copying is in progress, the <Cancel> button is replaced by the <Stop> button.
Cancel	Closes this dialog box. While copying is in progress, this button is replaced by the <Stop> button.
Restore	Restores the input data to the original status.
Help	Opens the Help window.

## Memory Compare dialog box

This dialog box is used to compare the memory contents in the Memory window.

Figure 7-30 Memory Compare Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

- Select [E]dit -> [M]emory -> [C]ompare from the menu bar.  
(or press Alt+E, M and P in that order).

### Explanation of each area

The Memory Compare dialog box consists of the following area:

- (1) Address (comparison range specification area)

**(1) Address (comparison range specification area)**

Address

Mem1:  --

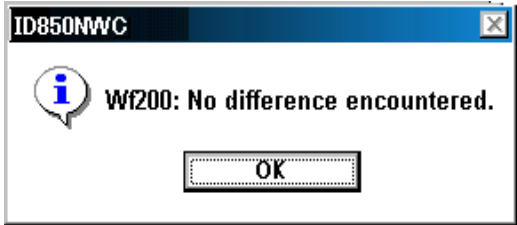
Mem2:

This area is used to specify the comparison source address and comparison destination address.

An address can be also specified by a symbol or expression (refer to "Table 7-10 Specifying Symbols"). The default radix for inputting a numeric value is hexadecimal.

Mem1	Specify the address range of the comparison source. Input the first address of the comparison source and then the end address.
Mem2	Specify the first address of the comparison destination.

**Function buttons**

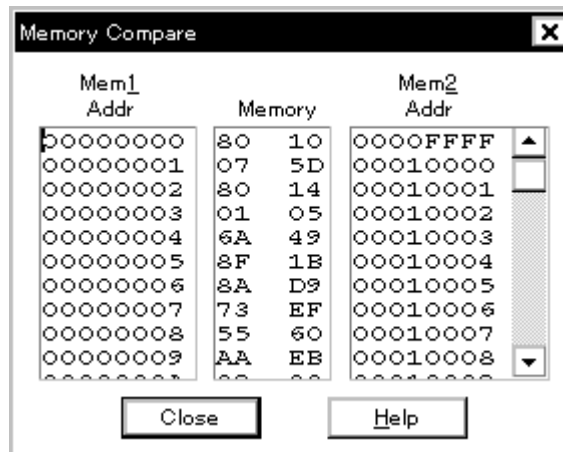
OK	<p>Compares the memory. If no difference is found as a result of comparison, a confirmation dialog box is displayed. If a difference is found, Memory Compare Result dialog box is opened.</p> <p style="text-align: center;">Figure 7-31 Confirmation Dialog Box</p>  <p>Clicking the &lt;OK&gt; button ends memory comparison.</p>
Stop (during comparison)	<p>Stops memory comparison. While comparison is in progress, the &lt;Cancel&gt; button is replaced by the &lt;Stop&gt; button.</p>
Cancel	<p>Closes this dialog box. While comparison is in progress, this button is replaced by the &lt;Stop&gt; button.</p>
Restore	Restores the input data to the original status.
Help	Opens the Help window.

## Memory Compare Result dialog box

Displays the result of comparing the memory.

This dialog box is displayed if any difference is found in the memory contents when the memory has been compared in Memory Compare dialog box.

Figure 7-32 Memory Compare Result Dialog Box



This section explains the following items:

- Explanation of each area
- Function buttons

### Explanation of each area

The Memory Compare Result dialog box consists of the following area:

- (1) Mem1, Mem2 (comparison result display area)

(1) **Mem1, Mem2** (comparison result display area)

Mem1 Addr	Memory	Mem2 Addr
00000000	80 10	0000FFFF
00000001	07 5D	00010000
00000002	80 14	00010001
00000003	01 05	00010002
00000004	6A 49	00010003
00000005	8F 1B	00010004
00000006	8A D9	00010005
00000007	73 EF	00010006
00000008	55 60	00010007
00000009	AA EB	00010008

This area displays the results of comparing the memory. Only differences that have been found as a result of comparison are displayed.

<b>Mem1 Addr</b>	Displays a comparison source address in which a difference has been found.
<b>Memory</b>	Displays the data in which a difference has been found. Left: Comparison source data Right: Comparison destination data
<b>Mem2 Addr</b>	Displays the comparison destination address at which a difference has been found.

**Function buttons**

Close	Closes this dialog box.
Help	Opens the Help window.

## DMM dialog box

This dialog box is used to set addresses and data for DMM (Dynamic Memory Modification).

DMM is used to rewrite the memory contents while the user program is being executed.

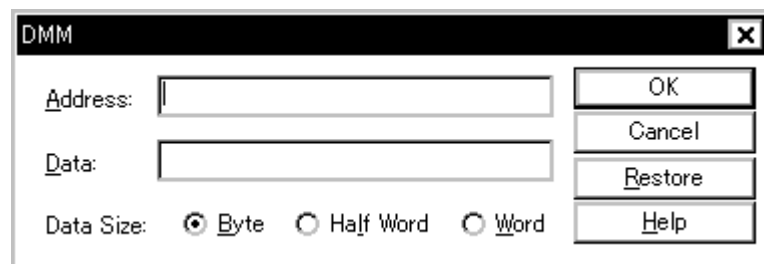
Because this function is implemented by software emulation, however, the user program under execution is temporarily stopped and then executed again.

**Caution1** Nothing can be written to the internal flash memory, during user program execution.

**Caution2** The following points must be noted when using the DMM function.

- If data is written to an address for which a software break is set, the program may not run correctly.
- If the program is stopped immediately before rewriting the memory because a break event is generated, the generated break event is invalid because program execution is resumed, regardless of the break event.
- Verify check is always invalid. Even if writing has failed, a verify error is not generated.

Figure 7-33 DMM Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

- Select [E]dit -> [M]emory -> [M]odify from the menu bar.  
(or press Alt+E, M and M in that order).
- Click the <Modify> button in the Memory window.

## Explanation of each area

The DMM dialog box consists of the following areas:

- (1) Address (address setting area)
- (2) Data (data setting area)
- (3) Data Size (data size setting area)

### (1) Address (address setting area)

Address:

This area is used to specify the address to which data is to be written.

An address can be also specified by a symbol or expression (refer to "Table 7-10 Specifying Symbols"). The default radix for inputting a numeric value is hexadecimal.

### (2) Data (data setting area)

Data:

This area is used to specify the data to be written to the memory address specified in [Address](#).

One binary data string can be specified at one time. Specify the data size in Data Size. Specify the data size in [Data Size](#). The default radix for inputting a numeric value is hexadecimal.

### (3) Data Size (data size setting area)

Data Size: ☒ [Byte](#) ☐ [Half Word](#) ☐ [Word](#)

This area is used to specify the size of the data to be written.

<a href="#">Byte</a>	Writes data specified in <a href="#">Data</a> as 8-bit data.
<a href="#">Half Word</a>	Writes data specified in <a href="#">Data</a> as 16-bit data.
<a href="#">Word</a>	Writes data specified in <a href="#">Data</a> as 32-bit data.

## Function buttons

OK	Writes the data.
Cancel	Closes this dialog box.
Restore	Restores the input data to the original status.
Help	Opens the Help window.

## Watch window

This window is used to display and change specified data.

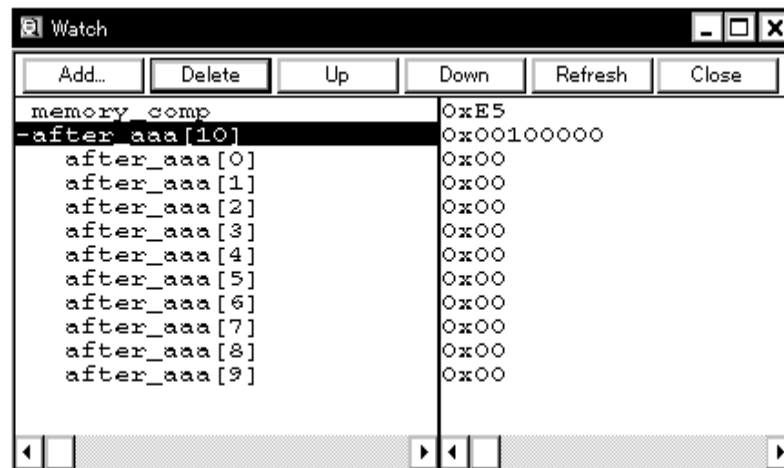
The result of updating and rewriting data in this window will be reflected in the Memory window.

This window can also display wide-ranging data (such as global variables and public symbols) in real time even during program execution, in the same way as the Memory window.

To set a breakpoint for read/write access to a variable can be performed by a single action from the menu that is displayed by clicking the right mouse button.

**Caution** If a local variable and a global variable exist with the same name, the local variable takes priority.

Figure 7-34 Watch Window



This section explains the following items:

- Opening
- Explanation of each area
- Functions often used (right-click menu)
- Function buttons
- Related operations
  - To add/delete displayed data (variable)
  - To change data on the selected line
  - To temporarily display a data (variable) value
  - To display/change a local variable value
  - To set a breakpoint for read/write access to a variable
  - To change displayed radix/size



## Opening

This window can be opened as follows:

- Click the **Wch** button.  
Select [Browse] -> [Watch] from the menu bar,  
(or press Alt+B, and W in that order).

Select data in the Source window or Assemble window,

- Select [View] -> [View Watch] from the menu bar,  
(or press Alt+ V, and I in that order),  
or click the <Watch> button.

In the Quick Watch dialog box or Add Watch dialog box dialog box,

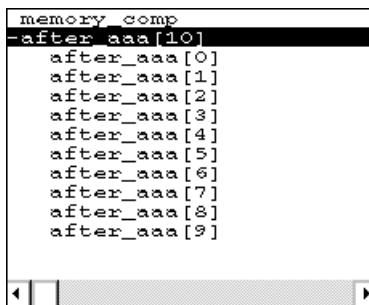
- Click the <Add> button.

## Explanation of each area

The Watch window consists of the following areas:

- Left field (symbol name display area)
- Right field (data value display/setting area)

### (1) Left field (symbol name display area)



This area is used to display variable names, symbol names and types, and tag names of structures or unions. Refer to "Name (symbol specification area)" in the Add Watch dialog box for details on the displayed contents.

'+' is prefixed to the displayed arrays, pointer variables, and structures or unions. These variables are expanded and displayed as follows when they are double-clicked:

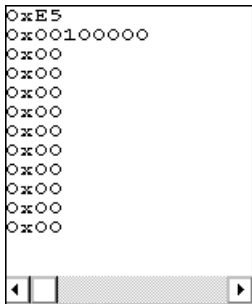
Array	All the elements of the variable are displayed in accordance with the type of the array variable.
Pointer variable	The data indicated by the pointer is displayed.
Structure/union	All the members of the structure/union are displayed in accordance with the type of the member variable. If a structure or union is defined in the structure or union, the structure name or union name of the internal structure or union is also displayed. The internal structure or union can be also expanded by using '+'. 

'+' of the variable that has been expanded and displayed changes to '-'. If the variable is double-clicked in this status, the expanded display is canceled. Expanded display can be also specified or canceled by selecting a variable with the arrow keys and then pressing the Enter key.

If an access breakpoint is set for a variable or a symbol in the Watch window, the symbol name display area is highlighted in gold.

**Reference** If an array has too many variables and takes too long to expand, a warning message is displayed.

## (2) Right field (data value display/setting area)



This area is used to display and change data values.

A value is updated when execution is stopped. To save a value, select [File] -> [Save As...] from the menu bar when the Save window is the current window. This area is blank if getting data has failed.

'?' is displayed for data that has been invalidated because of a change in the scope or optimized compiling. The contents of the change are written into the target memory when the Enter key is pressed. The previous value can be canceled by the ESC key.

The display format is as follows:

Table 7-16 Data Display Format (Watch-Related)

Display Data	Contents
Integer	Hexadecimal ( <b>0x</b> xxxx) Decimal (xxxx) Octal ( <b>0</b> xxxx) Binary ( <b>0b</b> xxxx)
Enumeration type	Member name
If scope is specified	Displayed in accordance with specified scope
Floating-point type	Single precision/double precision supported The input/display format is as follows: [ +   - ] inf [ +   - ] nan [ +   - ] integer e [ +   - ]exponent [ +   - ] integer.fraction [e [ +   - ]exponent

The size and radix of a data value can be specified on the [View] menu for each variable.

However, the display size cannot be selected if the display size is fixed, such as for variables in C or bit symbols or registers of the assembler.

If [View] -> [Adaptive] or [View] -> [Proper] from the menu bar is selected, the size and radix specified in the Debugger Option dialog box are used for display (refer to "To change displayed radix/size").

### Functions often used (right-click menu)

In this window, functions that are often used can be selected from the menu that is displayed by clicking the right mouse button. In this way, the same function can be used more easily than with the normal procedure.

Break when Access to this Variable	Creates a break event that occurs if the selected variable is accessed for read/write.
Break when Write to this Variable	Creates a break event that occurs if the selected variable is accessed for write.
Break when Read from this Variable	Creates a break event that occurs if the selected variable is accessed for read.
Clear	Deletes the break event corresponding to the selected variable.
Event?	Displays the event information of the line at the cursor position or the selected variable. If an event is set, Event dialog box is opened.
Change Watch...	Changes the selected data.
Delete Watch	Deletes the selected watch point from the window.
Bin	Displays the selected line in binary numbers.
Oct	Displays the selected line in octal numbers.
Dec	Displays the selected line in decimal numbers.
Hex	Displays the selected line in hexadecimal numbers.
String	Displays the selected line as a character string.
Proper	Displays the selected line as the default value of each variable. Symbols are displayed in accordance with the setting of Debugger Option dialog box (default).
Byte	Displays the selected line in 8-bit units.
Half Word	Displays the selected line in 16-bit units.
Word	Displays the selected line in 32-bit units.
Adaptive	Displays the selected line as the default value of each variable. Only this item is valid for a symbol in C language. Symbols in assembly language are displayed in accordance with the setting of Debugger Option dialog box (default).
Up	Moves the selected line one line up.
Down	Moves the selected line one line down.

## Function buttons

Add...	Opens the Add Watch dialog box. If data is specified and the <Add...> button is clicked in the Add Watch dialog box, the specified data is added to the Watch window.
Delete	Deletes the specified data from the Watch window. The DEL key performs the same operation.
Up	Moves the selected data one line up.
Down	Moves the selected data one line down.
Refresh	Updates the contents of the window with the latest data.
Close	Closes this window.

## Related operations

### To add/delete displayed data (variable)

#### Addition method (any of the following)

- Select a variable name or symbol name and click the <Watch> button in the Source window or Assemble window.
- Select a variable name or symbol name in the Source window or Assemble window, and drag it to the Watch window.
- Specify a variable name or symbol name and click the <Add> button in the Quick Watch dialog box or Add Watch dialog box.
- Select a variable name or symbol name and select [View] menu -> [Add Watch...] in the Source window or Assemble window.

#### Deletion method

- Click a variable name or symbol name (two or more variable or symbol names can be selected using the Shift or Ctrl key), and click the <Delete> button. A line with an expanded hierarchy, such as the elements of an array, and members of structures and unions cannot be changed.

### To change data on the selected line

To change data (variables), select a line to be changed in the Watch window and select right-click menu -> [Change Watch...]. The Change Watch dialog box will be opened to change the data.

A line with an expanded hierarchy, such as the elements of an array, and members of structures and unions cannot be deleted.

### To temporarily display a data (variable) value

A data (variable) value can be temporarily displayed in the following two ways:

#### To display using the balloon watch function

Select data (a variable name) in the Source window or Assemble window. Then place the mouse cursor on the selected character string. The value of the selected data will be displayed.

#### To display in the Quick Watch dialog box

Select data (a variable name) in the Source window or Assemble window. Then click the <Quick...>

button. The value of the selected data will be displayed in the Quick Watch dialog box.

**To display/change a local variable value**

A local variable value can be checked or changed in the Local Variable window, which is opened when the **Loc** button is clicked.

In this window, the local variable in the current function is automatically displayed and the displayed variable cannot be added or deleted.

To change a local variable value, move the cursor to the local variable value display/changing area on the window and write a value in this area directly from the keyboard.

To determine the displayed contents, use the Enter key. To cancel, use the ESC key.

**To set a breakpoint for read/write access to a variable**

Select a variable in the Watch window and select [Break when Access to this Variable], [Break when Write to this Variable] or [Break when Read from this Variable] from the right-click menu. In this way, a hardware break event will be created in the valid status in accordance with the selected item.

Whether the event has been correctly created can be confirmed on the Event Manager.

**To change displayed radix/size****Changing each data (variable) at any time**

Select a data (variable) name to be changed in the Watch window or Quick Watch dialog box, and select the radix and size from the right-click menu.

**Specifying default condition**

Adaptive (when display size is specified) or Proper (when display radix is specified) for watching a symbol that specifies a default value can be set as the default assumption.

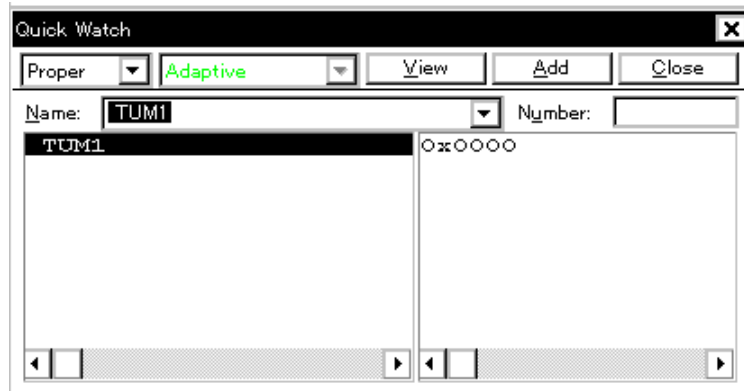
This is done in the Watch Default area of the Debugger Option dialog box, which is opened when [Option] -> [Debugger Option...] is selected from the menu bar.

The size specified under Size is displayed when Adaptive is specified, and the radix specified under Radix is displayed when Proper is specified.

## Quick Watch dialog box

This dialog box is used to temporarily display or change specified data.

Figure 7-35 Quick Watch Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

- Select [View] -> [Quick Watch...]  
(or press Alt+V, and Q in that order).
- Click the <Quick...> button in Source window.  
(or press Ctrl+W).
- Click the <Quick...> button in Assemble window.  
(or press Ctrl+W).

### Explanation of each area

The Quick Watch dialog box consists of the following areas:

- (1) Name (symbol specification area)
- (2) Symbol name display area
- (3) Data value display/setting area
- (4) Display radix selection area (upper left field)
- (5) Display size selection area (second upper left field)
- (6) Number (display number specification area)

**(1) Name (symbol specification area)**

Name:

This area is used to specify the data to be displayed.

In the default condition, the character string selected in the window that called this dialog box is displayed. As necessary, input data from the keyboard to change the character string.

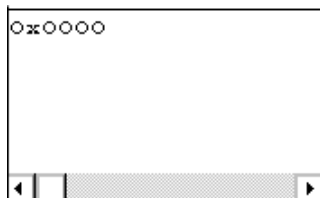
By displaying the input history, the contents previously input can be reused. Up to 16 input histories can be recorded.

If the contents of **Name** have been changed, the data specified in **Name** can be displayed in the field below by clicking the <View> button.

**(2) Symbol name display area**

This area is used to display the variable name and the symbol name.

The displayed contents are the same as those of Name (symbol specification area) in the Add Watch dialog box.

**(3) Data value display/setting area**

This area is used to display and change data values. The displayed contents and display method are the same as those of Watch window.

The displayed contents are the same as those of Right field (data value display/setting area) in the Watch window.

**(4) Display radix selection area (upper left field)**

This area is used to select the display radix.

The display radix of the data name selected in the symbol name display area will be changed.

<b>Proper</b>	<b>Variable</b> The default value of each variable is displayed. <b>Symbol</b> Displays data with the radix set in Debugger Option dialog box.
<b>Hex</b>	Displays data in hexadecimal numbers ( <b>0xxxxx</b> )

Dec	Displays data in decimal numbers (xxxx)
Oct	Displays data in octal numbers (0xxxx)
Bin	Displays data in binary numbers (0bxxxx)
String	Displays character strings.

(5) **Display size selection area (second upper left field)**



This area is used to select the display size.

The display size of the data selected in the symbol name display area will be changed.

If the display size is fixed, such as when a variable in C or register is to be displayed, it cannot be changed.

Adaptive	<b>Variable</b> The default value of each variable is displayed. <b>Symbol</b> Displays data with the size set in Debugger Option dialog box.
Byte	Displays data in 8 bits.
Half Word	Displays data in 16 bits.
Word	Displays data in 32 bits.

(6) **Number (display number specification area)**

Number:

This area is used to specify the number of data to be displayed.

Specify blank or a number of 1 to 256.

If this area is blank, data is displayed as a simple variable.

If a number of 1 or more is specified, data is displayed as an array variable.

If an array variable is displayed, '+' is prefixed to the data. By double-clicking this '+', all the elements of the data are expanded and displayed in accordance with the type of the data ('-' is prefixed to the expanded data. If this '-' is double-clicked, the expanded display is canceled).

If the number of data to be displayed is fixed, such as when a variable in C or register is to be displayed, the specified number of data is invalid.

### Function buttons

View	Displays the data specified in <b>Name</b> in the field below.
Add	Displays the data specified in <b>Name</b> to the Watch window.
Close	Closes this dialog box. Data that has not actually been written to the target will be canceled.

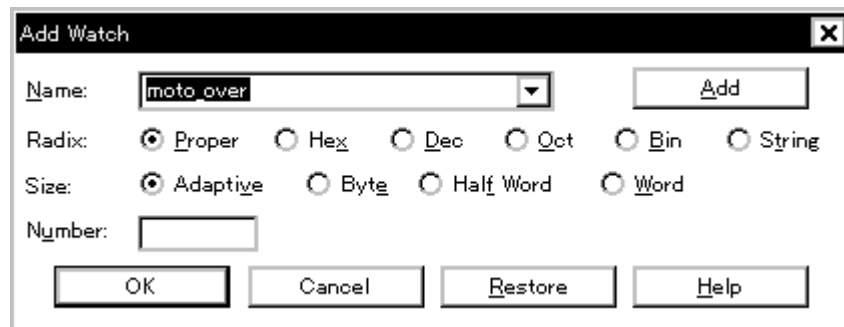


## Add Watch dialog box

This dialog box is used to register data to be displayed in the Watch window.

Multiple data with the same symbol name can be registered.

Figure 7-36 Add Watch Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

- Select [V]iew -> [A]dd W[atch...] from the menu bar (or press Alt+V, and W in that order).
- Click the <Add> button in Watch window.

### Explanation of each area

The Add Watch dialog box consists of the following areas:

- (1) Name (symbol specification area)
- (2) Radix (display radix selection area)
- (3) Size (display size specification area)
- (4) Number (display number specification area)

**(1) Name (symbol specification area)**

Name:

This area is used to specify data to be added to the Watch window.

In the default condition, the character string selected in the window that called this dialog box is displayed. As necessary, input data from the keyboard to change the character string. This area is blank if no character string is selected.

By displaying the input history, the contents previously input can be reused. Up to 16 input histories can be recorded.

The format in which a variable can be input and how a variable is handled when a scope is specified are as follows:

- Variable Name

Variable Name

Variable expression[Constant value | Variable Name]

Variable expression. Member name

Variable expression -> Member name

\*Variable expression

&Variable expression

Variable Name

Elements of array

Entity members of structure/union

Members of structure/union indicated by pointer

Value of pointer variable

Address where variable is located

- Register name
- IOR name, IOR bit name
- Label and address of immediate value
- Register name.bit
- Label name.bit , address of immediate value.bit
- Specification of scope

Table 7-17 How a Variable Is Handled When a Scope Is Specified

Scope Specification	Program name	File name	Function name	Variable Name
prog\$file#func#var	prog	file	func	var
prog\$file#var	prog	file	Global	var
prog\$func#var	prog	Global	func	var
prog\$var	prog	Global	Global	var
file#func#var	Current	file	func	var
file#var	Current	file	Global	var
func#var	Current	Current	func	var
var	Current	Current	Current	var

**(2) Radix (display radix selection area)**

Radix: ☒ Proper ☐ Hex ☐ Dec ☐ Oct ☐ Bin ☐ String

This area is used to change the radix of the data to be displayed.

Proper	<b>Variable</b> The default value of each variable is displayed. <b>Symbol</b> Displays data with the radix set in Debugger Option dialog box.
Hex	Displays data in hexadecimal numbers ( <b>0xxxxx</b> )
Dec	Displays data in decimal numbers ( <b>xxxx</b> )
Oct	Displays data in octal numbers ( <b>0xxxxx</b> )
Bin	Displays data in binary numbers ( <b>0bxxxxx</b> )
String	Displays character strings.

**(3) Size (display size specification area)**

Size: ☒ Adaptive ☐ Byte ☐ Half Word ☐ Word

This area is used to change the size of data to be displayed.

If the display size is fixed, such as when a variable in C or a bit symbol or register of the assembler is to be displayed, it cannot be changed.

Adaptive	<b>Variable</b> The default value of each variable is displayed. <b>Symbol</b> Displays data with the size set in Debugger Option dialog box.
Byte	Displays data in 8 bits.
Half Word	Displays data in 16 bits.
Word	Displays data in 32 bits.

**(4) Number (display number specification area)**

Number:

This area is used to specify the number of data to be displayed.

Specify blank or a number of 1 to 256.

If this area is blank, data is displayed in Watch window as a simple variable.

If a number of 1 or more is specified, data is displayed in the Watch window as an array variable.

The maximum number of lines that can be displayed in the Watch window is 10,000.

If an array variable is displayed, '+' is prefixed to the data. By double-clicking this '+', all the elements of the data are expanded and displayed in accordance with the type of the data ('-' is prefixed to the expanded data. If this '-' is double-clicked, the expanded display is canceled).

If the number of data to be displayed is fixed, such as when a variable in C or register is to be displayed, the number of data is invalid in the Watch window even if it has been changed.

**Function buttons**

---

Add	Adds the specified data to the Watch window. The dialog box remains open.
OK	Adds the specified data to the Watch window. Closes this dialog box.
Cancel	Closes this dialog box.
Restore	Restores the input data to the original status.
Help	Opens the Help window.

## Change Watch dialog box

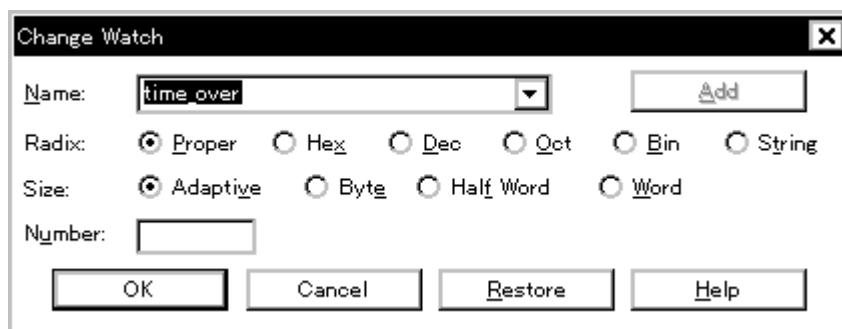
This window is used to change the data on a line selected in the Watch window.

A line with an open hierarchy, such as the elements of an array, and members of structures and unions cannot be changed.

When watch data is changed, the contents of the selected line are replaced with the new data.

The symbol name can be changed even if it results in duplication of a name already in use with existing data.

Figure 7-37 Change Watch Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

When the Watch window is open or a line is selected

- Select [V]iew -> [Change Watch...] from the menu bar (or press Alt+V, and G in that order).

### Explanation of each area

The Change Watch dialog box consists of the following areas:

- (1) Name (symbol specification area)
- (2) Radix (display radix selection area)
- (3) Size (display size specification area)
- (4) Number (display number specification area)

**(1) Name (symbol specification area)**

Name:

This area is used to change a symbol name.

By displaying the input history, the contents previously input can be reused. Up to 16 input histories can be recorded.

The format in which a symbol name can be input is the same as that of Name (symbol specification area) in the Add Watch dialog box.

**(2) Radix (display radix selection area)**

Radix: ☒ Proper ☐ Hex ☐ Dec ☐ Oct ☐ Bin ☐ String

This area is used to change the radix of data to be displayed.

Proper	<b>Variable</b> The default value of each variable is displayed. <b>Symbol</b> Displays data with the radix set in Debugger Option dialog box.
Hex	Displays data in hexadecimal numbers ( <b>0</b> xxxx)
Dec	Displays data in decimal numbers (xxxx)
Oct	Displays data in octal numbers ( <b>0</b> xxxx)
Bin	Displays data in binary numbers ( <b>0b</b> xxxx)
String	Displays character strings.

**(3) Size (display size specification area)**

Size: ☒ Adaptive ☐ Byte ☐ Half Word ☐ Word

Displays and modifies variables.

If the display size is fixed, such as when a variable in C or register is to be displayed, it cannot be changed.

Adaptive	<b>Variable</b> The default value of each variable is displayed. <b>Symbol</b> Displays data with the size set in Debugger Option dialog box.
Byte	Displays data in 8 bits.
Half Word	Displays data in 16 bits.
Word	Displays data in 32 bits.

**(4) Number (display number specification area)**Number: 

This area is used to specify the number of data to be displayed.

Specify blank or a number of 1 to 256.

If this area is blank, data is displayed in Watch window as a simple variable.

If a number of 1 or more is specified, data is displayed in the Watch window as an array variable.

The maximum number of lines that can be displayed in the Watch window is 10,000.

If an array variable is displayed, '+' is prefixed to the data. By double-clicking this '+', all the elements of the data are expanded and displayed in accordance with the type of the data ('-' is prefixed to the expanded data. If this '-' is double-clicked, the expanded display is canceled).

If the number of data to be displayed is fixed, such as when a variable in C or register is to be displayed, the number of data is invalid in the Watch window even if it has been changed.

**Function buttons**

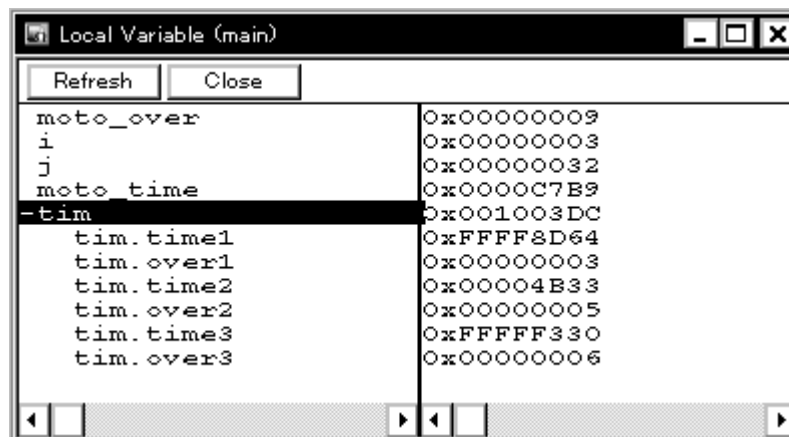
Add	Cannot be selected.
OK	Replaces the registration contents currently selected with the specified data, and then closes this dialog box. Closes this dialog box.
Cancel	Closes this dialog box.
Restore	Restores the input data to the original status.
Help	Opens the Help window.

## Local Variable window

This window is used to display and change the local variable in the current function.

This window automatically displays the local variable in the current function. It is linked with the jump function of the Stack Trace window, and displays the local variable in the function jumped when jumping to the Source window. Variables cannot be added or deleted.

Figure 7-38 Local Variable Window



This section explains the following items:

- Opening
- Explanation of each area
- Functions often used (right-click menu)
- Function buttons

### Opening

This window can be opened as follows:

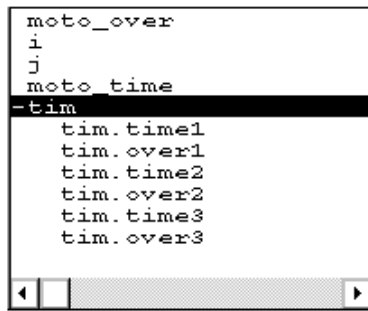
- Click the **Loc** button.
- Select [Browse] -> [Local Variable] from the menu bar, (or press Alt,+B, and L in that order).

### Explanation of each area

The Local Variable window consists of the following areas:

- (1) Left field (local variable name display area)
- (2) Right field (local variable value display/setting area)



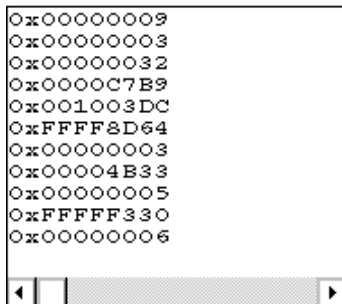
**(1) Left field (local variable name display area)**

This area displays local variable name.

Auto, internal Static, and Register variables can be displayed.

The display format is the same as that of the Watch window.

This area cannot be edited directly.

**(2) Right field (local variable value display/setting area)**

This area displays local variable values.

The display format of the variable value is the same as that of the Watch window.

The display radix of variables can be switched by the menu selection in accordance with each variable.

The new value is written into the target memory when the Enter key is pressed. The previous value can be canceled by the ESC key.

**Functions often used (right-click menu)**

In this window, functions that are often used can be selected from the menu that is displayed by clicking the right mouse button. In this way, the same function can be used more easily than with the normal procedure.

Add Watch	Registers a selected character string to the Watch window. Opens the Add Watch dialog box.
Bin	Displays the selected line in binary numbers.
Oct	Displays the selected line in octal numbers.
Dec	Displays the selected line in decimal numbers.
Hex	Displays the selected line in hexadecimal numbers.

String	Displays the selected line as a character string.
Proper	Displays the selected line as the default value of each variable. Symbols are displayed in accordance with the setting of Debugger Option dialog box (default).

---

**Function buttons**

---

Refresh	Updates the contents of the window with the latest data.
Close	Closes this window.

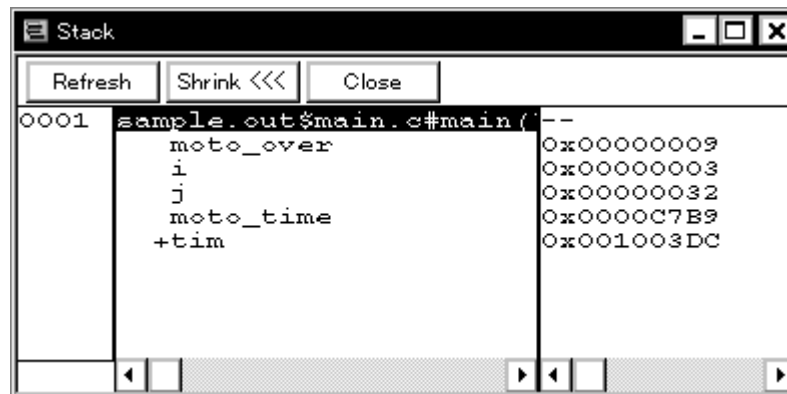
## Stack Trace window

This window is used to display or change the current stack contents of the user program.

From this window, you can also jump to the Source window, Assemble window, and Memory window corresponding to the stack contents.

**Caution** The stack trace display function may not operate correctly if there is a function that does not create a stack frame. In addition, [ERROR] may be displayed during prologue or epilogue processing of a function.

Figure 7-39 Stack Trace Window



This section explains the following items:

- Opening
- Explanation of each area
- Functions often used (right-click menu)
- Function buttons

### Opening

This window can be opened as follows:

- Click the **Stk** button.
- Select [Browse] -> [Stack Trace] from the menu bar, (or press Alt+B , and K in that order).

## Explanation of each area

The Stack Trace window consists of the following areas:

- (1) Left field (stack frame number display area)
- (2) Center field (stack frame contents display area)
- (3) Right field (stack contents display/setting area)

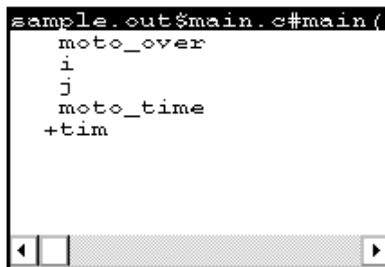
### (1) Left field (stack frame number display area)



This area assigns numbers to and displays the stack contents.

A stack frame number is a natural number starting from 1. The shallower the nesting of the stack, the higher the number. This means that a function having stack number one higher than that of a certain function is the function that calls the certain function.

### (2) Center field (stack frame contents display area)



This area displays the stack frame contents.

It displays function names or local variable names. Note, however, that this area cannot be directly edited.

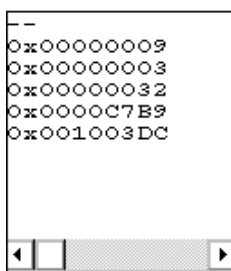
If the stack contents consist of a function, they are displayed as follows:

**program name\$file name#function name (argument list) #line number**

If this line is double-clicked, the operation will be the same as jumping to the Source window of the jump function (i.e., the local variable in the function to which execution has jumped will be displayed in the Local Variable window).

If the function has a local variable, the local variable will be displayed on the next and subsequent lines.

If the stack contents consist of a local variable, its type and name are displayed. These are displayed in the same manner as in the Watch window. Note that the internal Static and Register variables are not displayed.

**(3) Right field (stack contents display/setting area)**

This area is used to display or change the stack contents.

'-' or a local variable value is displayed. If the stack contents are a function '-' is displayed and the function cannot be changed.

If the stack contents are a local variable, the variable value is displayed in the same manner as in the Watch window.

The display radix of variables can be switched by the menu selection in accordance with each variable.

A variable value can be changed. The new value is written into the target memory when the Enter key is pressed. The previous value can be canceled by the ESC key.

This area is used as the jump source area of the jump function.

**Functions often used (right-click menu)**

In this window, functions that are often used can be selected from the menu that is displayed by clicking the right mouse button. In this way, the same function can be used more easily than with the normal procedure.

Bin	Displays the selected line in binary numbers.
Oct	Displays the selected line in octal numbers.
Dec	Displays the selected line in decimal numbers.
Hex	Displays the selected line in hexadecimal numbers.
String	Displays the selected line as a character string.
Proper	Displays the selected line as the default value of each variable. Symbols are displayed in accordance with the setting of Debugger Option dialog box (default).
Source Text	Displays the corresponding source text and source line from the jump destination address specified by the data value at the cursor position (the local variable in the function to which you have jumped is displayed in the Local Variable window). If no line information exists at the jump destination address, however, you cannot jump. Opens the Source window. If an active Source window is open, that window is displayed in the forefront (so that it can be manipulated).

<b>Assemble</b>	Disassembles and displays starting from the jump destination address specified by the data value at the cursor position. Opens the Assemble window. If an active Assemble window is open, that window is displayed in the forefront (so that it can be manipulated).
<b>Memory</b>	Displays the memory contents starting from the jump destination address specified by the data value at the cursor position. Opens the Memory window. If an active Memory window is open, that window is displayed in the forefront (so that it can be manipulated).

---

**Function buttons**

---

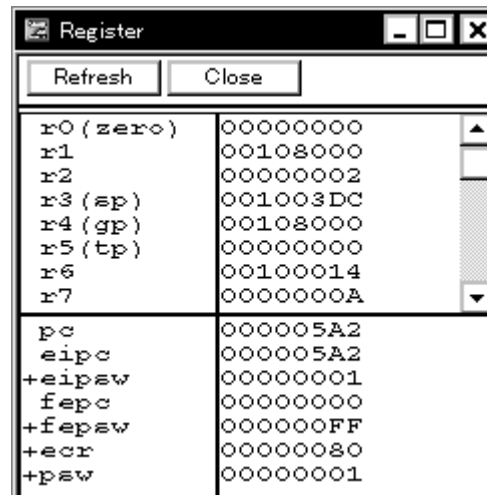
Refresh	Updates the contents of the window with the latest data.
Shrink<<<	Collapses the local variable list of the selected function.
Expand>>>	Displays the local variable list of the selected function.
Close	Closes this window.

## Register window

This window is used to display and change the contents of registers (program registers and system registers).

The register to be displayed in this window can be selected in the Register Select dialog box.

Figure 7-40 Register Window



This section explains the following items:

- Opening
- Explanation of each area
- Functions often used (right-click menu)
- Function buttons
- Related operations
  - To change a register value
  - To select the register to be displayed

### Opening

This window can be opened as follows:

- Click the **Reg** button.
- Select [Browse] -> [Register] from the menu bar, (or press Alt+B, and R in that order).

## Explanation of each area

The Register window consists of the following areas:

- (1) Upper left field (program register display area)
- (2) Lower left field (system register display area)

### (1) Upper left field (program register display area)

r0 (zero)	00000000
r1	00108000
r2	00000002
r3 (sp)	001003DC
r4 (gp)	00108000
r5 (tp)	00000000
r6	00100014
r7	0000000A

This area is used to display or change the contents of program registers.

The new value is written into the target memory when the Enter key is pressed. The previous value can be canceled by the ESC key.

This area is also used as a jump pointer of the jump function.

### (2) Lower left field (system register display area)

pc	000005A2
eip	000005A2
+eipsw	00000001
fep	00000000
+fepsw	000000FF
+ecr	00000080
+psw	00000001

This area is used to display or change the contents of system registers.

The new value is written into the target memory when the Enter key is pressed. The previous value can be canceled by the ESC key.

If the name of a register with '+' prefixed is double-clicked, the names and values of its flags are displayed, and '+' changes to '-'. This '+' can be also changed to '-' by selecting a register name with the arrow keys and pressing the Enter key.

This area is also used as a jump pointer of the jump function.

## Functions often used (right-click menu)

In this window, functions that are often used can be selected from the menu that is displayed by clicking the right mouse button. In this way, the same function can be used more easily than with the normal procedure.

Add Watch	Registers a selected character string to the Watch window. Opens the Add Watch dialog box.
Bin	Displays the selected line in binary numbers.
Oct	Displays the selected line in octal numbers.



Dec	Displays the selected line in decimal numbers.
Hex	Displays the selected line in hexadecimal numbers (default).
Pick Up	Displays only the registers selected in Register Select dialog box. Checked: Displays only the selected register. Not checked: Displays all the registers (default).
Select...	Opens the Register Select dialog box.

### Function buttons

Refresh	Updates the contents of the window with the latest data.
Close	Closes this window.

### Related operations

#### To change a register value

To change the value of a register, move the cursor to the system register display area or program register display area in the window, and directly write a value from the keyboard.

Press the return key to set the new value. To cancel the change, press the ESC key.

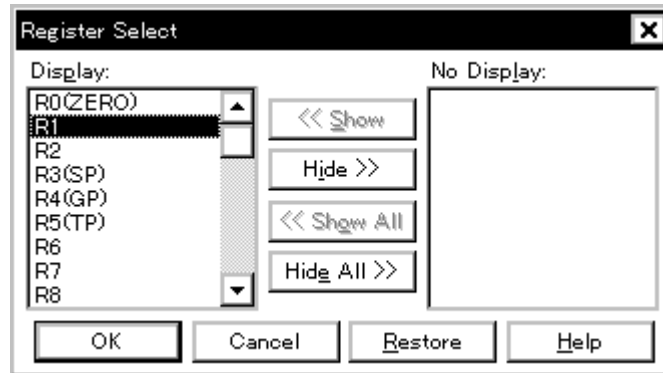
#### To select the register to be displayed

Use the Register Select dialog box, which can be opened by selecting right-click menu -> [Select...] in the Register window.

## Register Select dialog box

This dialog box is used to select registers that are not displayed in the Register window.

Figure 7-41 Register Select Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

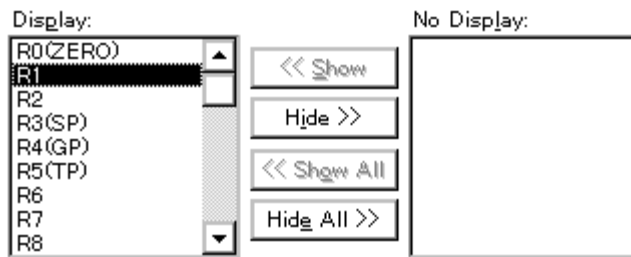
When Register window is the current window,

- Select [View] -> [Select...] from the menu bar (or press Alt+V, and E in that order).

### Explanation of each area

The Register Select dialog box consists of the following area:

- (1) Display, No Display (display register selection area)

**(1) Display, No Display (display register selection area)**

This area is used to select registers that are displayed in the Register window, and those that are not.

Display	Registers displayed in the Register window.
No Display	Registers not displayed in the Register window.

To display or not display a register in the Register window, select a register name from these lists, and click the <<<Show> or <Hide>>> button. To display all the registers, click the <<<Show All> button. To not display any of the registers, click the <<<Hide All> button.

The function of each button is as follows:

<< Show	Moves the register selected from the No Display list to Display.
Hide >>	Moves the register selected from the Display list to No Display.
<< Show All	Moves all registers to Display.
Hide All >>	Moves all registers to No Display.

When selecting a register from the lists, two or more registers can be selected by clicking any of the above buttons while holding down the Ctrl or Shift key.

**Function buttons**

OK	Reflects the selection in this dialog box in Register window and closes this dialog box.
Cancel	Cancels the changes and closes this dialog box.
Restore	Restores the original status.
Help	Opens the Help window.

## IOR window

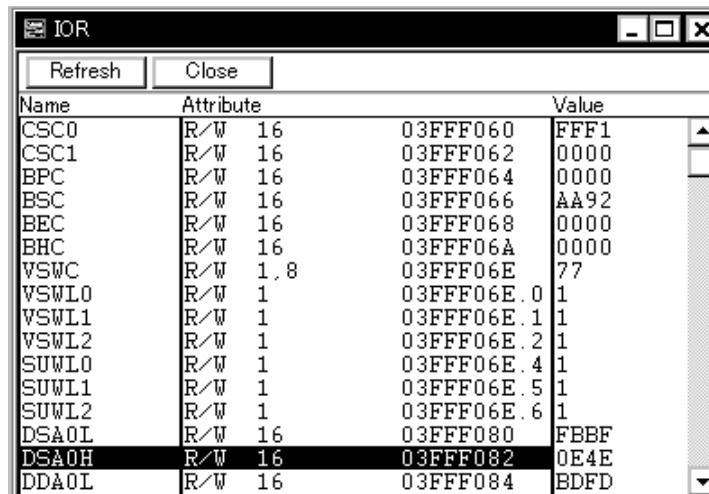
This window is used to display and change the contents of peripheral I/O registers and the I/O ports that have been registered in the Add I/O Port dialog box.

Note, however, that the values of read-only peripheral I/O registers and I/O ports cannot be changed. In addition, the peripheral I/O registers and I/O ports that cause the device to operate when they are read are read-protected and therefore cannot be read.

To read these registers, select a register, and select and execute [Compulsion Read] from the right-click menu.

**Reference** If the device supports programmable I/O registers and a programmable I/O area has been set in the Configuration dialog box, the programmable I/O registers are also displayed. If the value of an I/O port address is defined, the I/O port name is displayed in light color.

Figure 7-42 IOR Window



The screenshot shows a window titled "IOR" with "Refresh" and "Close" buttons. It contains a table with the following data:

Name	Attribute	Value
CSC0	R/W 16	03FFF060 FFF1
CSC1	R/W 16	03FFF062 0000
BPC	R/W 16	03FFF064 0000
BSC	R/W 16	03FFF066 AA92
BEC	R/W 16	03FFF068 0000
BHC	R/W 16	03FFF06A 0000
VSWC	R/W 1, 8	03FFF06E 77
VSWL0	R/W 1	03FFF06E.0 1
VSWL1	R/W 1	03FFF06E.1 1
VSWL2	R/W 1	03FFF06E.2 1
SUWL0	R/W 1	03FFF06E.4 1
SUWL1	R/W 1	03FFF06E.5 1
SUWL2	R/W 1	03FFF06E.6 1
DSA0L	R/W 16	03FFF080 FBBF
DSA0H	R/W 16	03FFF082 0E4E
DDA0L	R/W 16	03FFF084 BDFD

This section explains the following items:

- Opening
- Explanation of each area
- Functions often used (right-click menu)
- Function buttons
- Related operations
  - To change a peripheral I/O register value
  - To select a peripheral I/O register or I/O port to be displayed
  - To change the display sequence
  - To change the display radix

- To add, change the contents of, and delete a user-defined I/O port
- To change the display start position

## Opening

This window box can be opened as follows:

- Click the **IOR** button.
- Select **[Browse] -> [I/O Register]** from the menu bar (or press Alt, +B, and G in that order).

## Explanation of each area

The IOR window consists of the following areas:

- (1) Name (peripheral I/O register name display area)
- (2) Attribute (attribute display area)
- (3) Value (peripheral I/O register contents display area)

### (1) Name (peripheral I/O register name display area)

Name
CSC0
CSC1
BPC
BSC
BEC
BHC
VSWC
VSWL0
VSWL1
VSWL2
SUWL0
SUWL1
SUWL2
DSA0L
DSA0H
DDA0L

This area displays the names of peripheral I/O registers and I/O ports.

When the V850E is used, it also displays the programmable peripheral I/O registers and expansion peripheral I/O registers.

If the value of an I/O port address is not defined, the I/O port name displayed in light color.

### (2) Attribute (attribute display area)

Attribute
R/W 16 03FFF060
R/W 16 03FFF062
R/W 16 03FFF064
R/W 16 03FFF066
R/W 16 03FFF068
R/W 16 03FFF06A
R/W 1, 8 03FFF06E
R/W 1 03FFF06E.0
R/W 1 03FFF06E.1
R/W 1 03FFF06E.2
R/W 1 03FFF06E.4
R/W 1 03FFF06E.5
R/W 1 03FFF06E.6
R/W 16 03FFF080
R/W 16 03FFF082
R/W 16 03FFF084

This area displays the read/write attributes, access types, and absolute addresses of the peripheral I/O registers and I/O ports. When the bit peripheral I/O register is displayed, bit-offset value is also displayed.

It can be specified whether this area is displayed or not, by selecting [View] -> [Attribute] from the menu bar. As an attribute, the following contents are displayed.

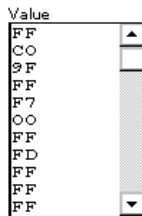
#### Read/Write Attribute

R	Read only
W	Write only
R/W	Read/write
*	Register that is read via an emulation register to prevent the device from operating when this register is read. To read this attribute directly from a peripheral I/O register, execute [View] -> [Compulsion Read] from the menu bar. Even a write-only peripheral I/O register can also be read via an emulation register. However, some devices do not support this function.

#### Access Type

1	Can be accessed in bit units.
8	Can be accessed in byte units.
16	Can be accessed in halfword units.
32	Can be accessed in word units.

### (3) Value (peripheral I/O register contents display area)



This area is used to display and change the contents of a peripheral I/O register and I/O port.

The contents are displayed differently as follows, depending on the attribute:

<b>Black Display</b>	Read only or read/write
--	Write only
**	Value changes if read

The new value is written into the target memory when the Enter key is pressed. The previous value can be canceled by the ESC key.

The value of read-protected peripheral I/O registers and I/O ports can be read by selecting right-click menu -> [Compulsion Read].

### Functions often used (right-click menu)

In this window, functions that are often used can be selected from the menu that is displayed by clicking the right mouse button. In this way, the same function can be used more easily than with the normal procedure.

Move...	Moves the display position. Opens the Address move dialog box.
Add Watch...	Registers a selected character string to the Watch window. Opens the Add Watch dialog box.
Add I/O Port...	Adds user-defined I/O ports. Opens the Add I/O Port dialog box.
Bin	Displays the selected line in binary numbers.
Oct	Displays the selected line in octal numbers.
Dec	Displays the selected line in decimal numbers.
Hex	Displays the selected line in hexadecimal numbers (default).
Sort By Name	Displays in alphabetical order.
Sort By Address	Displays in address order (default).
Unsort	Does not sort.
Attribute	Displays or does not display an attribute area (Attribute). Checked: Displayed (default) Not checked: Not displayed
Pick Up	Displays only the registers selected in IOR Select dialog box. Checked: Displays only the selected register. Not checked: Displays all the registers (default).
Select...	Opens the IOR Select dialog box.
Compulsion Read	Forcibly reads the peripheral I/O registers that are disabled from being read because their values will be changed, or the data of the I/O ports and I/O protect area added in Add I/O Port dialog box.

### Function buttons

Refresh	Updates the contents of the window with the latest data.
Close	Closes this window.

**Related operations**

---

**To change a peripheral I/O register value**

To change the contents of a peripheral I/O register or I/O port, move the cursor to the Value area on the window, and write a value directly from the keyboard.

Press the return key to set the new value. To cancel the change, press the ESC key.

Note that the values of read-only peripheral I/O registers and I/O ports cannot be changed.

**To select a peripheral I/O register or I/O port to be displayed**

Use the IOR Select dialog box, which can be opened by selecting right-click menu -> [Select...] in the IOR window.

**To change the display sequence**

The display sequence can be changed by selecting right-click menu -> [Sort By Name/Sort By Address/Unsort].

**To change the display radix**

Select the line whose radix is to be changed, and select right-click menu -> [Bin/Oct/Dec/Hex].

**To add, change the contents of, and delete a user-defined I/O port**

To manipulate an I/O port, use the Add I/O Port dialog box, which can be opened by selecting right-click menu -> [Add I/O Port...] in the IOR window.

**To change the display start position**

Use the Address move dialog box, which can be opened by selecting right-click menu -> [Move...] in the Assemble window.

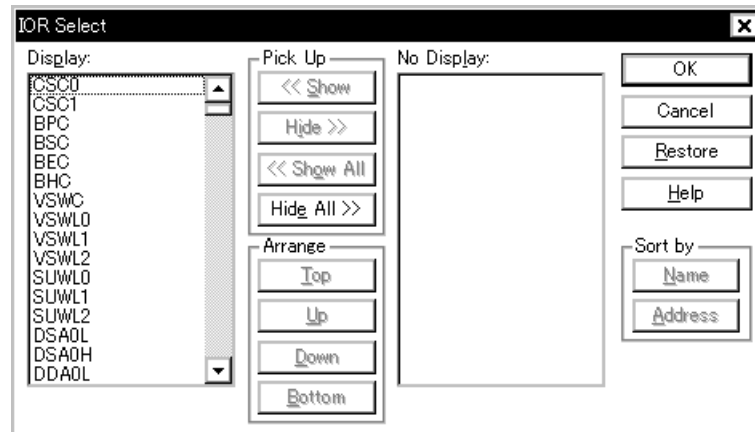


## IOR Select dialog box

This dialog box is used to select peripheral I/O registers and I/O ports that are not displayed the IOR window.

It is also used to specify the sequence in which registers and ports are displayed.

Figure 7-43 IOR Select Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

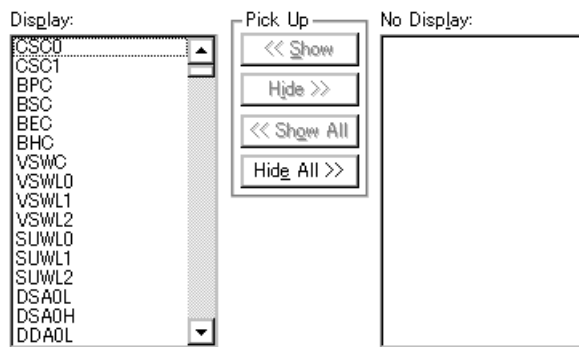
When IOR window is the current window

- Select [View] -> [Select...] from the menu bar  
(or press Alt+V, and E in that order).

### Explanation of each area

The IOR Select dialog box consists of the following areas and buttons:

- (1) Display, Pick Up, No Display (display peripheral I/O register selection area)
- (2) Arrange (display sequence specification button)
- (3) Sort by (No Display list display sequence change button)

**(1) Display, Pick Up, No Display (display peripheral I/O register selection area)**

This area is used to select registers and I/O ports that are displayed or not displayed on the IOR window.

<b>Display</b>	Registers and I/O ports are displayed in the IOR window
<b>No Display</b>	Registers and I/O ports are not displayed in the IOR window

To display or not display a register or I/O port in the IOR window, select the name of the register or I/O port from these lists, and click the <<<Show> or <Hide>>> button. To display all the peripheral I/O registers, click the <<<Show All> button; to not display any of the registers, click the <Hide All>>> button.

<< Show	Moves the register or I/O port selected from the <b>No Display</b> list to <b>Display</b> .
Hide >>	Moves the register or I/O port selected from the <b>Display</b> list to <b>No Display</b> .
<< Show All	Moves all the registers and I/O ports to <b>Display</b> .
Hide All >>	Moves all the registers and I/O ports to <b>No Display</b> .

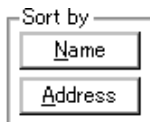
When selecting a peripheral I/O register from the list, two or more registers can be selected by clicking any of the above buttons with the Ctrl or Shift key held down.

**(2) Arrange (display sequence specification button)**

This button sets the sequence in which the registers and I/O ports are displayed in the **Display** list. A peripheral I/O register or I/O port selected in the **Display** list can be moved to the top of the list by pressing the <Top> button, or to the bottom of the list by pressing the <Bottom> button. The <Up> button moves the register or port one line up. The <Down> button moves the register or port one line down.

Top	Moves the selected peripheral I/O register or I/O port to the top of the list.
Up	Moves the selected peripheral I/O register or I/O port one line up.
Down	Moves the selected peripheral I/O register or I/O port one line down.
Bottom	Moves the selected peripheral I/O register or I/O port to the bottom of the list.

(3) **Sort by (No Display list display sequence change button)**



This button selects the sequence in which peripheral I/O registers and I/O ports are displayed in the **No Display** list.

Pressing the <Name> button sorts the registers and ports into alphabetical order. The <Address> button sorts them into address order.

Name	Displays in alphabetical order.
Address	Displays in address order.

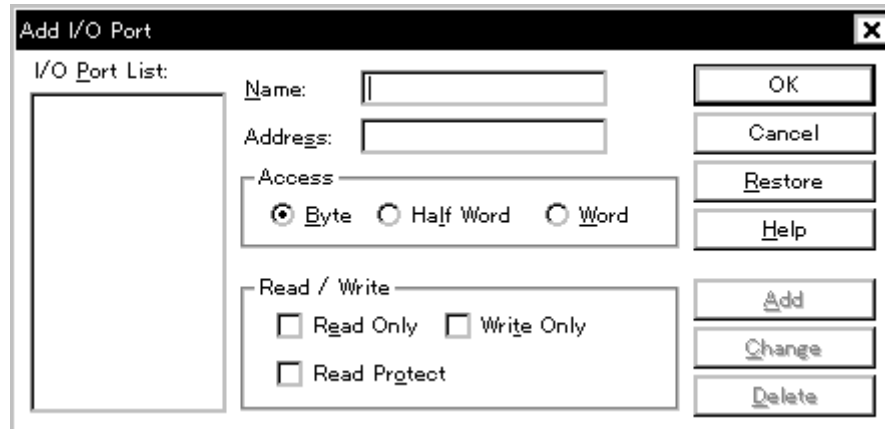
**Function buttons**

OK	Reflects the selection in this dialog box in IOR window and closes this dialog box.
Cancel	Cancels the changes and closes this dialog box.
Restore	Restores the original status.
Help	Opens the Help window.

## Add I/O Port dialog box

This dialog box is used to register an I/O port to be added to the IOR window.

Figure 7-44 Add I/O Port Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

- Select [Option] ->[Add I/O Port...] from the menu bar (or press Alt+O, and P in that order).

### Explanation of each area

The Add I/O Port dialog box consists of the following areas:

- (1) I/O Port List: (I/O port selection area)
- (2) Name: (I/O port name specification area)
- (3) Address: (I/O port address specification area)
- (4) Access (access size specification area)
- (5) Read / Write (access attribute specification area)

**(1) I/O Port List: (I/O port selection area)**

I/O Port List:



This area lists the I/O ports currently registered.

If a new I/O port is registered, it is added to this list. In addition, an I/O port already registered can be selected and changed or deleted.

**(2) Name: (I/O port name specification area)**Name: 

This area is used to specify an I/O port name to be added. An I/O port name can be up to 15 characters long.

**(3) Address: (I/O port address specification area)**Address: 

This area is used to specify the address of the I/O port to be added.

Only an address in one of these areas can be specified: Target area, IOR area

An address can be also specified by a symbol or expression (refer to "Table 7-10 Specifying Symbols"). The default radix for inputting a numeric value is hexadecimal.

**(4) Access (access size specification area)**

Access

☒ Byte ☐ Half Word ☐ Word

This area is used to specify the access size of the I/O port to be added.

Byte	8 bits (default)
Half Word	16 bits
Word	32 bits

**(5) Read / Write (access attribute specification area)**

Read / Write

☐ Read Only    ☐ Write Only

☐ Read Protect

This area is used to specify the access attribute of the I/O port to be added.

In the default condition, all the attributes are not checked (i.e., the I/O port can be both read and written).

<a href="#">Read Only</a>	Read only
<a href="#">Write Only</a>	Write only
<a href="#">Read Protect</a>	Read-protected

**Function buttons**

OK	Reflects the result of addition in IOR window and closes this dialog box.
Cancel	Cancels changes and closes this dialog box.
Restore	Restores the original status.
Help	Opens the Help window.
Add	Adds an I/O port of the specified address.
Change	Changes the setting of the I/O port selected in <a href="#">I/O Port List</a> .
Delete	Deletes the I/O port selected in <a href="#">I/O Port List</a> .

## Event Manager

This window is used to display, enables/disables, and delete each event.

It is also used to register the various conditions events and manage the event conditions for setting them.

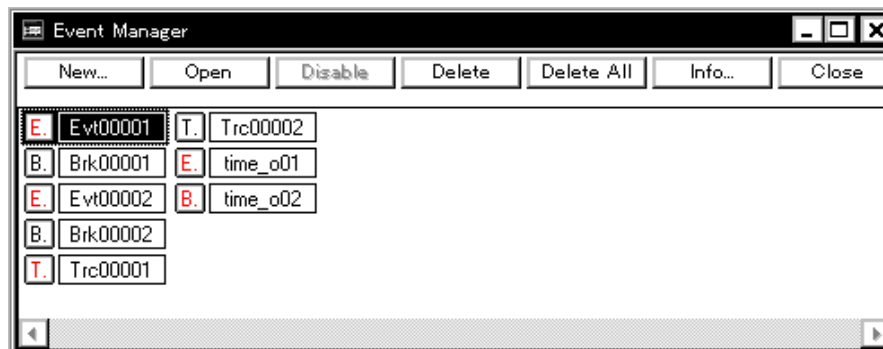
Up to 256 conditions can be registered as event conditions or various event conditions. However, the number of event conditions that can be used (enabled) at the same time is limited as follows:

Table 7-18 Maximum Number of Valid Events for Each Event Condition

Connected IE		Event		Event link	Break condition
		Execution	Access		
IE-V850E1-CD-NW	Nx85ET(RCU0+TEU+TRCU)	10 <sup>a</sup>	4 <sup>b</sup>	1	14
	Nx85E901(RCU0), RCU1	2 <sup>c</sup>		1 <sup>d</sup>	2 <sup>c</sup>

- Two events before execution (range cannot be specified) and eight events after execution (four events if a range is specified)
- Two events when a range specified
- The address range of the before execution cannot be specified.
- Can be set only in Phase 1 and Phase 2 (must always be set in two stages), without pass count and Disable condition specification.

Figure 7-45 Event Manager



The section explains the following items:

- Opening
- Explanation of each area
- Functions often used (right-click menu)
- Function buttons

- Related operations
  - To save/restore event information
  - To enable/disable an event
  - To change the display sequence of event icons
  - To display details of event condition
  - To delete an event
  - To manage a software break event
  - To create an event condition
  - To create an event link condition
  - To set an event condition and event link condition as various event condition

## Opening

---

This window can be opened as follows:

- Click the **Mgr** button.
- Select [Event] -> [Event Manager] from the menu bar  
(or press Alt+N, and M in that order).

In the Event dialog box

- Click the <Manager> button.  
(or press Alt+G).

In the dialog box for selecting the event type

- Click the <Manager> button.  
(or press Alt+M).

## Explanation of each area

---

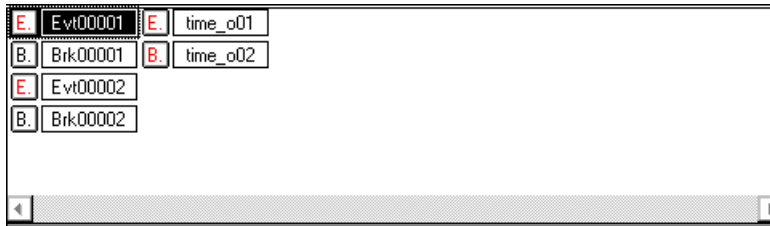
The Event Manager consists of the following areas:

- (1) Event display area
- (2) Event details display area



**(1) Event display area**

[When list is displayed]



The event display area displays the icons (event icons) of the registered event, event link, break event conditions.

An event icon consists of a mark indicating the type of event, and an event name.

By selecting the right click menu -> [Detail], the details can be displayed (refer to "Event details display area").

The meaning of each mark is as follows:

Table 7-19 Marks of Event Icon

Mark	Meaning
<b>E.</b>	Event condition
<b>L.</b>	Event link condition
<b>B.</b>	Break event condition

The color of the character on the mark indicates the set status of the event.

Table 7-20 Color of Character on Mark and its Meaning

Character Color	Mark	Meaning
Red	E. L.	Indicates that the event condition or event link condition which is used for various event conditions is enabled.
	B.	Indicates that the various event conditions is enabled. The various events occurs when its condition is satisfied.
Black	E. L.	Indicates that the event condition or event link condition which is used for various event conditions is disabled.
	B.	Indicates that the various event conditions is disabled. The various events do not occur even when its condition is satisfied.
Yellow	E. L.	Indicates that the symbol specified for an event is held pending because it cannot be recognized by the program currently loaded.
	B.	Indicates that the various events is held pending. The various events do not occur even when its condition is satisfied.

This area also has the following four functions:

### Jump function

If the selected icon is that of an event condition, you can jump to the Source window, Assemble window, or Memory window, using an address condition as the jump pointer. You can jump easily by using the right click menu.

The jump pointer will be as follows, depending on the set address condition.

Table 7-21 Address Condition and Jump Pointer

Setting of Address Condition	Jump Pointer
Setting as a point	Specified address
Setting as a range	Lower address (Point address before the mask if a mask is specified)
Setting as a bit	Address at the bit position

The jump destination window will be displayed from the jump pointer.

### Open function

Opens the event setting dialog box corresponding to the selected event condition. Each setting dialog box displays the contents of the selected event condition.

#### Operation

- Double-click the icon, or click the <Open> button with selecting the icon.

### Enable status selection function

The event corresponding to the selected icon can be enabled or disabled.

The break events (but not the event and event link conditions) can be manipulated via their icons.

#### Operation

- 1 Click on the mark of the icon using the left mouse button.
- 2 Select the icon, click the <Enable> or <Disable> button.

If the event is enabled, it is disabled (black); if the event is disabled, it is enabled (red). The < Enable> button is displayed if the icon of an event that is disabled is selected; the <Disable> button is displayed if the icon of an event that is enabled is selected.

### Deletion function

The registered event and set contents of the selected icon can be deleted.

#### Operation

- 1 Select an icon.  
Two or more icons can be selected by using Shift and Ctrl keys. To select all the icons, select [View] -> [Select All] from the menu bar.
- 2 Click the <Delete> button or press the Delete key.

**Caution** An event condition or event link condition can be deleted only when it is not being used for another event condition. If it is being used for another event, delete the event that is using the event condition or event link condition, and then delete the event condition or event link condition.

**(2) Event details display area**

[In detailed display mode]

<b>E</b>	Evt00001	[S]EX [A]main.c#82(0x44e)
<b>E</b>	Evt00002	[S]EX [A]main.c#94(0x4be)
<b>E</b>	time_o01	[S]W [Z]W [A]time_over(0x100410)
<b>B</b>	Brk00001	[B]Evt00001
<b>B</b>	Brk00002	[B]Evt00002
<b>B</b>	time_o02	[B]time_o01

This area displays the detailed information corresponding to each event icon.

This area is displayed only in the detailed display mode.

Contents are displayed by using the following key information as a separator.

Table 7-22 Separator for Displaying Event Details

Key Information	Contents
<b>Event condition</b>	
[S]	Status condition
[Z]	Access size condition
[A]	Address condition Symbol or expression: (actual address)
[D]	Data condition Symbol or expression: (actual address)
[M]	Mask condition
<b>Event link condition</b>	
[P1] - [P4]	Event link condition on "n"th line
[D]	Disable condition
[P]	Pass count condition
<b>Break condition</b>	
[B]	Break condition

**Functions often used (right-click menu)**

In this window, functions that are often used can be selected from the menu that is displayed by clicking the right mouse button. In this way, the same function can be used more easily than with the normal procedure.

Sort By Name	Displays icons in the order of event names.
Sort By Kind	Displays icons in the order of event types.
Unsort	Does not sort icons (default).
Detail	Displays the details.

Overview	Displays a list (default).
Source Text	Displays the corresponding source text and source line, using the position of the selected event as the jump destination address. If no line information exists at the jump destination address, however, you cannot jump. Opens the Source window. If an active Source window is open, that window is displayed in the forefront (so that it can be manipulated).
Assemble	Displays the Assemble window from the position of the selected event, which is used as the jump destination address. Opens the Assemble window. If an active Assemble window is open, that window is displayed in the forefront (so that it can be manipulated).
Memory	Displays the memory contents from the position of the selected event, which is used as the jump destination address. Opens the Memory window. If an active Memory window is open, that window is displayed in the forefront (so that it can be manipulated).

### Function buttons

New...	Opens the dialog box to create new event condition. By clicking each button of <Event...>, <Event.Link.>, <Break...>, the corresponding event setting dialog box can be opened with the new event name set. After the event setting dialog box has been opened, this dialog box is closed. <Manager...> Opens the Event Manager. <Cancel> Closes the dialog box to create new event condition.
Open	Opens the various event setting dialog box corresponding to the selected event condition. Each setting dialog box displays the contents of the selected event condition. This button is invalid if no event condition is selected or if two or more event conditions are selected. The Enter key performs the same operation.
Enable/ Disable	Enables or disables the selected various event condition (except event conditions and event link conditions). This button is invalid if an enabled/disabled event condition is not selected.
Delete	Deletes the selected event condition. When an event condition or an event link condition is to be deleted, an error occurs and the event condition or event link condition cannot be deleted if the event is used as a various event condition. This button is invalid if no event condition is selected.
Delete All	Deletes all event conditions except software break events
Info...	Opens the Select Display Information dialog box. This dialog box is used to change the display mode and rearrange event names. <Sort by Name> Sorts events into name order. <Sort by Kind> Sorts events into type order. <Unsort> Displays events in the order in which they have been registered without sorting the events. <Detail> Sets the detailed display mode. <Overview> Sets the list display mode. <Cancel> Closes this dialog box (same as ESC key).
Close	Closes this window.

## Related operations

### To save/restore event information

Use this window as the current window, and open the View File Save dialog box by selecting [File] -> [Save As...] from the menu bar.

To restore the set status of an event, open the View File Load dialog box by clicking the **Open** button. By loading an event setting file (\*.evn), the previously set event information can be restored.

### To enable/disable an event

An event can be enabled or disabled by clicking the mark of the event icon to be changed. By clicking Enable (red), the event can be disabled (black): by clicking Disable (black), the event can be enabled (red). This change can be also performed on the Event Manager or in each event setting dialog box.

### To change the display sequence of event icons

Select the right-click menu -> [Sort By Name/Sort By Kind/Unsort] on the Event Manager.

### To display details of event condition

Select the right-click menu -> [Detail] on the Event Manager.

By selecting [Overview], the list can be displayed again.

### To delete an event

Select the icon of the event to be deleted, and click the <Delete> button.

**Caution** An event condition or event link condition cannot be deleted if it is being used for an event. Delete the event first, and then delete the event condition or event link condition.

### To manage a software break event

Use the Software Break Manager, which is opened by selecting [Event] -> [Software Break Manager] from the menu bar.

### To create an event condition

Create an event condition in the Event dialog box, which can be opened by clicking the **Evn** button. In addition, an event condition that is created when a hardware break is set in the source can also be used (refer to "Breakpoint setting/deletion function").

### To create an event link condition

Create an event link condition in the Event Link dialog box, which can be opened by selecting [Event] -> [Event Link] from the menu bar.

### To set an event condition and event link condition as various event condition

Set an event condition or event link condition as a various event condition in the Break dialog box which can be opened by clicking the corresponding button, such as <Break...> in the Event dialog box or Event Link dialog box.

By dragging and dropping the event condition to be set from the Event Manager area at the lower part of the screen to the condition setting area on each event setting dialog box, the event condition can be easily set. Each event setting dialog box can also be opened by selecting from the [Event] menu.

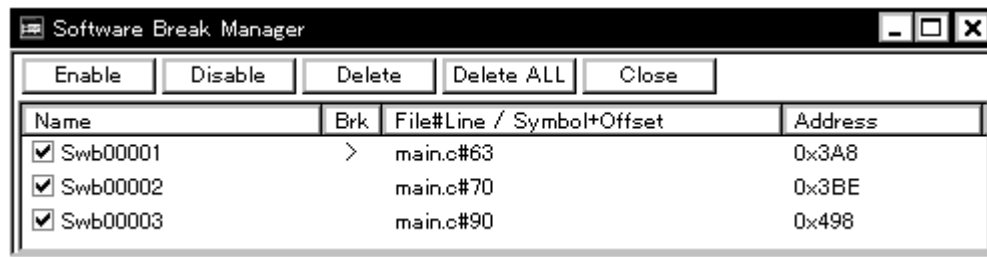
## Software Break Manager

This window is used to display, enable or disable, and delete software breaks.

Up to 2000 software break events can be enabled at the same time (refer to "Software break").

Software breakpoints cannot be set in this window; they can be set in the Source window or Assemble window.

Figure 7-46 Software Break Manager



The section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

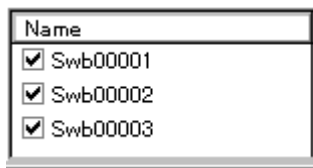
This window can be opened as follows:

- Select [Event] -> [Software Break Manager] from the menu bar (or press Alt+N, and M in that order).

### Explanation of each area

The Software Break Manager consists of the following areas:

- (1) Name (event name display area)
- (2) Brk (break mark display area)
- (3) File#Line / Symbol+Offset (breakpoint display area)
- (4) Address (address display area)

**(1) Name (event name display area)**

This area displays the names of registered events, and the check boxes that indicate whether each event is enabled or disabled.

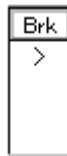
An event name is displayed in the form of **Swb+[number]** in the default condition. It can be changed to an alphanumeric string of up to 256 characters.

To change an event name, select and click a name. Then directly edit the name. To set the editing, press the Enter key.

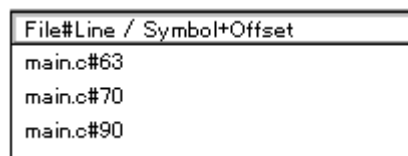
When an event is enabled, the check box is checked.

Furthermore, the name jumps to the Source window by double-clicking an event name if the event name corresponds to the source line, whereas the name jumps to the Assemble window if it does not correspond to the source line.

**Reference** By clicking Name (on the label), the character strings of the displayed items can be compared and sorted lexicographically (in alphabetical order). Whether the character strings are compared or sorted in ascending or descending order can be alternately selected by clicking the mouse.

**(2) Brk (break mark display area)**

The '>' mark is displayed for a software break event that is set at the current PC position (so that the software break event that caused a break can be easily identified).

**(3) File#Line / Symbol+Offset (breakpoint display area)**

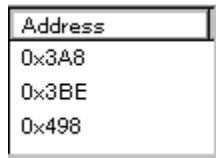
This area displays the location at which a software break event was set as follows:

If the event corresponds to the source line.	Program\$file name#line number
If the event dose not correspond to the source line.	Program\$file name#symbol+offset

Events are evaluated based on this when a symbol is re-downloaded.

**Reference** By clicking File#Line/Symbol+Offset (on the label), the character strings of the displayed items can be compared and sorted lexicographically (in alphabetical order). Whether the character strings are compared or sorted in ascending or descending order can be alternately selected by clicking the mouse.

(4) **Address** (address display area)



This area displays the address at which a software break event is set.

**Reference** By clicking **Address** (on the label), the numeric values of the displayed items can be compared and sorted. Whether the values are compared or sorted in ascending or descending order can be alternately selected by clicking the mouse.

#### Function buttons

---

Enable	Enables the selected event.
Disable	Disables the selected event.
Delete	Deletes the selected event.
Delete ALL	Deletes all the set software break events.
Close	Closes this window.



## Event dialog box

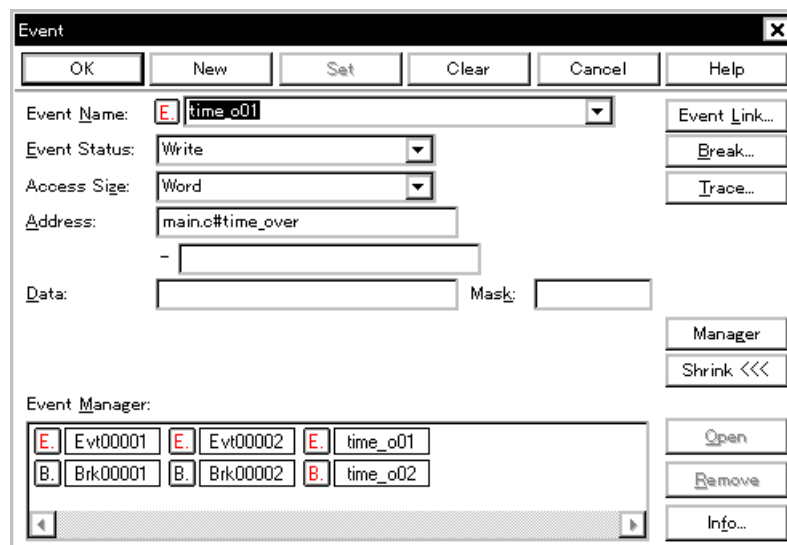
This window is used to register and display event conditions.

An event condition registered in this dialog box will be automatically registered to Event Manager.

As an event condition, one event can be set for two or more event conditions (break and event link).

Up to 256 event conditions can be registered. However, the number of event conditions that can be simultaneously used is limited (refer to "Table 7-18 Maximum Number of Valid Events for Each Event Condition").

Figure 7-47 Event Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

## Opening

This dialog box can be opened as follows:

### In normal mode

If the Event dialog box is opened as follows, an event condition can be registered without its purpose being specified.

- Click the **Ev~~n~~** button.
- Select [Event] -> [Event...] from the menu bar  
(or press Alt+N, and E in that order).

### In select mode

If the <OK> button is pressed when the Event dialog box has been opened as follows, an event condition can be registered in the setting dialog box from which this dialog box was opened.

- In each various event setting dialog box  
Click the <Add Event...> button.  
(or press Alt+E).  
In the select mode, the setting dialog box from which the Event Link dialog box was opened is displayed on the title bar.

## Explanation of each area

The Event dialog box consists of the following areas:

- (1) Event Name (event name setting area)
- (2) Event Status (status selection area)
- (3) Access Size (access size selection area)
- (4) Address (address setting area)
- (5) Data, Mask (data setting area)
- (6) Event Manager (event manager area)

### (1) Event Name (event name setting area)

Event Name: 

This area is used to set a event name.

Directly input an alphanumeric string of up to eight characters as a name.

To display the contents of an already created event condition, select from the drop-down list.

In the select mode, the selected event condition can be set in the event condition setting area of the setting dialog box that called the Event dialog box.

The mark on the left of this area indicates the utilization status of events (refer to "Table 7-20 Color of Character on Mark and its Meaning"). The gray E. mark indicates that the event condition is being edited and has not been registered yet.

**(2) Event Status (status selection area)**

Event Status:

This area is used to select a status condition.

By specifying a status condition, the type of the execution event and an access event is determined (if an execution event is specified, nothing can be input to the [Access Size](#), [Data Mask](#), and [Data](#) areas).

The status conditions that can be specified are listed below. A status condition can also be specified in abbreviated form.

Uppercase and lowercase characters are not distinguished in the input status condition.

Table 7-23 List of Status Conditions

**<Execution event>**

Status	Abbreviation	Meaning
<b>Execution</b>	EX	Program execution
<b>Before Execution</b>	EX-B	Program execution (break before execution) <sup>a</sup>

- a. Two or more break events can be specified. but up to two enable break events including access events.

**<Access event>**

Status	Abbreviation	Meaning
<b>R/W</b>	RW	Data read/write
<b>Read</b>	R	Data read
<b>Write</b>	W	Data write

**(3) Access Size (access size selection area)**

Access Size:

This area is used to set and select an access size condition.

By specifying an access size condition, the access width of a data condition to be detected by an access event is determined.

The access size conditions that can be specified are listed below. An access size condition can also be specified in abbreviated form. Uppercase and lowercase characters are not distinguished in the input access size condition.

Table 7-24 List of Access Size Conditions (Event)

Size	Abbreviation	Meaning
Byte	B	Detects data condition with 8-bit width (only during 8-bit access).
Half Word	HW	Detects data condition with 16-bit width (only during 16-bit access).
Word	W	Detects data condition with 32-bit width (only during 32-bit access).

Table 7-24 List of Access Size Conditions (Event)

Size	Abbreviation	Meaning
No Condition	NC	Does not detect access size (nothing can be input to the <a href="#">Data</a> area).
Bit	1	<p>Detects data condition with 1-bit width (only during 8-bit access).</p> <p>In this case, a data condition is detected with 1-bit width. Because of the operation of the in-circuit emulator, access to a bit is not directly detected; the ID850NWC detects a dummy bit access by internally setting address conditions and data conditions as follows:</p> <p><b>Input example:</b>  Address: FE20.1  Data: 1   </p> <p><b>Setting to in-circuit emulator:</b>  Address: FE20  Data: 00000010B  Mask: 11111101B</p> <p>If another bit of the same address is accessed or if all the 8 bits of the same address are accessed, therefore, an event is detected in accordance with the specified status if the address and bit match the specified value of [address.bit].</p>

When data is written to a bit, all the 8 bits are read/written. If read or read/write is specified as the status, an event occurs if a read operation is performed at this time if the value of the specified [Address.bit] matches.

If no access size is specified, judgment is automatically made from the address condition and data condition, and the following is set:

- Bit if the address condition is set in bit units
- Byte if the address condition is set in 8-bit units
- Half Word if the address condition is set in 16-bit units
- Word if the address condition is set in 32-bit units
- No Condition if no data condition is specified

#### (4) [Address](#) (address setting area)

Address:   
-

An address can be also specified by a symbol or expression (refer to "Table 7-10 Specifying Symbols"). The default radix for inputting a numeric value is hexadecimal.

##### Settable range

0 =< address value =< 0xFFFFFFFF

##### [Address](#)

Set an address condition (lower address - higher address).

**Caution** Specify a 28-bit address, since physical address and image space are distinguished in setting event.

The following can be set:

<b>Setting as a point</b>	Set a value to only the lower address, or set the same value to the lower address and the higher address.
<b>Setting as a range</b>	Set a value to the lower address and the higher address.
<b>Setting as a bit</b>	Set a value to only the lower address, or set the same value to the lower address and the higher address. Specify a value in the form of <b>address.bit</b> . The value of bit, which indicates the bit position, must be $0 \leq \text{bit} \leq 7$ .

(5) **Data, Mask** (data setting area)

Data:  Mask:

**Data**, which sets an data value, and **Mask**, which sets the mask value of an data value, are set as data conditions.

The default radix for inputting a numeric value is hexadecimal.

Both the data and mask may be omitted.

The settable range differs as follows depending on the access size condition specified in the **Access Size** area.

Table 7-25 Settable Range of Data Condition

<b>Access Size</b>	Settable range
<b>Byte</b>	0 ≤ data value ≤ 0xFF 0 ≤ mask value ≤ 0xFF
<b>Half Word</b>	0 ≤ data value ≤ 0xFFFF 0 ≤ mask value ≤ 0xFFFF
<b>Word</b>	0 ≤ data value ≤ 0xFFFFFFFF 0 ≤ mask value ≤ 0xFFFFFFFF
<b>Bit</b>	Data value = 0 or 1 Mask value = Cannot be specified.

**Data**

Set a data value. Specify a value within the settable range.

A symbol or expression can be also specified by a symbol or expression ( refer to "Table 7-10 Specifying Symbols").

**Mask**

Set a mask value for the data value.

When a mask is set, the data value for the bit whose mask value is 1 may be 0 or 1.

The data value for a bit whose mask value is 1 may be 0 or 1.

Example 1:

<b>Data</b>	0x4000
-------------	--------

Mask	0xFF
------	------

With this setting, addresses 0x4000 to 0x40FF satisfy the condition.

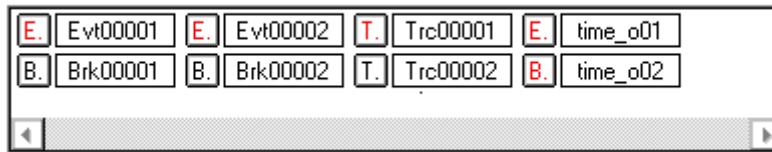
Example 2:

Data	0x4000
Mask	0x101

With this setting, addresses 0x4000, 0x4001, 0x4100, and 0x4101 satisfy the condition.

#### (6) Event Manager (event manager area)

Event Manager:



This area displays a list of various event conditions registered.

In this area, the following operations can be performed.

##### Displaying set contents

Select an event and click the <Open> button or double-click the event. The setting dialog box corresponding to the selected event will be opened and the set contents of the event will be displayed.

##### Delete

A selected event can be deleted by selecting the icon of the event and clicking the <Delete> button or pressing the Delete key while focusing on the event manager area.

##### Changing display mode and sorting

The display mode of and sorting in the event manager area can be selected by clicking the <Info...> button.

#### Function buttons

Related setting event	
OK	Automatically registers the event condition being edited, if any, and closes this dialog box. <b>In the select mode</b> An event condition is selected and the setting dialog box (indicated on the title bar) that called the Event Link dialog box is displayed again. If the calling dialog box has already been closed, the select mode is returned to the normal mode, and the Event dialog box is not closed. Otherwise, this dialog box will be closed.
New	Newly creates an event condition in this dialog box. An event condition name is automatically created and a new event condition is prepared.

Set (when a new event link is created or an event link is changed)	Registers the various event condition. Because the dialog box is not closed even after an event has been registered, new event conditions can be registered. <b>In the select mode</b> An event condition is selected. If there is an event being edited, it is automatically registered and selected.
Clear (when an event is not edited)	Clears the contents of the event condition. This button is displayed when an event condition has not been edited. This button and the <Restore> button are alternately displayed.
Restore (when an event is edited)	Restores the contents of an edited event condition. If an event condition not registered is displayed, all the fields other than the event name field are blank or the default values are set. This button and the <Clear> button are alternately displayed.
Cancel Close	Closes this dialog box. Even if an event condition is being edited, it is not registered and the dialog box is closed. <Cancel> is displayed if a condition is not created, changed, or deleted; <Close> is displayed if it is.
Help	Opens the Help window.
<b>Related dialog box</b>	
Event Link...	Opens the Event Link dialog box.
Break...	Opens the Break dialog box.
Trace...	Cannot be selected.
Manager	Opens the Event Manager.
Open (When only one event is selected)	Opens the various event setting dialog box corresponding to the selected event condition. Each setting dialog box displays the contents of the selected event condition. The Enter key performs the same operation.
Remove (When event is selected)	Deletes the selected event condition. When an event condition or an event link condition is to be deleted, an error occurs and the event condition or event link condition cannot be deleted if the event is used as a various event condition.
<b>Other</b>	
Shrink <<< (When area is displayed)	Turns off display of the event manager area. At this time, the size of the dialog box is reduced.
Expand >>> (When area is not displayed)	Displays the event manager area. At this time, the size of the dialog box is expanded.
Info...	Opens the Select Display Information dialog box. This dialog box is used to change the display mode and rearrange event names. <Sort by Name> Sorts events into name order. <Sort by Kind> Sorts events into type order. <Unsort> Displays events in the order in which they have been registered without sorting the events. <Detail> Sets the detailed display mode. <Overview> Sets the list display mode. <Cancel> Closes this dialog box (same as ESC key).

## Event Link dialog box

This window is used to register and display event link conditions.

A event link condition registered in this dialog box will be automatically registered to the Event Manager.

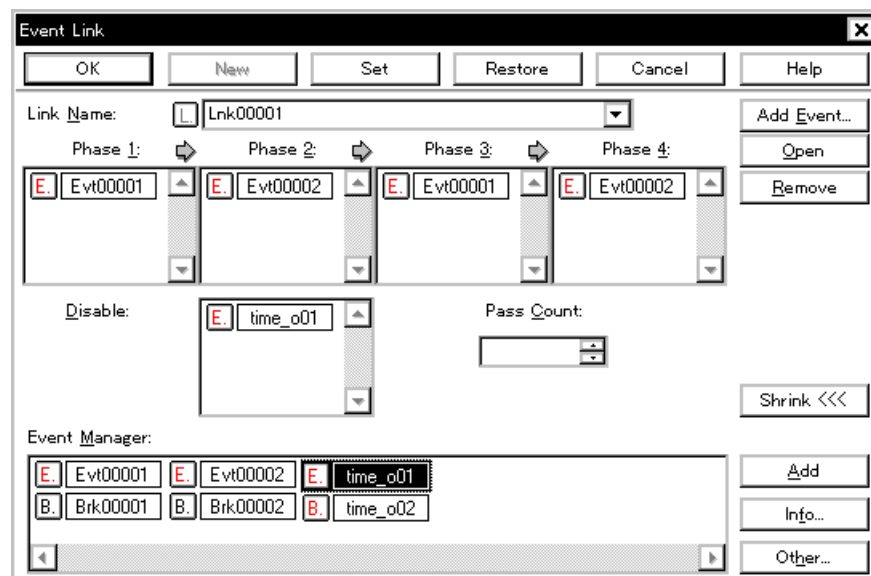
An event link condition is an event condition that is generated only if the user program is executed in the sequence specified by a set event condition.

Up to four sequences can be specified. If a disable condition is detected while the program is being executed, however, the event conditions that have so far been satisfied are initialized, and the event conditions are detected again starting from the first event condition. If a link condition and a disable condition are detected at the same time, the disable condition takes precedence.

As an event condition, one event can be set for two or more event conditions.

Up to 256 event link conditions can be registered. However, the number of event link conditions that can be simultaneously used (valid) is limited (refer to "Table 7-18 Maximum Number of Valid Events for Each Event Condition").

Figure 7-48 Event Link Dialog Box





The section explains the following items:

- Opening
- Explanation of each area
- Function buttons

## Opening

This dialog box can be opened as follows:

### In normal mode

If the Event Link dialog box is opened as follows, an event link condition can be registered without its purpose being specified.

- Select [Event] -> [Event Link...] from the menu bar  
(or press Alt+N, and L in that order).

### In select mode

If the <OK> button is pressed when the Event Link dialog box has been opened as follows, an event link condition can be registered in the setting dialog box from which this dialog box was opened.

- In each various event setting dialog box  
Click the <Add Link...> button.  
(or press Alt+L).  
In the select mode, the setting dialog box from which the Event Link dialog box was opened is displayed on the title bar.

## Explanation of each area

The Event Link dialog box consists of the following areas:

- (1) Link Name (event link name setting area)
- (2) Phase1, Phase2, Phase3, Phase4 (link condition setting area)
- (3) Disable (disable condition setting area)
- (4) Pass Count (pass count setting area)
- (5) Event Manager (event manager area)

### (1) Link Name (event link name setting area)

Link Name: 

This area is used to set a event link name.

Directly input an alphanumeric string of up to eight characters as a name.

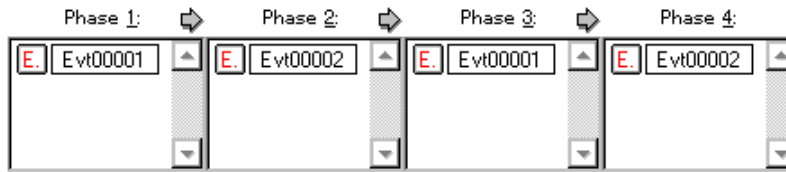
To display the contents of an already created event link condition, select from the drop-down list.

In the select mode, the selected event condition can be set in the event link condition setting area of the setting dialog box that called the Event Link dialog box.

The mark on the left of this area indicates the utilization status of event link condition (refer to "Table 7-

20 Color of Character on Mark and its Meaning"). The mark "L" in gray indicates that an event link condition is being edited and has not been registered yet.

(2) **Phase1, Phase2, Phase3, Phase4 (link condition setting area)**



This area is used to specify the sequence in which event conditions and events are detected.

Set Phase 1 -> Phase 2 -> Phase 3 -> Phase 4, in that order. Phase 4 does not have to be set. In this case, an event occurs when the event condition set for the last phase has been detected. An event condition can be set for only Phase 1 or the same event condition can be set for two or more Phases.

The number of event conditions that can be set to each phase of this area and whole the dialog box, is as follows:

Table 7-26 The Number of Event Conditions in Event Link Dialog Box

Connected IE		Each phase	Total(Each phase area and Disable area) (execution/access)
IE-V850E1-CD-NW	Nx85ET(RCU0+TEU+TRCU)	10 <sup>a</sup>	12(max)
	Nx85E901(RCU0), RCU1	1 <sup>b</sup>	2 <sup>c</sup>

- Can be set in Phase1-4 (expect the events before execution)
- Can be set only in Phase 1 and Phase 2 (must always be set in two stages)
- Only events before execution

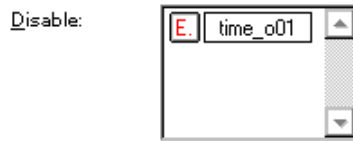
**Setting event condition in each area**

Select an event condition in the [Event Manager](#) area, and click the <Add> button or drag and drop an event condition. An event link condition can also be set by dragging and dropping not only in this dialog box, but also in the Event Manager area of each various event setting dialog box or the Event Manager.

In addition, the Event dialog box can be opened in "select mode" by clicking the <Add Event...> button, and the event condition to be set can be selected.

In the event condition setting area in this dialog box, an event condition can be alternately copied or moved by dragging and dropping.

- If an event condition is dropped only by the mouse, it is moved.
- If an event condition is dropped with the Shift key held down, it is moved.
- If an event condition is dropped with the Ctrl key held down, it is copied.

**(3) Disable (disable condition setting area)**

This area is used to set an event condition that invalidates the event conditions that have so far been satisfied (refer to "Setting event condition in each area").

The number of event conditions that can be set in this area is up to 2 in case of Nx85ET(RCU0+TEU+TRCU).

In case of Nx85E901(RCU0), RCU1, this area does not have to be set.

**(4) Pass Count (pass count setting area)**

Pass Count:



This area is used to set a pass count condition (settable range: 1 to 4095) .

A pass count condition specifies how many times an event condition must be satisfied during user program execution before a given condition is satisfied.

If the pass count is set to 1, the condition is satisfied as soon as the event condition is satisfied.

If no pass count is specified, 1 is assumed.

In case of Nx85E901(RCU0), RCU1, this area does not have to be set.

**(5) Event Manager (event manager area)**

Event Manager:



This area displays a list of various event conditions registered.

For details, refer to "Event Manager (event manager area)".

---

**Function buttons**


---

<b>Related setting event</b>	
OK	<p>Automatically registers the event condition being edited, if any, and closes this dialog box.</p> <p><b>In the select mode</b>            An event condition is selected and the setting dialog box (indicated on the title bar) that called the Event Link dialog box is displayed again. If the calling dialog box has already been closed, the select mode is returned to the normal mode, and the Event dialog box is not closed. Otherwise, this dialog box will be closed.</p>
New	<p>Newly creates an event condition in this dialog box.</p> <p>An event condition name is automatically created and a new event condition is prepared.</p>
Set (when a new event link is created or an event link is changed)	<p>Registers the various event condition. Because the dialog box is not closed even after an event has been registered, new event conditions can be registered.</p> <p><b>In the select mode</b>            An event condition is selected. If there is an event being edited, it is automatically registered and selected.</p>
Clear (when an event is not edited)	<p>Clears the contents of the event condition.</p> <p>This button is displayed when an event condition has not been edited. This button and the &lt;Restore&gt; button are alternately displayed.</p>
Restore (when an event is edited)	<p>Restores the contents of an edited event condition.</p> <p>If an event condition not registered is displayed, all the fields other than the event name field are blank or the default values are set.</p> <p>This button and the &lt;Clear&gt; button are alternately displayed.</p>
Cancel Close	<p>Closes this dialog box.</p> <p>Even if an event condition is being edited, it is not registered and the dialog box is closed.</p> <p>&lt;Cancel&gt; is displayed if a condition is not created, changed, or deleted; &lt;Close&gt; is displayed if it is.</p>
Help	Opens the Help window.
<b>Related dialog box</b>	
Add Event...	<p>Opens the Event dialog box in the select mode, and selects or newly creates an event condition to be set. The event condition will be added to the area selected when the &lt; Add Event...&gt; button is pressed.</p>
Open (When only one event is selected)	<p>Opens the various event setting dialog box corresponding to the selected event condition. Each setting dialog box displays the contents of the selected event condition.</p> <p>The Enter key performs the same operation.</p>
Remove (When event is selected)	<p>Deletes the selected event condition.</p> <p>When an event condition or an event link condition is to be deleted, an error occurs and the event condition or event link condition cannot be deleted if the event is used as a various event condition.</p>
Add	<p>The event condition and event link condition selected in Event Manager area add to setting area with a focus. The event condition will be added to the area selected when this button is pressed.</p>

Other...	<p>Opens the dialog box for selecting the event type.</p> <p>By clicking each button of &lt;Event...&gt;, &lt;Event.Link...&gt;, &lt;Break...&gt;, the corresponding event setting dialog box can be opened with the new event name set. After the event setting dialog box has been opened, this dialog box is closed.</p> <p>&lt;Manager...&gt; Opens the Event Manager.</p> <p>&lt;Cancel&gt; Closes the dialog box to create new event condition.</p>
<b>Other</b>	
Shrink <<< (When area is displayed)	<p>Turns off display of the event manager area.</p> <p>At this time, the size of the dialog box is reduced.</p>
Expand >>> (When area is not displayed)	<p>Displays the event manager area.</p> <p>At this time, the size of the dialog box is expanded.</p>
Info...	<p>Opens the Select Display Information dialog box.</p> <p>This dialog box is used to change the display mode and rearrange event names.</p> <p>&lt;Sort by Name&gt; Sorts events into name order.</p> <p>&lt;Sort by Kind&gt; Sorts events into type order.</p> <p>&lt;Unsort&gt; Displays events in the order in which they have been registered without sorting the events.</p> <p>&lt;Detail&gt; Sets the detailed display mode.</p> <p>&lt;Overview&gt; Sets the list display mode.</p> <p>&lt;Cancel&gt; Closes this dialog box (same as ESC key).</p>

## Break dialog box

This dialog box is used to register; set, and display break event conditions.

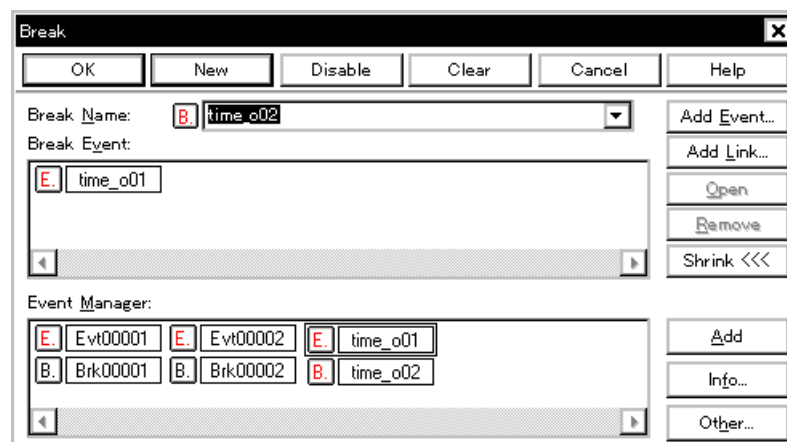
A break event condition registered in this dialog box will be automatically registered to Event Manager.

A break event condition automatically becomes valid when it has been registered.

Up to 256 break event conditions can be registered. However, the number of break event conditions that can be simultaneously used is limited (refer to "Table 7-18 Maximum Number of Valid Events for Each Event Condition"). If the maximum number of break event conditions that can be used at the same time is exceeded or if the number of event conditions or event link conditions used exceeds the maximum limit, therefore, register a break event condition in the disabled state.

If [Run] -> [Ignore Break Point] is selected from the menu bar, a break event condition does not operate even if it is valid.

Figure 7-49 Break Dialog Box



The section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

- Click the **Brk** button.
- Select [Event] -> [Break...] from the menu bar (or press Alt+N, and B in that order).

## Explanation of each area

The Break dialog box consists of the following areas:

- (1) Break Name (break event name setting area)
- (2) Break Event (break condition setting area)
- (3) Event Manager (event manager area)

### (1) Break Name (break event name setting area)

Break Name:  time\_o02

This area is used to set a break event name.


Directly input an alphanumeric string of up to eight characters as a name.

To display the contents of an already created event condition, select from the drop-down list.

The mark on the left of this area indicates the utilization status of events (refer to "Table 7-20 Color of Character on Mark and its Meaning"). The gray mark indicates that an event condition is being edited and has not been registered yet.

By clicking the left mark, an event condition can be validated or invalidated.

### (2) Break Event (break condition setting area)

Break Event:  time\_o01

This area is used to set an event condition for break.

The number of event conditions that can be set in this area, combining execution events and access events, is as follows:

Table 7-27 Number of Events Settable in Break Condition Setting Area

Connected IE		Total (execution/access)
IE-V850E1-CD-NW	Nx85ET(RCU0+TEU+TRCU)	14(10 <sup>a</sup> /4 <sup>b</sup> )
	Nx85E901(RCU0), RCU1	2

a. Two events before execution and eight events after execution(max)

b. four(max)

The number of event link conditions explained in Table 7-18 Maximum Number of Valid Events for Each Event Condition can be used as long as the number of event conditions does not exceed the maximum number of event conditions simultaneously usable.

### Setting event condition and event link condition in each area

Select an event condition or event link condition in the [Event Manager](#) area, and click the <Add> button or drag and drop an event condition or event link condition. An event link condition can also be set by dragging and dropping not only in this dialog box, but also in the Event Manager area of each various event setting dialog box or the Event Manager.

In addition, Event dialog box or Event Link dialog box can be opened in "select mode" by clicking the <Add Event...> or <Add Link...> button, and the event condition or event link condition to be set can be selected.

In the event condition setting area in this dialog box, an event condition can be alternately copied or moved by dragging and dropping.

- If an event condition is dropped only by the mouse, it is moved.
- If an event condition is dropped with the Shift key held down, it is moved.
- If an event condition is dropped with the Ctrl key held down, it is copied.

### (3) [Event Manager](#) (event manager area)

This area displays a list of various event conditions registered.

For details, refer to "Event Manager (event manager area)".

### Function buttons

Related setting event	
OK	Automatically registers the event condition being edited, if any, and closes this dialog box. The event condition becomes valid as soon as it has been registered.
New	Newly creates an event condition in this dialog box. An event condition name is automatically created and a new event condition is prepared.
Set (when a new event link is created or an event link is changed)	Registers the various event condition. Because the dialog box is not closed even after an event has been registered, new event conditions can be registered. Each event condition becomes valid as soon as it has been registered. If an already registered event condition is displayed, the <Enable> or <Disable> button is alternately displayed.
Enable/Disable (when a registered event is displayed)	Validates (enables) or invalidates (disables) the selected event condition. However, event conditions and event link conditions cannot be enabled or disabled. This button and the <Set> button are alternately displayed.
Clear (when an event is not edited)	Clears the contents of the event condition. This button is displayed when an event condition has not been edited. This button and the <Restore> button are alternately displayed.
Restore (when an event is edited)	Restores the contents of an edited event condition. If an event condition not registered is displayed, all the fields other than the event name field are blank or the default values are set. This button and the <Clear> button are alternately displayed.



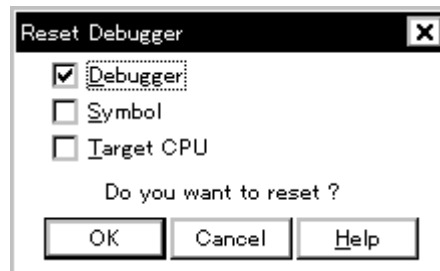
Cancel Close	Closes this dialog box. Even if an event condition is being edited, it is not registered and the dialog box is closed. <Cancel> is displayed if a condition is not created, changed, or deleted; <Close> is displayed if it is.
Help	Opens the Help window.
<b>Related dialog box</b>	
Add Event...	Opens the Event dialog box in the select mode, and selects or newly creates an event condition to be set. The event condition will be added to the area selected when the < Add Event...> button is pressed.
Add Link...	Opens the Event Link dialog box in the select mode, and selects or newly creates an event link condition. The event condition will be added to the area selected when the < Add Link...> button is pressed.
Open (When only one event is selected)	Opens the various event setting dialog box corresponding to the selected event condition. Each setting dialog box displays the contents of the selected event condition. The Enter key performs the same operation.
Remove (When event is selected)	Deletes the selected event condition. When an event condition or an event link condition is to be deleted, an error occurs and the event condition or event link condition cannot be deleted if the event is used as a various event condition.
Add	The event condition and event link condition selected in Event Manager area add to setting area with a focus. The event condition will be added to the area selected when this button is pressed.
Other...	Opens the dialog box for selecting the event type. By clicking each button of <Event...>, <Event.Link...>, <Break...>, the corresponding event setting dialog box can be opened with the new event name set. After the event setting dialog box has been opened, this dialog box is closed. <Manager...> Opens the Event Manager. <Cancel> Closes the dialog box to create event condition.
<b>Other</b>	
Shrink <<< (When area is displayed)	Turns off display of the event manager area. At this time, the size of the dialog box is reduced.
Expand >>> (When area is not displayed)	Displays the event manager area. At this time, the size of the dialog box is expanded.
Info...	Opens the Select Display Information dialog box. This dialog box is used to change the display mode and rearrange event names. <Sort by Name> Sorts events into name order. <Sort by Kind> Sorts events into type order. <Unsort> Displays events in the order in which they have been registered without sorting the events. <Detail> Sets the detailed display mode. <Overview> Sets the list display mode. <Cancel> Closes this dialog box (same as ESC key).

## Reset Debugger dialog box

This dialog box is used to initialize the ID850NWC, CPU, and symbol information.

Specify what is to be initialized by using the check box. In the default condition, the ID850NWC is initialized.

Figure 7-50 Reset Debugger Dialog Box



This section explains the following items:

- Opening
- Explanation of each area
- Function buttons

### Opening

This dialog box can be opened as follows:

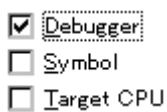
- Select [File] -> [Debugger Reset...] on the menu bar (or press Alt+F, and R in that order).

### Explanation of each area

The Reset Debugger dialog box consists of the following areas:

- (1) Reset subject selection area

#### (1) Reset subject selection area



This area is used to select what is to be initialized.

Debugger	Initializes the ID850NWC (default).
Symbol	Initializes the symbol information.
Target CPU	Initializes the CPU.

---

**Function buttons**

---

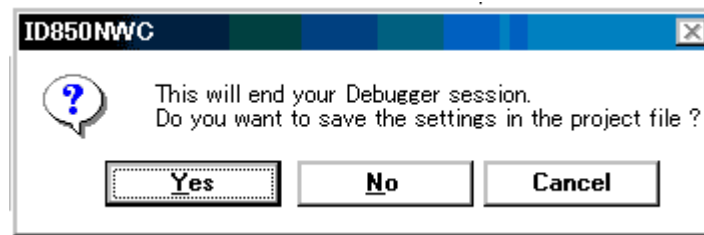
OK	Initializes the ID850NWC, symbol information, or CPU.
Cancel	Ignores the setting and closes this dialog box.
Help	Opens the Help window.

## Exit Debugger dialog box

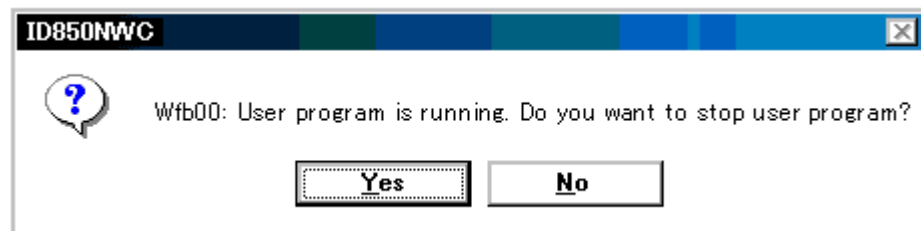
This dialog box is used to select whether the current debug environment is saved to a project file or not before terminating the ID850NWC.

It can be specified in the Debugger Option dialog box that the ID850NWC is terminated without this confirmation dialog box being opened.

Figure 7-51 Exit Debugger Dialog Box



**Reference** If an attempt is made to terminate the ID850NWC while the user program is being executed, the following message is displayed and the program execution can be stopped.



### <Yes> button is selected

Execution of the user program is stopped and then the Exit Debugger dialog box is displayed. If it is specified in the Debugger Option dialog box that the Exit Debugger dialog box is not to be displayed, however, the ID850NWC is terminated.

### <No> button is selected

Execution of the user program is not stopped and the Exit Debugger dialog box is not displayed. The ID850NWC is not terminated.

This section explains the following items:

- Opening
- Function buttons

---

## Opening

---

This dialog box can be opened as follows:

- Select [E]ile -> [E]xit from the menu bar (or press Alt+F, and X in that order).
- Press the Alt+F4 keys simultaneously.
- If forcible termination, such as to terminate the application, has been executed on the task list that terminates Windows.

---

## Function buttons

---

Yes	Saves the current debug environment to a project file, closes all the windows, and terminates the ID850NWC. If a project file name is not specified, the Project File Save dialog box is opened. If the < Cancel> button is selected on the Project File Save dialog box, the environment is neither saved to a project file nor is the ID850NWC terminated. If a project file is loaded or saved during debugger operation, this button has the default focus.
No	Closes all the windows and terminates the ID850NWC. If a project file is not loaded or saved during debugger operation, this button has the default focus.
Cancel	Closes this dialog box without executing anything.

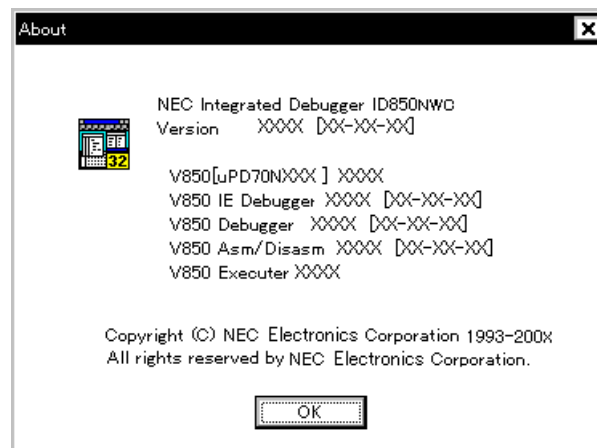
## About dialog box

This dialog box displays the version information of the ID850NWC and in-circuit emulator (the year is displayed in 4 digits).

The following version information is displayed:

- Product version of ID850NWC
- Version of device file
- Version of GUI
- Version of debugger DLL
- Version of assembler DLL
- Version of executor

Figure 7-52 About Dialog Box



This section explains the following items:

- Opening
- Function buttons

### Opening

This dialog box can be opened as follows:

- Select [H]elp->[A]bout  
(or press Alt+H, and A in that order).

### Function buttons

OK	Closes this dialog box.
----	-------------------------

## Console window

This window is used to input commands that control the ID850NWC.

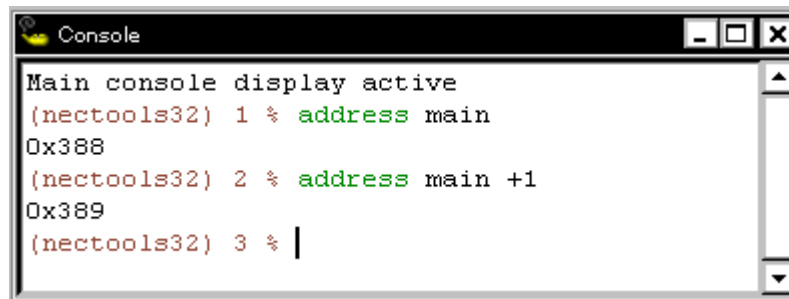
Because the key bind is Emacs-like, the accelerator key is not acknowledged if the Console window is active.

However, the F1 key displays the help of the Console window.

While the Console window is open, an error message window with only an <OK> button is displayed in the Console window.

Refer to "COMMAND REFERENCE" for details on the command specifications.

Figure 7-53 Console Window



This section explains the following items:

- Opening

### Opening

This window can be opened as follows:

- Select [Browse] -> [Console] from the menu bar (or press Alt+B, and N in that order).

## CHAPTER 8 COMMAND REFERENCE

This section explains the command functions of the ID850NWC in detail.

### 8.1 Command Line Rules

The specification of command lines has the following rules:

- Command name, option, and argument are specified for command line.
- To divide words, a space (space key or tab key) is used.
- At the end of a line, a line feed character or a semicolon is used.
- When a command name and an option are entered to the point of identifiability, they are recognized.
- In script, command names have to be entered completely.

#### Command format

command *-options arg1 arg2 arg3 ...*

### 8.2 Command List

The list of ID850NWC commands is shown in Table 8-1 and Table 8-2 .

Table 8-1 List of Debugger Control Commands

Command Name	Function
address	Evaluation of address expression Converts the address expression specified with expression into address.
assemble	Disassemble/line assemble (a) Assembles the character strings specified with code from the address specified with address.
batch	Executing batch (with echo) Executes in batch displaying files specified with script name on the screen.
breakpoint	Setting/deletion of breakpoint (b) Operates breakpoint specified with options and address.
cache	Setting of cache Sets I cache and D cache.
dbgexit	Terminating ID850NWC Terminates the ID850NWC.
download	Download of files (l) Downloads files specified with filename according to options.
extwin	Creation of expansion window Creates expansion window with script file.
finish	Returning from function Executes until it returns to the program that called the current function.
go	Continuous execution (g) Executes program continuously.



Table 8-1 List of Debugger Control Commands

Command Name	Function
help	Display of help Displays help of Dcl.
hook	Setting of hook Sets the procedure for hook with script file.
ie	Display/setting of IE register This command depends on the IE.
jump	Jump to window (j) Displays the window specified with options.
map	Setting/deletion of memory mapping Sets, deletes, and displays memory mapping.
mdi	Setting of expansion window Sets the size and title name of the expansion window.
memory	Display/setting of memory (m) Sets value in the memory of the address specified with address according to options.
module	Display of the list of files and functions Displays the list of files and functions of the load module specified with progname.
next	Procedure step (n) Executes the procedure steps.
refresh	Redrawing of window Redraws the window and updates the data.
register	Display/setting of register value and I/O register value Sets value in the register specified with regname.
reset	Reset Resets CPU, ID850NWC or symbols.
run	Reset and execution of CPU (r) Resets program and executes it.
step	Step execution (s) Executes step execution.
stop	Stop execution Stops the program forcibly.
upload	Upload Saves the memory data or coverage data within the specified range in a file.
version	Display of the version information Displays the version information of the debugger.
watch	Display/setting of variables (w) Displays and sets the variables.
where	Trace of stack Executes the back-trace of the stack.
wish	Start of Tclet Starts the script using Tk (Tclet).

Table 8-2 List of Console/Tcl Commands

Command Name	Function
alias	Creation of another name
cd	Change of directory
clear	Clears the screen
echo	Echo
exit	Close/end
history	Display of history
ls	Display of files
pwd	Check of the directory
source	Execution of batch
time	Measurement of time for commands
tkcon	Console control
unalias	Deletion of another name
which	Display of the command path or another name
Other commands	Based on Tcl/Tk 8.1

### 8.3 List of Variables

- dcl(chip) - Chip name read only
- dcl(prjfile) - Project file name read only
- dcl(srcpath) - Source path read only
- dcl(ieid) - IE type read only
- dcl(iestat) - IE status read only
- dcl(bkstat) - Break status read only
- env(LANG) - Language
- dcl\_version - Dcl version read only

### 8.4 List of Packages

- tcltest - Restoration test
- cwind - Automatic window control
- BWidget - Toolkit

### 8.5 Key Bind

- tcsh + emacs like
- Complement of command name [Tab]
- Complement of file name [Tab]
- HTML help [F1]

## 8.6 Expansion window

The expansion windows can be created using Tk.

In the expansion windows, Widget is allocated with '.dcl' as a root instead of '.'.

When the following script files are allocated in bin/itcl/tools/, an expansion window is added on selecting [Browse] -> [Others] from the menu bar.

The mdi command, an exclusive command for expansion windows, has been added.

```
# Sample.tcl
wm protocol .dcl WM_DELETE_WINDOW { exit }
mdi geometry 100 50
button .dcl.b -text Push -command exit
pack .dcl.b
```

In expansion windows, Tk menu commands cannot be used because of the restrictions of MDI windows.

## 8.7 Callback Procedure

Expansion windows can hold dcl\_asyncproc procedures called by asynchronous messages.

```
proc dcl_asyncproc {mid} {
    if {$mid == 19} {
        redraw
    }
}
```

The asynchronous message ID is passed for the argument of the dcl\_asyncproc procedure.

The message IDs are shown below:

Table 8-3 Message ID

Message ID	Meaning
9	After changing configuration
10	After registering event
11	After deleting event
12	Before executing
13	After breaking
14	After resetting CPU
15	After resetting ID850NWC
17	After changing extended option
18	After changing debugger option
19	After downloading
20	After changing memory or register
45	After resetting symbol

## 8.8 Hook Procedure

A hook can be set in the ID850NWC using the hook procedure.

The hook procedures are shown below:

BeforeDownload	Hook before downloading
AfterDownload	Hook after downloading
AfterCpuReset	Hook after resetting CPU
BeforeCpuRun	Hook before starting execution
AfterCpuStop	Hook after breaking

By using hook procedures, register values can be changed before downloading programs or after resetting the CPU.

An actual example of the procedure is shown below. A hook is valid till the ID850NWC is closed.

### [When hook is set with ID850NWC control command]

- 1 Create script file a. with an editor.
- 2 Start up the ID850NWC, select [Browse] menu -> [Console], and open the Console window.
- 3 If the script file is executed in the window as below, the hook in the script file is set.  
%hook test.tcl

### [When hook is set on downloading of project file]

- 1 Create script file a. with an editor.

**Caution** Be sure that the script file name is the same as the project file.

Example:

The script file corresponding to test.prj is test.tcl.

Allocate test.prj, test.pri, and test.tcl in the same directory.

- 2 Start up the ID850NWC and read test.prj.  
The hook in the script file is set.

An example of the contents of a script file is shown as below (using the ICE for the V850E/MS1).

```
proc BeforeDownload {} {  
    register MM 0x7  
    register PMC8 0xff  
    register PMC9 0xff  
    register PMCX 0xe0  
}  
  
proc AfterCpuReset {} {  
    register MM 0x7  
    register PMC8 0xff  
    register PMC9 0xff  
    register PMCX 0xe0  
}
```

## 8.9 Related Files

- Executes when the aliases.tcl console is opened.  
Sets the default alias etc.
- Executes when the project file name.tcl project is opened.  
The hook of BeforeDownload, AfterDownload, AfterCpuReset, BeforeCpuRun and AfterCpuStop can be used.
- Executes when the load module name.tcl load module is downloaded.  
The hook of BeforeDownload, AfterDownload, AfterCpuReset, BeforeCpuRun and AfterCpuStop can be used.

## 8.10 Cautions

- The separator for file and path is a slash (/).
- When a console or an expansion window is active, the accelerator key cannot be acknowledged.
- When a console is open, error messages are output to the console.
- To terminate the command forcibly, close the console.
- The execution of external commands (DOS commands) is OFF by default.

## 8.11 Commands

In this section, each command is explained using the format shown below.

## Command name

---

---

Gives a brief explanation of the function of the command.

### **Name**

---

Describes the command name.

### **Input format**

---

Describes the input format of the command.

### **Functions**

---

Explains the functions of the command.

### **Usage example**

---

Shows an example of the usage of the command.

## address

---

---

- Name
- Input format
- Functions
- Usage example

### Name

---

address - Evaluation of address expression

### Input format

---

**address** *expression*

### Functions

---

Converts the address expression specified by ***expression*** into address.

### Usage example

---

```
(IDCON) 1 % address main
0xaa
(IDCON) 2 % address main+1
0xab
```

## assemble

- Name
- Input format
- Functions
- Usage example

### Name

assemble - Disassemble/line assemble

### Input format

**assemble** *?options? address ?code?*

### Functions

Assembles the character strings specified by **code** from the **address** specified by address.

When '.' is specified for address, it is understood as an **address** continuing from the immediately previous assemble.

When **code** is omitted, it is assembled from the **address** specified by address.

The followings are **options**. They are ignored for assembly.

- code** Command code is also displayed. It is ignored for assembly.
- number number** *number* line is displayed. It is ignored for assembly.

### Usage example

```
(IDCON) 1 % assemble -n 5 main
0x000000aa B7      PUSH HL
0x000000ab B1      PUSH AX
0x000000ac 891C    MOVW AX,SP
0x000000ae D6      MOVW HL,AX
0x000000af A100    MOV  A,#0H
(IDCON) 2 % assemble main mov a,b
(IDCON) 3 % assemble . mov a,b
```



## batch

---

---

- Name
- Input format
- Functions
- Usage example

---

### Name

batch - Executing batch (with echo)

---

### Input format

**batch** *scriptname*

---

### Functions

Executes in batch with displaying files specified by ***scriptname*** on the screen.

Nesting is possible.

---

### Usage example

```
(IDCON) 1 % clear
(IDCON) 2 % batch bat_file.tcl
(IDCON) 3 % tkcon save a:/log.txt
```

# breakpoint

- Name
- Input format
- Functions
- Usage example

## Name

breakpoint - Setting/deletion of breakpoint

## Input format

**breakpoint** *?options? ?address1? ?address2?*

**breakpoint** *-delete brkno*

**breakpoint** *-enable brkno*

**breakpoint** *-disable brkno*

**breakpoint** *-information*

## Functions

Operates the breakpoint specified by **options** and **address**.

If a breakpoint can be set correctly, the breakpoint number is returned.

The following are **options**:

<b>-software</b>	A software break is specified.
<b>-hardware</b>	A hardware break is specified (default).
<b>-execute</b>	The <b>address</b> execution break is set (default).
<b>-beforeexecute</b>	The break before <b>address</b> execution is set (default : RCU0, RCU1).
<b>-read</b>	An <b>address</b> data read break is set.
<b>-write</b>	An <b>address</b> data write break is set.
<b>-access</b>	An <b>address</b> data access break is set.
<b>-size size</b>	The access size is set (8, 16, or 32).
<b>-data value</b>	The data condition is set.
<b>-datamask value</b>	The data mask is set.
<b>-information</b>	The list of breakpoints is displayed.
<b>-delete</b>	The breakpoint whose number is specified is deleted.

<b>-disable</b>	The breakpoint whose number is specified is disabled.
<b>-enable</b>	The breakpoint whose number is specified is enabled.

---

**Usage example**

---

```
(IDCON) 1 % breakpoint main
1
(IDCON) 2 % breakpoint -i
1 Brk00001 enable rammon.c#17

(IDCON) 3 % breakpoint -software sub
2
(IDCON) 4 % breakpoint -i
1 Brk00001 enable rammon.c#17
2 Brk00001 enable rammon.c#8

(IDCON) 5 % breakpoint -disable 2
(IDCON) 6 % breakpoint -i
1 Brk00001 enable rammon.c#17
2 Brk00001 disable rammon.c#8

(IDCON) 7 % breakpoint -delete 1
2 Brk00001 disable rammon.c#8
```

## cache

---

---

- Name
- Input format
- Functions
- Usage example

---

### Name

---

cache - Setting of cache

---

### Input format

---

#### cache

**cache** *config* *?-icache itype? ?-dcache dtype?*

**cache** *clear* *?-icache bool? ?-dcache bool?*

---

### Functions

---

When **config** is specified for the subcommand, the cache type is set.

When **clear** is specified for the subcommand, whether cache clear is to be done by EXEC, or not, is set (default is clear)

When subcommand or below is omitted, the current status is displayed.

**itype** is selected from the following:

NB85E212	NB85E212 is used.
NB85E213	NB85E213 is used.
nouse	Not used (default).

**dtype** is selected from the following:

NB85E252	NB85E255 is used.
NB85E263	NB85E263 is used.
nouse	Not used (default).

**Usage example**

---

```
(IDCON) 1 % cache config -i NB85E212 -d NB85E252
(IDCON) 2 % cache
i-cache: NB85E212
d-cache: NB85E252
(IDCON) 3 % cache clear -i false
(IDCON) 4 % cache
i-cache: NB85E212 (persist)
d-cache: nouse
```

## dbgexit

---

---

- Name
- Input format
- Functions
- Usage example

### Name

---

dbgexit - Terminating ID850NWC

### Input format

---

**dbgexit** *?options?*

### Functions

---

Terminates the ID850NWC.

The following are *options*:

<b>-saveprj</b>	Project is saved on terminating ID850NWC.
-----------------	---

### Usage example

---

(IDCON) 1 % dbgexit -saveprj

## download

---

---

- Name
- Input format
- Functions
- Usage example

---

### Name

download - Download of files

---

### Input format

**download** *?options? filename ?offset?*

---

### Functions

Downloads file specified by **filename** according to **options**.

If **offset** is specified, the address is shifted by the **offset** (if the data is in binary format, the load start address is specified for **offset**).

<b>-binary</b>	Binary format data is downloaded.
<b>-append</b>	Additional download is executed.
<b>-nosymbol</b>	Download is executed. Symbol information is not read.
<b>-reset</b>	CPU is reset after download.
<b>-information</b>	Download information is displayed.
<b>-symbolonly</b>	Symbol information is read.
<b>-erase</b>	The contents of the internal flash memory are erased all before download (only a product with internal flash memory). In default condition, they are not erased.

---

### Usage example

(IDCON) 1 % download test.lmf

## extwin

---

---

- Name
- Input format
- Functions
- Usage example

---

### Name

extwin - Creation of expansion window

---

### Input format

**extwin** *scriptfile*

---

### Functions

Creates expansion window with ***scriptfile***.

---

### Usage example

(IDCON) 1 % extwin d:/foo.tcl



## finish

---

---

- Name
- Input format
- Functions
- Usage example

### **Name**

---

finish - Returning from function

### **Input format**

---

**finish**

### **Functions**

---

Executes until it returns to the program that called the current function.

### **Usage example**

---

(IDCON) 1 % finish

## go

---

---

- Name
- Input format
- Functions
- Usage example

---

### Name

---

go - continuous execution

---

### Input format

---

**go** *?options?*

---

### Functions

---

Executes the program continuously. When -waitbreak is specified, it waits for the program to stop.

The following are **options**:

- |                     |  |
|---------------------|--|
| <b>-ignorebreak</b> | Breakpoint is ignored.                     |
| <b>-waitbreak</b>   | The command waits for the program to stop. |

---

### Usage example

---

(IDCON) 1 % go -w

# help

---

---

- Name
- Input format
- Functions
- Usage example

## **Name**

---

help - Display of help

## **Input format**

---

**help**

## **Functions**

---

Displays Dcl help.

## **Usage example**

---

(IDCON) 1 % help

## hook

---

---

- Name
- Input format
- Functions
- Usage example

### Name

---

hook - Setting of hook

### Input format

---

**hook** *scriptfile*

### Functions

---

Sets the procedure for hook with ***scriptfile***.

### Usage example

---

(IDCON) 1 % hook d:/foo.tcl

## ie

---

---

- Name
- Input format
- Functions
- Usage example

---

### Name

ie - Display/setting of IE register

---

### Input format

**ie** *reg address ?value?*

**ie** *dcu address ?value?*

---

### Functions

The **ie** command depends on the IE.

When **reg** is specified for the subcommand, referencing and setting of the IE register is executed.

When **dcu** is specified for the subcommand, referencing and setting of the IE register is executed.

**Caution** The value of a register will be reset by 0 if a DCU register is referred to.

---

### Usage example

(IDCON) 1 % ie reg 0x100 1

(IDCON) 2 % ie dcu 0x100 1

# jump

---

---

- Name
- Input format
- Functions
- Usage example

---

## Name

jump - Jump to window

---

## Input format

**jump** *-source -line filename ?line?*

**jump** *?options? address*

---

## Functions

Displays the window specified by *options*.

<b>-source</b>	The Source window is displayed from the address specified by <b>address</b> .
<b>-assemble</b>	The Assemble window is displayed from the address specified by <b>address</b> .
<b>-memory</b>	The Memory window is displayed from the address specified by <b>address</b> .
<b>-line</b>	The command is moved to the line specified by <i>line</i> .

---

## Usage example

```
(IDCON) 1 % jump -s main
(IDCON) 2 % jump -s -l mainfile.c 10
(IDCON) 3 % jump -m array
```

# map

- Name
- Input format
- Functions
- Usage example

## Name

map - Setting/deletion of memory mapping

## Input format

**map** *options address1 address2 ?accsize?*

## Functions

Sets, deletes, and displays memory mapping.

The access size of 8, 16, or 32 is specified by **accsize** (the default is 8).

The following are **options**:

<b>-target</b>	Target area is mapped.
<b>-protect</b>	I/O protect area is mapped.
<b>-clear</b>	All the settings for the mapping are deleted.
<b>-information</b>	Refer to the setting for the mapping.

## Usage example

```
(IDCON) 1 % map -i
1: 0 0x7fff 8 {IROM}
2: 0x8000 0x87ff 8 {Target RRM}
3: 0x8800 0x9fff 8 {Target}
4: 0xa000 0xf7ff 8 {NonMap}
5: 0xf800 0xfaff - {NonMap}
6: 0xfb00 0xfedf 8 {Saddr}
7: 0xfef0 0xfeff 8 {Register}
8: 0xff00 0xffff 8 {SFR}
(IDCON) 2 % map -c
```

## mdi

---

---

- Name
- Input format
- Functions
- Usage example

---

### Name

mdi - Setting of expansion window

---

### Input format

**mdi** geometry *?x y? width height*

**mdi** title *string*

---

### Functions

Sets the size and title name of the expansion window.

The command can be used only from the expansion window.

---

### Usage example

(IDCON) 1 % mdi geometry 0 0 100 100

(IDCON) 2 % mdi title foo



## memory

---

---

- Name
- Input format
- Functions
- Usage example

---

### Name

memory - Display/setting of memory

---

### Input format

**memory** *?options? address ?value?*

**memory** *?options? -fill address1 address2 value*

**memory** *?options? -copy address1 address2 address3*

---

### Functions

Sets **value** in the memory of the **address** specified by address according to **options**.

If **value** is omitted, display the value of the memory of the address specified by **address**.

If **-fill** is specified, value is filled from **address1** to **address2**.

If **-copy** is specified, data from **address1** to **address2** is copied to **address3**.

The following are **options**:

<b>-byte</b>	Displayed/set in one-byte units (default).
<b>-halfword</b>	Displayed/set in halfword units.
<b>-word</b>	Displayed/set in word units.
<b>-fill</b>	The data is filled in.
<b>-copy</b>	The data is copied.
<b>-noverify</b>	Verification is not executed on writing.

---

### Usage example

```
(IDCON) 1 % memory 100
0x10
(IDCON) 2 % memory 100 2
(IDCON) 3 % memory 100
0x02
(IDCON) 4 % memory -fill 0 1ff 0
```

## module

---

---

- Name
- Input format
- Functions
- Usage example

---

### Name

---

module - Display of the list of files and functions

---

### Input format

---

**module** *prognam* *?filename?*

---

### Functions

---

Displays the list of files and functions of the load module specified by ***prognam***.

If ***filename*** is not specified, the list of files is displayed.

If ***filename*** is specified, the list of functions of the specified files is displayed.

---

### Usage example

---

```
(IDCON) 1 % module rammon.lmf
1: rammon.c
(IDCON) 2 % module rammon.lmf rammon.c
1: rammon.c sub1
2: rammon.c main
```

## next

---

---

- Name
- Input format
- Functions
- Usage example

---

### Name

next - Procedure step

---

### Input format

**next** *?options?*

---

### Functions

Executes the procedure steps. If functions are called, the step stops after executing function.

The following are *options*:

<b>-source</b>	Command is executed in source line units (default).
<b>-instruction</b>	Command is executed in command units.

---

### Usage example

(IDCON) 1 % next -i  
(IDCON) 2 % next -s

## refresh

---

---

- Name
- Input format
- Functions
- Usage example

### Name

---

refresh - Redrawing of window

### Input format

---

**refresh**

### Functions

---

Redraws the window and updates the data.

### Usage example

---

```
(IDCON) 1 % batch foo.tcl
(IDCON) 2 % refresh
```

## register

---

---

- Name
- Input format
- Functions
- Usage example

---

### Name

---

register - Display/setting of register value and I/O register value

---

### Input format

---

**register** *?options? regname ?value?*

---

### Functions

---

Sets **value** in the register specified with **regname**.

If **value** is omitted, displays the value of the register specified by **regname**.

The following are **options**:

**-force**                      Compulsory reading is executed.

---

### Usage example

---

(IDCON) 1 % register pc  
0x100  
(IDCON) 2 % register pc 200  
(IDCON) 3 % register pc  
0x200

## reset

---

---

- Name
- Input format
- Functions
- Usage example

---

### Name

reset - Reset

---

### Input format

**reset** *?options?*

---

### Functions

Resets the CPU, ID850NWC or symbols.

If ***options*** are omitted, the CPU is reset.

The following are ***options***:

<b>-cpu</b>	CPU is reset (default).
<b>-debugger</b>	ID850NWC is reset.
<b>-symbol</b>	Symbol is reset.
<b>-event</b>	Event is reset.

---

### Usage example

(IDCON) 1 % reset

## run

---

---

- Name
- Input format
- Functions
- Usage example

---

### Name

run - Reset and execution of CPU

---

### Input format

`run ?options?`

---

### Functions

Resets the program and executes it.

If **-waitbreak** is not specified, the command does not wait until the program stops.

The following are **options**:

<b>-waitbreak</b>	The command waits for the program to stop.
-------------------	--

---

### Usage example

(IDCON) 1 % run  
(IDCON) 2 % run -w

## step

---

---

- Name
- Input format
- Functions
- Usage example

---

### Name

step - Step execution

---

### Input format

**step** *?options?*

---

### Functions

Executes step execution.

If functions are called, the command stops at the head of the functions.

The following are **options**:

- |                     |   |
|---------------------|---|
| <b>-source</b>      | The command is executed in source line units (default). |
| <b>-instruction</b> | The command is executed in command units.               |

---

### Usage example

(IDCON) 1 % step -i  
(IDCON) 2 % step -s



# stop

---

---

- Name
- Input format
- Functions
- Usage example

## Name

---

stop - Stop executing

## Input format

---

**stop**

## Functions

---

Stops the program forcibly.

## Usage example

---

(IDCON) 1 % run  
(IDCON) 2 % stop

# upload

---

---

- Name
- Input format
- Functions
- Usage example

---

## Name

upload - Upload

---

## Input format

**upload** *?options? filename address1 address2*

---

## Functions

Saves the memory data within the specified range in a file.

The following are **options**:

<b>-binary</b>	The data is saved in binary format.
<b>-intel</b>	The data is saved in Intel HEX format (default).
<b>-motorola</b>	The data is saved in Motorola HEX format.
<b>-tektronix</b>	The data is saved in Tektronix HEX format.
<b>-force</b>	The file is overwritten.

---

## Usage example

```
(IDCON) 1 % upload -b foo.hex 0 0xffff
(IDCON) 2 % upload -c -f foo.cov 0 0xffff
```

## version

---

---

- Name
- Input format
- Functions
- Usage example

---

### Name

version - Display of the version information

---

### Input format

**version**

---

### Functions

Displays the version information of the ID850NWC.

---

### Usage example

```
(IDCON) 1 % version
GUI           : E2.00y [31-May-99]
Devicefile    : 78K0[uPD780034] E1.01a
Debugger      : 78K/0 Debugger E2.50c [02-Apr-99]
Executer      : 78K/0 Executer E1.3c
Packet translator : 78K/0 Packet E2.00w
Assembler     : 78K/0 Asm/Disasm E1.15a [01-Apr-99]
Tcl/Tk        : 8.1.1
```

## watch

---

---

- Name
- Input format
- Functions
- Usage example

---

### Name

watch - Display/setting of variables

---

### Input format

**watch** *?options? variable ?value?*

---

### Functions

Displays and sets the variables.

The following are *options*:

<b>-binary</b>	The value is displayed in binary digits.
<b>-octal</b>	The value is displayed in octal digits.
<b>-decimal</b>	The value is displayed in decimal digits.
<b>-hexdecimal</b>	The value is displayed in hexadecimal digits.
<b>-string</b>	The value is displayed in character strings.
<b>-sizeof</b>	The size, instead of the value, of variables is displayed in decimal digits.

---

### Usage example

```
(IDCON) 1 % watch var
0x10
(IDCON) 2 % watch -d var
16
(IDCON) 3 % watch array\[0\] 0xa
```

## where

---

---

- Name
- Input format
- Functions
- Usage example

### Name

---

where - Trace of stack

### Input format

---

**where**

### Functions

---

Executes the back-trace of the stack.

### Usage example

---

```
(IDCON) 1 % where
1: test2.c#sub2(int i)#13
2: test.c#num(int i)#71
3: test.c#main()#82
```

## wish

---

---

- Name
- Input format
- Functions
- Usage example

---

### Name

wish - Startup of Tclet

---

### Input format

**wish** *scriptname*

---

### Functions

Starts up the script using Tk (Tclet).

The expansion window can be created with Tclet.

---

### Usage example

(IDCON) 1 % wish test.tcl

## APPENDIX A N-WIRE CHECKER

This chapter explains the following item about N-Wire Checker.

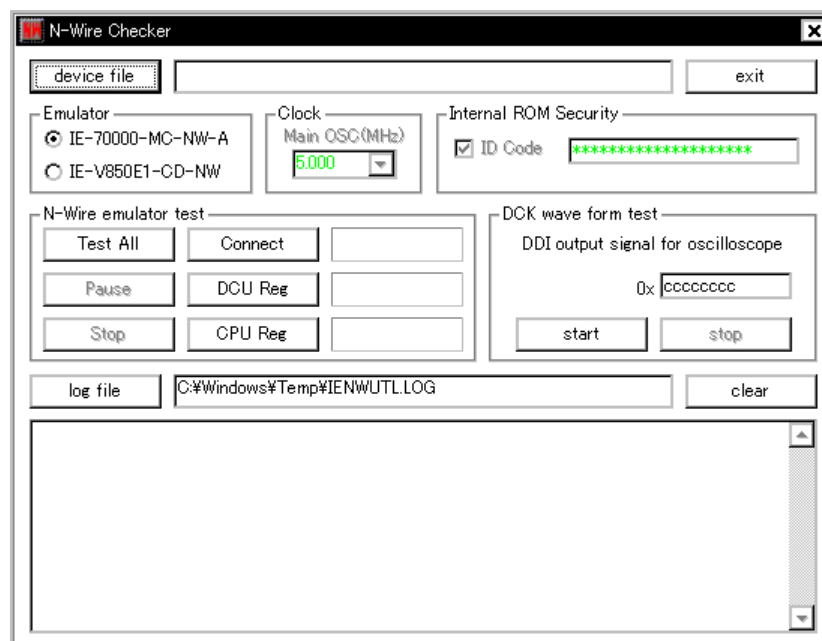
- Overview
- Operation
- Explanation of each area and button
- Corrective Actions in Case of NG
- Restrictions

### A.1 Overview

N-Wire Checker is an application for simply testing the PC connected N-Wire emulator and the target CPU operation.

N-Wire Checker is installed at the same time as the ID850NWC.

Figure A-1 N-Wire Checker



The following tests are possible with N-Wire Checker.

**(1) N-Wire Emulator test**

Checks N-Wire emulator connection, initialization, and target CPU register read/write.

The test results can be output to the log viewer and a log file.

**(2) DCK wave form test**

Outputs the data set from the N-Wire emulator to DDI signal in order to check the waveform of the DCK signal on the target board with an oscilloscope.

## **A.2 Operation**

**(1) Starting and preparation N-Wire Checker**

- 1 A N-Wire emulator and a target board are turned on.  
Please refer to the user's manual of each N-Wire emulator about connection of a N-Wire emulator, a target board, and a host machine.
- 2 Select [Start] menu->[Program]->[NEC Tools32]->[N-Wire Checker]. This starts N-Wire Checker.  
If, at this time, the ID850NWC is already started up, end it.
- 3 Click the <device file> button. Specified a device file name of target CPU.
- 4 In the Emulator area, select the N-Wire emulator that is connected (IE-70000-MC-NW-A or IE-V850E1-CD-NW).
- 5 In the Clock area, specify the frequency of the main clock input to the device (only when the IE-V850E1-CD-NW is connected).
- 6 In the Internal ROM Security area, check the check box and input an ID code when a product with security unit is used in the single-chip mode (only when the IE-V850E1-CD-NW is connected).

**(2) N-Wire Emulator test**

- 1 Press the <Test All> button. This is the same operation as pressing the <Connect>, <DCU reg>, and <CPU reg> buttons in sequence.
- 2 The test results (OK, NG) and log are displayed.  
Check that all the test are OK.  
If NG is displayed, refer to "A.4 Corrective Actions in Case of NG".  
For details of the dialog box, refer to "A.3 Explanation of each area and button".



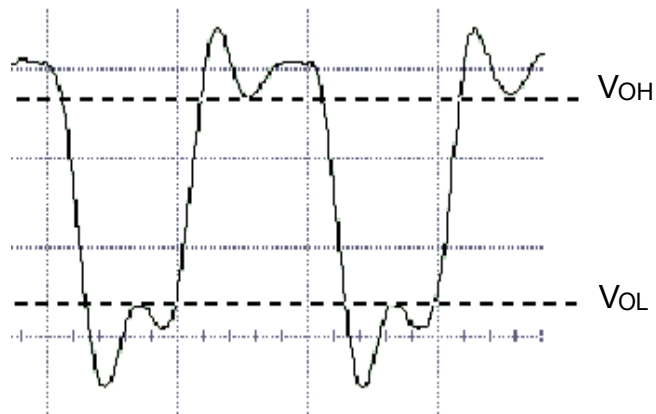
**(3) DCK waveform test**

- 1 Connect the DCK signal of the target board to the oscilloscope probe.

**Caution** Switch off the power supply of the N-wire emulator and target board before performing this connection.

- 2 Input 4-byte data (0xffffffff at the time of shipping) in the DDI output signal for oscilloscope area and then click the <start> button.
- 3 Monitor the waveform on the oscilloscope.
- 4 Check whether the waveform noise (reflection) is within the standard range below.  
If it is not within the standard range "A.4 Corrective Actions in Case of NG".  
For details of the dialog box, refer to "A.3 Explanation of each area and button".

Figure A-2 Waveform of N-Wire Checker



VOH, VOL: Refer to the data sheet supplied with the emulator.

**Caution** The data is output from the lowermost bit.

**-In the IE-70000-MC-NW-A**

If data from the target board changes, the changed data is subsequently output because data is shifted via the target board.

**-In the IE-V850E1-CD-NW**

Even if data from the target board changes, the same data is always output because data is shifted inside the emulator.

### A.3 Explanation of each area and button

#### (1) device file (operating environment setting area)

Object	Function
<device file>button	Displays the dialog box for specifying the device file name of the target CPU.
Text area	Displays the device file name of the target CPU and its path. Specify this at N-Wire Checker startup.

#### (2) Emulator(emulator selecting area)

Object	Function
IE-70000-MC-NW-A	Select this to connect IE-70000-MC-NW-A.
IE-V850E1-CD-NW	Select this to connect IE-V850E1-CD-NW.

#### (3) Clock(frequency of the main clock setting area)

Object	Function
Main OSC (MH z )	Input the frequency of the main clock input to the device.

#### (4) Internal ROM Security (ID code specification area)

Object	Function
ID Code	Check the check box when a product with security unit is used in the single-chip mode.
Input area	Input an ID code when a product with security unit is used in the single-chip mode. For details of the ID code, refer to the "IE-V850E1-CD-NW User's Manual".

#### (5) N-Wire emulator test (N-Wire emulator test area)

Object	Function
<Test All>button	Tests Connect, DCR Reg, and CPU Reg in sequence.
<Pause>button	Pauses the test. (Testing resumes when this button is clicked again.)
<Stop>button	Stops the test.
<Connect>button	Opens communication with the target and executes testing for initialization.
<DCU Reg>button	Performs testing for DCU register read/write.
<CPU Reg>button	Executes testing for CPU register read/write.
Text area	A test result is displayed.

**(6) DCK wave form test (DCK waveform test area)**

Object	Function
DDI output signal for oscilloscope area	Sets the data of the DDI signal output from the N-Wire emulator. When a value of 0xffffffff or greater is set, an error occurs when the <start> button is pressed.
<start>button	Outputs the value (0 to 0xffffffff) specified in the DDI output signal for oscilloscope text area.
<stop>button	Stops the signal for value testing specified in the DDI output signal for oscilloscope text area.

**(7) Log viewer area**

Object	Function
<log file> button	Displays the dialog box for specifying the test result log file name. (Window\TEMP\IENWULT.log at the time of shipping) The log file is overwritten when N-Wire Checker is started up. The following occurs when the file path name is not specified. - When environment variable TEMP is not defined -> A log file is created under environment variable TMP. - When environment variable TMP is not defined -> A log file is created in the current folder.
Text area	Displays the log file name and its path.
<clear>button	Clears the log viewer display. If a log file name is specified in the log file text area, the log file contents are not cleared.
Log viewer area	Displays the test results as a log (up to approx. 21,000 characters). The same contents are output to the log file specified in the log file text area.

**(8) Other**

Object	Function
<exit> button	Ends N-Wire Checker.

## A.4 Corrective Actions in Case of NG

### (1) NG in N-Wire Emulator test

#### If [Connect] is NG

Take action to correct the error by referring to Table A-1 .

Table A-1 Errors(N-Wire Checker)

Error Number	Error Messages	Corrective Action
0x0100	Driver cannot be opened.	The driver may not be correctly installed. Reinstall the driver.
0x0c43	Connection of emulator cannot be performed.	<b>In the IE-70000-MC-NW-A</b> Check the power to the in-circuit emulator. Or, the PC-IF cable may not be correctly connected. Check the connection. <b>In the IE-V850E1-CD-NW</b> The switch setting may be wrong if a desktop computer is used and two or more PC cards are inserted. Check the setting. Or it may have malfunctioned.
0x03a0	Power supply of target is OFF.	Check the target power supply. Check the cable connecting the in-circuit emulator and target board. Check that the VDD signal is input to the connector of the target board.
0x0c70	DCU cannot be accessed.	Check the power to the chip. Check the connection of the signal lines (DCK, DMS, DDI, DDO, and DRSTZ). Check the noise level by a DCK waveform test.
0x0105	Failed in reading device file (dxxxx.800).	The necessary file may be damaged. Reinstall the device file.
0x0c71	Reset cannot be performed.	Check the clock signal. This may be caused by a stopped clock or a slow clock.
0x0c24	It cannot shift to debug mode.	Check the clock signal. This may be caused by a stopped clock or a slow clock.
0x0c72	Monitor memory cannot be accessed.	Check the noise level by a DCK waveform test. The problem may also be caused by an internal chip problem.
0x0c73	Monitor execution cannot be performed.	Check the noise level by a DCK waveform test. The problem may also be caused by an internal chip problem.
0x0c74	CPU register cannot be accessed.	Check the noise level by a DCK waveform test. The problem may also be caused by an internal chip problem.
0xf603	Incorrect ID Code.	Input the correct ID code (IE-V850E1-CD-NW).
0xf604	Incorrect ID Code. Abort the N-Wire checker.	Input the correct ID code (IE-V850E1-CD-NW).

#### If [DCU reg] is NG

Normal access to the DCU register seems impossible.

Perform the DCK waveform test.

If no anomaly is found with the DCK waveform test, the register may be damaged, so try again with a different chip .

**If [CPU reg] is NG**

Normal access to the CPU register seems impossible.

Perform the DCK waveform test.

If no anomaly is found with the DCK waveform test, the register may be damaged, so try again with a different chip.

**(2) NG in DCK waveform test**

Check the following items regarding the design of the target board, in accordance with the User's Manual of the N-Wire emulator.

**In the IE-70000-MC-NW-A**

- Is the pattern of the N-Wire interface signal 100 mm or less?
- Is the DCK signal shielded through a 4.7 k Ohm pull-up resistor + GND?
- Is a 4.7 k Ohm pull-up resistor connected to the DMS signal and the DDI signal?
- Is a 22 Ohm damping resistor connected to the DDO signal?
- Is a 50 k Ohm pull-down resistor connected to the DRSTZ signal?

**In the IE-V850E1-CD-NW**

- Is the pattern of the N-Wire interface signal 100 mm or less?
- Is the DCK signal shielded through a pull-up resistor\* + GND?
- Is a pull-up resistor\* connected to the DMS, DDI, and DDO signals?
- Is a damping resistor\* connected to the DDO signal?
- Is a pull-down resistor\* connected to the \_DRSTZ signal (or DRST signal)?

**Caution** The resistance of the resistor marked \* above must be as specified for each device. Some devices have on-chip pull-up/down resistors. With these devices, an external resistor is not necessary.

Also check the following items regarding the power supply capacity of the target board.

- Is a power supply with a sufficient capacity used?
- Are sufficient capacitors provided on the target board?
- Is a bypass capacitor connected to all the power supply pins of the CPU?
- Is the GND plane on the target board sufficiently large?

**A.5 Restrictions**

Up to 21000 characters can be displayed in the log view area. If this is exceeded, the displayed information is overwritten starting from the oldest data. If the system resources are running extremely short, the N-Wire Checker stops output to the log view area. For accurate information, therefore, use the log file. Information is continuously recorded in the log file even while screen output is stopped. The log file is overwritten once when the N-Wire checker is started.

## APPENDIX B EXPANSION WINDOW

This chapter explains the following item about Expansion Window.

- Overview
- Activation
- Explanation of each sample window

### B.1 Overview

With the ID850NWC, the user can create custom windows in addition to the existing windows.

The Tcl (Tool Command Language) interpreter and the commands for controlling the debugger are implemented in the ID850NWC. Users can create windows using this Tcl.

The ID850NWC is supplied with samples of the following expansion windows.

Table B-1 List of Expansion Window

Window Name	Function
List window	Displays a list of the source files and functions.
Grep window	Searches a character string.
RRM window	This is the memory window for real-time RAM monitoring.
Hook window	Sets the hook procedure.
Memory Mapped I/O window	Writes to or reads from the specified address only.

### B.2 Activation

The expansion window can be activated by selecting List, Grep, RRM, Hook, or Memory in [Others] on the [Browse] menu.

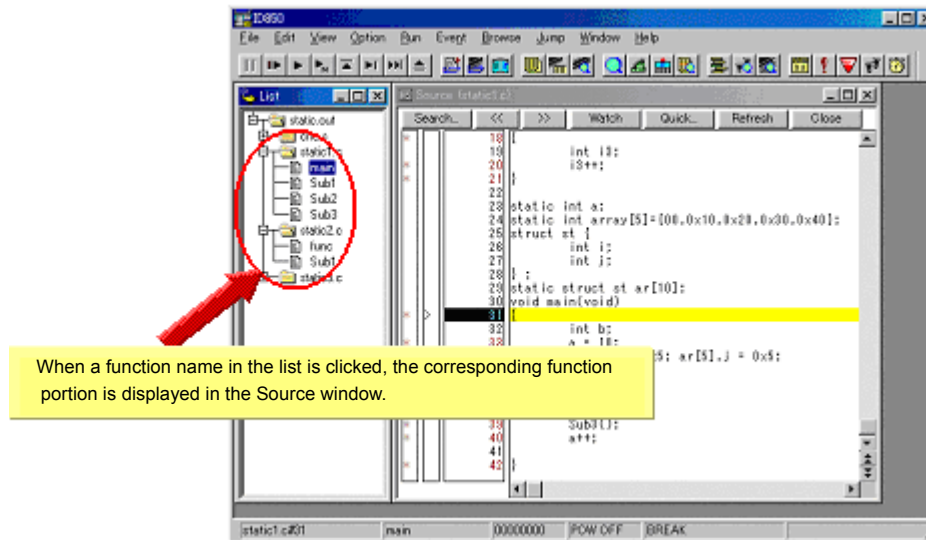
**Remark** Each .tcl file is installed in NECTools32\BIN\idctl\tools.

### B.3 Explanation of each sample window

## List window

The lists of the source files and functions are displayed in a tree format in this window. When a function name in the list is clicked, the corresponding source is displayed.

Figure B-1 List Window

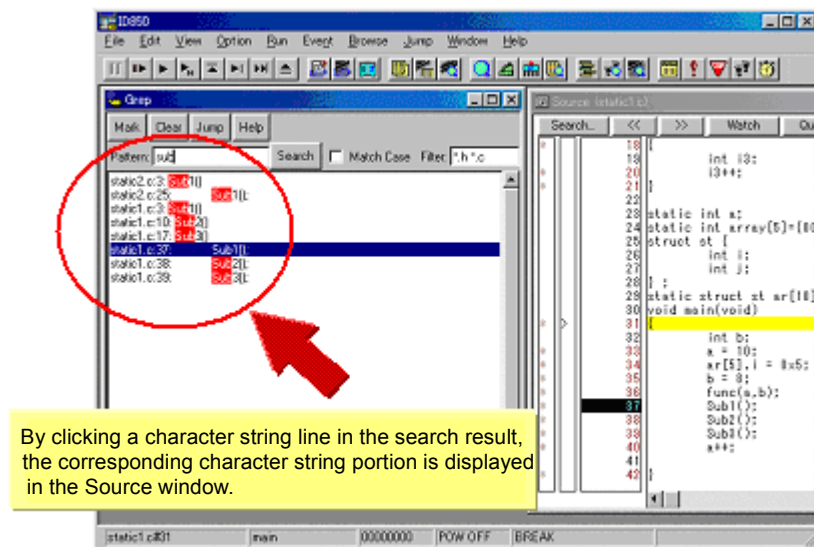


## Grep window

Search for a character string is performed in the files under the source path.

When the search result is clicked, the corresponding source is displayed.

Figure B-2 Grep Window



Object	Function
Pattern	Input the character string to be searched.
<Mark>button	Marks the searched character string.
<Clear>button	Clears the marking
<Jump>button	Put the cursor on a section in the search result and click this button to open the corresponding file
Match Case	Select whether or not to distinguish uppercase and lowercase.
Filter	Specify the type of the file to be searched.



## RRM window

This is a dedicated window for RAM monitoring.

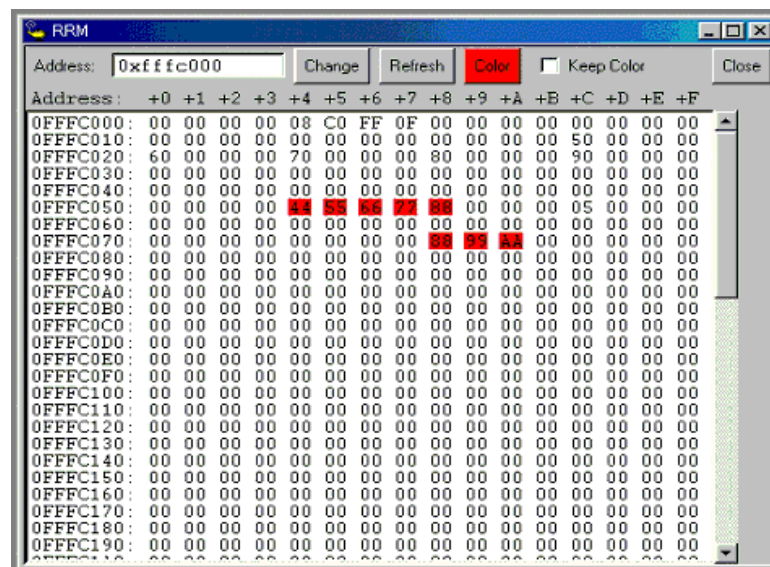
The address area in which a value was changed in the RAM area during program execution is highlighted with a color. The display range is 1 KB. With reading RAM, execution of the user program momentarily breaks. On this window, the start address of the RAM area can be changed while the user program is being executed.

**Caution 1** All data are not read at the same time (because data of 1 KB is divided and read in word units).

**Caution 2** The sampling interval is about 0.3 - 0.7 seconds (20MHz), but it depend on the frequency of the CPU operation.

**Caution 3** This RRM window is opened even when the RAM monitor function is set to OFF in the Extended Option dialog box.

Figure B-3 RRM Window



Object	Function
Address	Input the start address to be displayed (automatically aligned to 1 KB.)
<Change>button	Switch the start address display.
<Refresh>button	Reads data from the memory
<Color>button	The color can be customized. The default color is red.
Keep Color	Specify whether or not to hold the color highlighting. Selected: Once a value is changed, the color highlighting is held until a break occurs. Unselected: The color is cleared if there is no change of values(default).
<Close>button	Closes this window.

## Hook window

This window is used to set a hook to the debugger, using a hook procedure.

The hook procedure enables changing the register value before downloading a program, or after a CPU reset.

On this window, a hook can be set by using the following four tabs.

[BeforeDownload] tab: Hook before downloading

[AfterCpuReset] tab: Hook after CPU reset

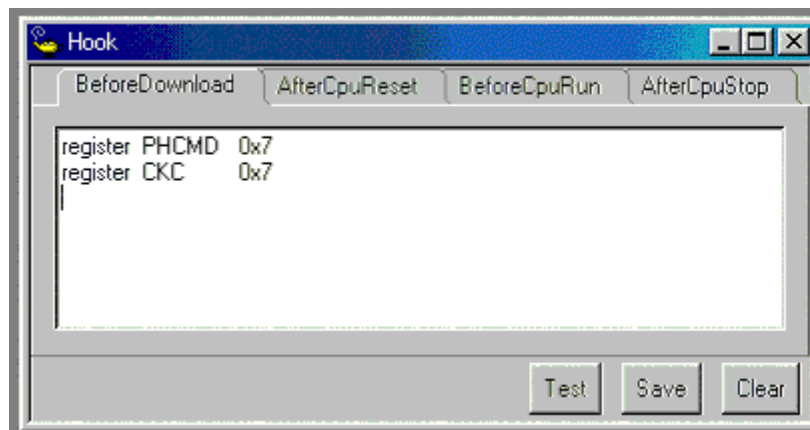
[BeforeCpuRun] tab: Hook before start of execution

[AfterCpuStop] tab: Hook after break

By setting a peripheral I/O register by using the [BeforeDownload] tab before downloading the load module, for example, downloading can be executed at high speeds. Access to the external memory is also facilitated by using this tab.

If the setting is saved as "project-file-name.tcl" in the directory where the project is stored, the setting is executed when the project is next opened.

Figure B-4 Hook Window



Object	Function
[BeforeDownload] tab	Hook before downloading Before downloading is performed, the register values input to the tab are automatically overwritten by the specified value.
[AfterCpuReset] tab	Hook after resetting CPU after resetting CPU, the register values input to the tab are automatically overwritten by the specified value.
[BeforeCpuRun] tab	Hook before starting execution before starting execution, the register values input to the tab are automatically overwritten by the specified value.

Object	Function
[AfterCpuStop] tab	Hook after breaking After breaking, the register values input to the tab are automatically overwritten by the specified value.
<Test> button	All the commands described on the tabs are tested.
<Save>button	Saves all the tab contents to a file. If the ID850NWCwas activated from a project file, the file is saved as "project-file-name.tcl".
<Clear> button	Clears all the descriptions on the tabs.

**Caution** Specify the program register and the peripheral I/O register for the register name.

## Memory Mapped I/O window

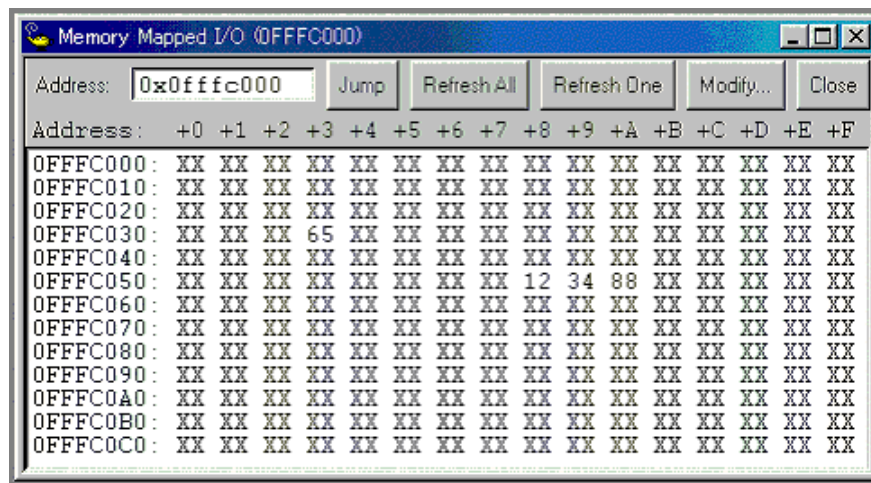
Data is explicitly read or written at a specified address in this window.

When a write is performed in the Memory window, the data is internally read and verified by the ID850NWC. In addition, the memory can also be read simply by scrolling in the Memory window. On the Memory Mapped I/O window, however, the above operations are not performed.

Therefore, this window is useful for reading or writing a specific address.

While the user program is being executed, it momentarily breaks before data is written in this window.

Figure B-5 Memory Mapped I/O Window



Object	Function
Address	Input the address to display. The display target address changes by pressing the Enter key or clicking the <Jump> button. The data contents are not read at this time, so the address (numerical value) is displayed in the address display section, but "XX" is displayed in the data section.
<Jump> button	Jumps to the address input in the Address field.
<Refresh All> button	Reads all the areas currently displayed only once. "ZZ" will be displayed in the data section when an attempt is made to read an unmapped area, or when an error occurs upon a read.
<Refresh One> button	Reads data in the memory of the address at which the cursor is placed only once. The read data size depends on the display format. "ZZ" will be displayed in the data section when an attempt is made to read an unmapped area, or when an error occurs upon a read.
<Modify...>button	Opens the Memory Mapped I/O dialog box. The address at which the cursor is placed is the input address displayed in the Memory Mapped I/O dialog box. If this button is clicked after the cursor position is changed in the Memory Mapped I/O dialog box, the Address field in the Memory Mapped I/O dialog box is also changed.
<Close>button	Closes this window.
Right-click menu	Select the display format from Byte, HalfWord, and Word.

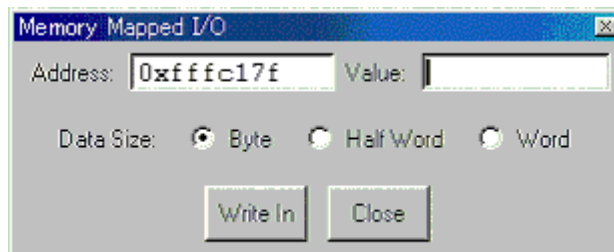
## Memory Mapped I/O dialog box

This dialog box can be opened by clicking the <Modify...> button on the Memory Mapped I/O window. It is used to write data to any address.

**Caution 1** When the area to which data is written is displayed in the Memory window or Watch window, data is read in these windows after the <Write in> button is clicked.

**Caution 2** If Data Size is less than Access Size specified in the Configuration dialog box, ID850NWC reads data in Access Size once, changes the corresponding part of the read data, and writes the changed data in Access Size.

Figure B-6 Memory Mapped I/O Dialog Box



Object	Function
Address	Input the address to be written. The address corresponding to the data for which the cursor is placed in the Memory Mapped I/O window is displayed.
Value	Input the value to be written.
Data Size	Select the size of the data to be written. The size specified in the Memory Mapped I/O window is selected by default.
<Write in> button	Data is written to the specified address with the specified size.
<Close> button	Closes this dialog box.

## APPENDIX C INPUT CONVENTIONS

This appendix explains the input conventions of the following items:

- Character set
- Symbols
- Numeric Values
- Expressions and Operators

### C.1 Character set

The characters listed in Table C-1 can be used as the character set.

The characters listed in Table C-2 can be used as special characters.

Specify a line number as an integer constant, starting from 1.

Table C-1 Usable Character Set

Classification	Character
Alphabetic characters	Uppercase characters: A B C D E F G H I J K L M N O P Q R S T U V X W 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 x w z
Numerals	0123456789
Character equivalent to alphabetic character	@_

Table C-2 List of Special Characters

Character	Name	Usage
(	Left parenthesis	Changes operation order.
)	Right parenthesis	Changes operation order.
+	Plus	Addition operator or positive sign
-	Minus	Subtraction operator or negative sign
*	Asterisk	Multiplication operator or indirect reference operator
/	Slash	Division operator
%	Percent	Remainder operator
~	Tilde	Complement operator
	Vertical line	Bit sum operator
^	Circumflex	Bit difference operator
&	Ampersand	Bit product operator or address operator
[	Left bracket	Array subscript operator or base register specification symbol
]	Right bracket	Array subscript operator or base register specification symbol

Table C-2 List of Special Characters

Character	Name	Usage
.	Period	Direct member operator or bit position specifier
,	Comma	Delimiter between operands

## C.2 Symbols

The following rules apply to symbols:

- 1 A symbol consists of characters A to Z, a to z, @, \_ (underbar), . (period), and 0 to 9.
- 2 A symbol must start with a character other than numerals 0 to 9.
- 3 Uppercase characters (A to Z) and lowercase characters (a to z) are distinguished.
- 4 A symbol must be no more than 256 characters long (if a symbol of more than 265 characters is defined, only the first 256 characters are valid).
- 5 A symbol is defined by loading a load module file.
- 6 Symbols are classified into the following types by the valid range:
  - Global symbol (assembly language, C language)
  - Static symbol (C language)
    - In-file static symbol
    - In-function static symbol
  - Local symbol (C language)
    - In-file local symbol
    - In-function local symbol
    - In-block local symbol
- 7 The following symbols are available for each language used:
  - Assembly language
    - Label name
  - C language
    - Variable name (including pointer function name, enumeration type variable name, array name, structure name, and union name)
    - Function name, label name
    - Array element, structure element, union element, bit field (if the symbol is an array, structure, or union)
- 8 A symbol can be described instead of an address or numeric value.
- 9 The valid range of a symbol is determined based on the source debug information when the source file is assembled or compiled.
- 10 Describe only the symbol name of a global symbol.
- 11 A local symbol is expressed in pairs with a file name.

## C.3 Numeric Values

The following four types of numeric values can be used. The input format of each type is as shown

below.

The suffix (**bold**) and the alphabetic characters of hexadecimal numbers may be uppercase or lowercase characters. If the first character is A to F, 0 must be prefixed to it.

In the input field of the debugger, decimal numbers or hexadecimal numbers are alternately selected, depending on the default radix.

Table C-3 Input Format of Numeric Values

numeric value	Input format
Binary number	n <b>Y</b> n...n <b>Y</b> (n=0,1)
Octal number	n <b>O</b> n...n <b>O</b> (n=0,1,2,3,4,5,6,7) n <b>Q</b> n...n <b>Q</b> (n=0,1,2,3,4,5,6,7)
Decimal number	n n...n n <b>T</b> n...n <b>T</b> (n=0,1,2,3,4,5,6,7,8,9)
Hexadecimal numbers	n n...n n <b>H</b> n...n <b>H</b> <b>0xn</b> <b>0xn</b> ...n (n=0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)

## C.4 Expressions and Operators

### Expressions

An expression consists of constants, register names, IOR names, and symbols coupled by operators. If an IOR name, label name, function name, or variable name is described as a symbol, an address is calculated as the value of the symbol.

The elements making up an expression, except operators, are called terms (constants and labels).

Terms are called the first term, the second term, and so on, starting from the left.

### Operators

The following operators of the C language can be used:

Table C-4 List of Operators

Symbol	Meaning	Explanation
<b>Operators</b>		
+	Addition	Returns the sum of the first and second terms.
-	Subtraction	Returns the difference between the first and second terms.
*	Multiplication	Returns the product of the first and second terms.
/	Division	Divides the value of the first term by the value of the second term, and returns the integer of the result.



Table C-4 List of Operators

Symbol	Meaning	Explanation
MOD %	Remainder	Divides the value of the first term by the value of the second term, and returns the remainder of the result.
- sign	Unary operator (negative)	Returns 2's complement of the value of the term.
+ sign	Unary operator (positive)	Returns the value of the term.
<b>Logical operator</b>		
NOT ~	Negation	Logically negates each bit of the term, and returns the result.
AND &	Logical product	Obtains the logical product of the values of the first and second terms on each bit, and returns the result.
OR 	Logical sum	Obtains the logical sum of the values of the first and second terms on each bit, and returns the result.
XOR ^	Exclusive logical sum	Obtains the exclusive logical sum of the values of the first and second terms on each bit, and returns the result.
<b>Shift operator</b>		
SHR >>	Right shift	Shifts the value of the first term by the value (number of bits) of the second term to the right, and returns the result. As many 0s as the number of shifted bits are inserted in the higher bits.
SHL <<	Left shift	Shifts the value of the first term by the value (number of bits) of the second term to the left, and returns the result. As many 0s as the number of shifted bits are inserted in the lower bits.
<b>Byte separation operator</b>		
HIGH	Higher byte	Of the lowest 16 bits of the term, returns the higher 8 bits.
LOW	Lower byte	Of the lowest 16 bits of the term, returns the lower 8 bits.
<b>Word separation operator</b>		
HIGHW	Higher word	Of the 32 bits of the term, returns the higher 16 bits.
LOWW	Lower word	Of the 32 bits of the term, returns the lower 16 bits.
<b>Other</b>		
(	Left parenthesis	Performs the operation in ( ) before the operation outside ( ). '(' and ')' are always used in pairs.
)	Right parenthesis	

**Rules of operation**

Operations are performed according to the priority of the operators.

Table C-5 Operator Priority

Priority	Operators
1 Higher	(, )

Table C-5 Operator Priority

Priority	Operators
2	+ sign, - sign, NOT, , HIGH, LOW, HIGHW, LOWW
3	*, /,, MOD, %, SHR, >>, SHL, <<
4	+, -
5	AND, &
6 Lower	OR,  , XOR, ^

- If the priorities of the operators are the same, the operation is performed from the left toward the right.
- Performs the operation in ( ) before the operation outside ( ).
- Each term in an operation is treated as unsigned 32-bit data.
- All operation results are treated as unsigned 32-bit data.
- If an overflow occurs during operation, the lower 32 bits are valid, and the overflow is not detected.

### Terms

To describe a constant for a term, the following numeric values can be described.

Table C-6 Range of Radixes

Radix	Range
Binary number	0Y <= value <= 11111111111111111111111111111111Y (32 digits)
Octal number	0O <= value <= 37777777777O
Decimal number	-2147483648 <= value <= 4294967295 (A negative decimal number is internally converted into a 2's complement.)
Hexadecimal numbers	0H <= value <= 0FFFFFFFH

## APPENDIX D TERMINOLOGY

The major terms used throughout this manual are as follows:

- Current window
- Delimiter (separator)
- Program
- File
- Function
- Structure
- Stack frame number
- Line

### D. 1 Terminology

#### Current window

The current window is the window you are now in and operating by means of key input and menu selection.

#### Delimiter (separator)

The following separators can be used as the delimiter of file names, function names, variable names, and line numbers.

Table D-1 Delimiter

Delimiter	Meaning
#	Used as a separator of file names, variable names, function names, and line numbers.
\$	Used as a separator of load module names, file names, variable names, and function names.

#### Program

A program is the execution unit to be debugged. The debugger manages each one of the load module files specified on downloading as a program.

To specify a program name, use a load module file name on downloading.

#### Current program

The current program is the program that contains the instruction currently breaking execution (instruction indicated by the PC).

When specifying a file in the current program, the program name may be omitted.

#### File

Up to 127 characters can be handled in a file name with path.

Depending on the type of file, however, a file name of more than 127 characters can be specified.

**Current file**

The current file is the source file that contains the instruction currently breaking execution (instruction indicated by the program counter (PC)).

When specifying a line or function in the current file, the file name may be omitted.

**Function**

Indicates the functions constituting a C source program.

**Current function**

The current function is the function that contains the instruction currently breaking execution (instruction indicated by the program counter (PC)).

When accessing a local variable in the current function, specifying the function name may be omitted.

**Structure**

Structures and unions in C language are called structures.

The name structure is used if a structure or union variable is used without members explicitly specified.

**Stack frame number**

The stack frame number is a decimal integer starting from 1 and is assigned to the stack contents. The shallower the nesting level of the stack, the higher the number. A function having a stack number one less than that of a certain function is the function that calls the certain function.

**Line**

A line is specified in order to identify one line in the source file.

Specify a line number as an integer constant, starting from 1.

## APPENDIX E MESSAGES

This appendix explains the following item about messages displayed in the Error/Warning dialog box of the ID850NWC.

- Display format
- Error messages at startup
- Fatal error messages
- Syntax error messages
- Warning message or question message

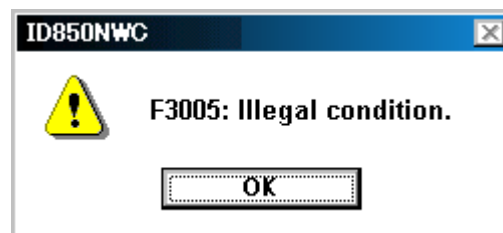
### E.1 Error/Warning Messages

#### E. 1. 1 Display format

Messages from the ID850NWC are generated and output to the Error/Warning dialog box when "information that should be informed to users (fatal error, syntax error, and warning or questions)" is detected while the debugger is executing processing.

The product name is displayed on the title bar of the dialog box.

Figure E-1 Error/Warning Messages



There are three types of messages in the ID850NWC (fatal error message, syntax error message, and warning message or question message).

**A** : Fatal error messages

When a fatal error is detected in the debugger, a message is output to the Error dialog box before aborting the processing and terminating the ID850NWC.

**F** : Syntax error messages

When a syntax error is detected in the ID850NWC, a message is output to the Error dialog box before aborting the processing. At this time, opened windows and dialog boxes are closed.

**W** : Warning message or question message

When a warning or question is detected in the ID850NWC, a message is output to the Warning dialog box before aborting the processing.

**E. 1. 2 Error messages at startup**

The errors listed in next table may be output when the ID850NWC is started up. If such an error is output, fix the problem by following the indicated corrective action.

F0100: Driver cannot be opened.

- 1) The driver may not be correctly installed. Reinstall the driver.

F0c43: Connection of emulator cannot be performed.

- 1) The switch setting may be wrong if a desktop computer is used and two or more PC cards are inserted. Check the setting. Or it may have malfunctioned.

W03a0: Power supply of target is OFF.

- 1) Check the target power supply. Check the cable connecting the in-circuit emulator and target board. Check that the VDD signal is input to the connector of the target board.

F0c70: DCU cannot be accessed.

- 1) Check the power to the chip. Check the connection of the signal lines (DCK, DMS, DDI, DDO, and DRSTZ). Check the noise level by a DCK waveform test on the N-WIRE CHECKER.

F0105: Failed in reading device file (dxxxx.800).

- 1) The necessary file may be damaged. Re-install the device file.

F0c71: Reset cannot be performed.

- 1) Check the clock signal. This may be caused by a stopped clock or a slow clock.

F0c24: It cannot shift to debug mode.

- 1) Check the clock signal. This may be caused by a stopped clock or a slow clock.

F0c72: Monitor memory cannot be accessed.

- 1) Check the noise level by a DCK waveform test on the N-WIRE CHECKER. The problem may also be caused by an internal chip problem.

F0c73: Monitor execution cannot be performed.

- 1) Check the noise level by a DCK waveform test on the N-WIRE CHECKER. The problem may also be caused by an internal chip problem.

F0c74: CPU register cannot be accessed.

- 1) Check the noise level by a DCK waveform test on the N-WIRE CHECKER. The problem may also be caused by an internal chip problem.

F0c23: Bus hold under continuation.

- 1) Check the setting of the target board, or mask the HOLD pin.

### **E. 1. 3 Fatal error messages**

A list of the messages that are output when fatal errors are detected while the ID850NWC is executing processing is shown below in numerical order.

A0101: Can not find initialization file (expc.ini).

A0102: Host name not found.

A0105: Failed in reading device file (d3xxx.800).

- 1) Necessary files may be damaged. Reinstall the device file.

A0106: Illegal data received.

- 1) Check the power of the in-circuit emulator, cable connections, and setting of the interface board and restart the ID850NWC.

A01a1: Failed in reading ie703000.ie.

A01a2: Break board is not connected.

A01a3: Emulation board is not connected.

A01a4: Board configuration of ICE is not consistent.

A01a5: POD/EM1 board is not connected.

A01a6: Executor is running.

A01a8: Failed in reading initialization file (expc.ini).

A04a1: Not enough memory for emulation.

A0600: Not enough memory for buffer.

- 1) There is not enough system memory. Close the applications being executed and the open files.

A0c01: During access of register, CPU did timeout.

- 1) Check the clock signal, etc. The register value may not be correct.

A0c02: During access of memory, CPU did timeout.

1) Check the HOLD signal, WAIT signal, clock signal, etc. The memory value may not be correct.

A0c03: During access of I/O register, CPU did timeout.

1) Check the HOLD signal, WAIT signal, clock signal, etc. The IOR value may not be correct.

A1000: Failed in initializing ICE.

A1001: No entry exists for specified number.

A1002: Can not relocate internal RAM.

A1005: Illegal relocation address.

A1007: Not enough memory on ICE.

A1008: Not enough memory for tables.

1) There is not enough system memory. Close the applications being executed and the open files.

A1009: Already initialized.

A100a: Not initialized.

A1015: Programmable-IOR does not exist.

A10ff: Can not communicate with ICE.

A1dbe: Error occurred inside debugger.

A2001: Illegal address.

A2009: Device file is damaged or error is in file.

A200b: Can not copy.

A200c: Not enough memory.

1) There is not enough system memory. Close the applications being executed and the open files.

A20ff: Can not communicate with ICE.

A2222: Illegal condition.



A3012: Not enough memory.

- 1) There is not enough system memory. Close the applications being executed and the open files.

A30ff: Can not communicate with ICE.

A4011: Not enough memory.

- 1) There is not enough system memory. Close the applications being executed and the open files.

A4018: Not enough memory.

- 1) There is not enough system memory. Close the applications being executed and the open files.

A5000: Illegal device file type.

A5001: Not enough memory.

- 1) There is not enough system memory. Close the applications being executed and the open files.

A5002: Can not open device file.

A5003: Reading of device file went wrong.

A5004: Can not close device file.

A5005: Illegal device file format.

- 1) Necessary files may be damaged. Reinstall the device file.

A5006: Failed in initializing ICE.

A5007: Device file has broken or error is in a file.

A500c: Failed in reading expc.ini.

A500d: Not enough memory.

- 1) There is not enough system memory. Close the applications being executed and the open files.

A5300: Illegal device file type.

A5301: Not enough memory.

- 1) There is not enough system memory. Close the applications being executed and the open files.

A5302: Can not open database file.

- 1) Necessary files may be damaged. Reinstall the ID850NWC and device file.

A5303: Reading of database file went wrong.

A5304: Can not close database file.

A5305: Illegal database file format.

1) Necessary files may be damaged. Reinstall the ID850NWC, and device file.

A5306: Database information has been already initialized.

A5307: Database information does not exist.

A7012: Not enough memory.

1) There is not enough system memory. Close the applications being executed and the open files.

A70fe: Bus hold error.

1) CPU is in the bus-hold status. Reset the ID850NWC.

A70ff: Can not communicate with ICE.

A7f03: Failed in canceling RUN/STEP.

A9000: Specified register symbol does not exist.

A9001: Specified register symbol ID does not exist.

A9003: Illegal condition.

A9004: Too large register size.

Aa005: Not enough memory.

1) There is not enough system memory. Close the applications being executed and the open files.

Aa00b: Can not close file.

Aa00c: Failed in reading file.

1) The file is damaged or does not exist. Recreate the file.

Aa010: Can not communicate with ICE.

Aa013: Reading of file went wrong.

Ab009: Not enough memory.

- 1) There is not enough system memory. Close the applications being executed and the open files.

Ab00d: Current function does not exist.

Ab00e: Current line does not exist.

Ab00f: Tag not found.

Ab010: Failed in loading symbol table.

Ab011: Illegal line number.

Ab015: Reading of file went wrong.

- 1) The file is damaged or does not exist. Recreate the file.

Ab016: Can not open file.

- 1) The file is damaged or does not exist. Recreate the file.

Ab017: Failed in writing file.

- 1) The file is damaged or does not exist. Recreate the file.

Ab019: Reading of file went wrong.

Ab01a: Can not close file.

Ab01c: Too many entries of the task kind .

Ab023: Current stack frame is not active.

Ab024: Different section.

Ab02a: Can not communicate with ICE.

Ab030: Monitor timed out.

- 1) Check the power of the in-circuit emulator, cable connections, and setting of the interface board and restart the ID850NWC.

Ab031: Already set in memory.

Ab032: Out of scope.

Ab033: LP is not stored.

Ab039: Failed in loading debug information.

Ab03a: No more section information.

Ab041: Specified file is not load module.

Ac002: Can not close file.

Ac003: Reading of file went wrong.

1) The file is damaged or does not exist. Recreate the file.

Ac004: Reading of file went wrong.

Ac009: Not enough memory.

1) There is not enough system memory. Close the applications being executed and the open files.

Ac010: Can not communicate with ICE.

Ad000: Error occurred inside debugger.

Ad001: Not enough memory.

1) There is not enough system memory. Close the applications being executed and the open files.

Ad002: Failed in reading initialization file (expc.ini).

Ad003: ICE is not connected.

Ae008: Not enough memory

1) There is not enough system memory. Close the applications being executed and the open files.

Af000: Not enough memory.

1) There is not enough system memory. Close the applications being executed and the open files.

Af604: Incorrect ID Code. Abort the debugger.

**E. 1. 4 Syntax error messages**

A list of the messages that are output when syntax errors are detected while the ID850NWC is executing processing is shown below in numerical order.

F0002: This feature is not supported.

F0103: 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.

F0104: 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.

F01a0: No response from the evachip. Please confirm the signal of the CLOCK or RESET WAIT, HLDRQ and so on.

F0200: No mapped address was accessed.

- 1) External memory could not be accessed, as it is not set. Change the register values necessary for accessing the external memory using the IOR window or Hook Procedure before download (Refer to "To change the value of a register required for access of an external memory").

F02a0: Bus hold error.

- 1) CPU is in the bus-hold status. Reset the debugger.

F02a2: Can not compulsory break.

F02d2: Not enough memory for trace-buffer.

F0300: User program is running.

F0301: User program is being breaked.

F0302: User program is being traced.

F0303: Not traced.

F0304: Trace memory is not set.

F0306: No trace block exists.

F0307: No event condition exists.

F0308: No timer measurement is done.

F0309: No trigger frame exists.

F030a: Tracer is being stopped.

F030b: Specified snap-event has not been registered.

F030c: Specified stub-event has not been registered.

F030d: Timer is running.

F030e: Memory copy area is overlapped.

F030f: Trace has been already set.

F0310: Event condition is not set.

F0311: Too many valid timer event conditions.

F0312: Specified timer event is not set.

F0313: illegal map range.

1) Check the setting in "Memory Mapping (mapping setting area)" in the Configuration dialog box.

When mapping to external memory has been performed, change the register values necessary for accessing the external memory using the IOR window or Hook Procedure before download (Refer to "To change the value of a register required for access of an external memory").

F0314: Only trace delay mode can set with delay trigger.

F0315: Delay trigger cannot set without trace delay mode.

F03a1: Step execution is being done.

F03a2: Timer and Tracer are running.

F03d0: Back-trace is being executed.

F03d1: Back-trace is being stopped.

F03d2: Back-trace execution point overrun oldest frame.

- F03d3: Register status or Memory status cannot be set up other than Phase1 of event link.
- F03d4: No back-trace information exists.
- F03d5: Last command can not be backstepped.
- F0400: Illegal condition.
- 1) Settings of the used in-circuit emulator and those of the Configuration dialog box may not match.  
Check the Chip selection of "Chip (CPU selection area)".
- F0401: Result of timer measurement overflowed.
- F0402: Too many event conditions with path count.
- F0403: Too many address range conditions.
- F0404: Too many simultaneously-usable-event conditions.
- F0405: Too many snap-events.
- F0406: Too many stub-events.
- F0407: Too many initialization data.
- F0408: Too large search data (> 16 byte).
- F0409: Too large search data (> search range).
- F040a: Too many Linking-event conditions.
- F04a0: Software break conditions number overflow.
- F04a2: Too many partition of bus size.
- F04a3: Too many execution-event conditions.
- F04a4: Too many bus-event conditions.
- F0b61: Section Trace event conditions overflow.

F0c00: Monitor file read error.

- 1) Necessary files may be damaged. Reinstall the ID850NWC.

F0c20: Guarded area can not be accessed.

F0c21: Memory was unready status.

F0c25: Flash macro service ROM was accessed or stepped in.

F0c26: FLMD terminal is in a write-protected state.

- 1) FLMD is not in the write-enabled status. Check the status of the FLMD0 and FLMD1 pins. Check the clock signal. The possible causes are that the clock is stopped or the clock is slow.

F0c27: Security flag is in a write-protected state.

- 1) The security flag of the flash memory has disabled writing, block erasure, or chip erasure. Nothing can be written to the flash memory.

F0c28: Internal RAM is not enough, the writing to flash memory is not made.

- 1) The internal RAM size is less than 4 KB and flash self-programming cannot be executed.

F0c29: The blank check of flash memory failed.

F0c2a: The erasing of flash memory failed.

F0c2b: The writing of flash memory failed.

F0c2c: The internal verification of flash memory failed.

F0c40: Status of effective event conditions cannot be changed.

F0c41: Coverage test is being executed.

F0c60: Event before execution cannot be set up other than break conditions.

F0c61: Can not register event numbers which can not be used for hardware break.

F0c62: Event numbers reserved for hardware breaks can not be used.

F0c63: Event link conditions cannot set.

F0c64: Too many ROM-emulation-RAM areas.



F0ca0: 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.

- 1) Can not communicate with in-circuit emulator. Check the power of the in-circuit emulator, cable connections, and setting of the interface board etc.

F0ca1: Monitor file not found.

- 1) Necessary files may be damaged. Reinstall the ID850NWC.

F1003: Illegal relocation address.

F1004: Illegal condition

F1006: Illegal address.

F100b: User program is running.

F100c: Different bus size has been already specified.

F100d: Too large bus size.

F100e: Too large bus partition size.

F100f: Target is not turned on.

F1010: Illegal map range.

F1011: Failed in setting internal ROM and RAM.

F1012: This feature is not supported.

F1013: No terminal name.

F1016: Programmable-IOR does not movable.

- 1) Necessary files may be damaged. Reinstall the latest device file.

F1017: I/O Protect mapping is possible a target attribute only.

F1018: Illegal Internal ROM size.

F1019: Illegal Internal ROM size or Internal RAM size.

F2000: Illegal I/O register name.

F2002: User program is running.

F2003: Illegal I/O register number.

F2004: Illegal bit number.

F2006: Hidden I/O register was specified.

F2007: I/O register of ban read/write was specified.

F2008: I/O register not existing was specified.

F200a: Illegal value specified for I/O register.

F3000: No mapped address was accessed.

- 1) The allocation addresses of the program and the addresses of the ID850NWC may not match.  
Set the mapping to the external memory in "Memory Mapping (mapping setting area)" in the Configuration dialog box according to the allocation addresses specified in the link directive file on compilation. When mapping to external memory has been executed, change the register values necessary for accessing the external memory using the IOR windowor Hook Procedure before download (Refer to "To change the value of a register required for access of an external memory").

F3001: Memory has different value.

F3002: Illegal start address.

F3003: Illegal end address

F3004: Illegal start address and end address.

F3005: Illegal condition.

F3006: User program is running.

F3007: Verification error.

F3008: No condition specified.

- F3009: Parameter size does not align with access size alignment.
- F300a: Specified address does not align with access size alignment.
- F300b: Source address does not align with access size alignment.
- F300c: Destination address does not align with access size alignment.
- F300d: Illegal end address.
- F300e: Different access size in specified area.
- F300f: Different access size both in source and destination areas.
- F3010: Different access size in destination area.
- F3011: Different access size, source & destination.
- F3013: Failed in writing DMM.
- F3014: Overflowed mapping area.
- F3015: Processing was interrupted.
- F3016: This feature is not supported.
- F4000: Can not delete specified event.
- 1) The specified event cannot be deleted as it is being used under another condition. Invalidate it for other usages before deleting.
- F4001: Illegal table number.
- F4002: Illegal start address.
- F4003: Illegal end address.
- F4004: Illegal status.
- F4005: Illegal data.
- F4006: Specified event number has been already used.

- F4007: Too many same events are registered.
- F4008: Specified event has not been registered.
- F4009: Illegal data size.
- F400a: Illegal mode.
- F400b: Setting value is inaccurate.
- F400c: Event link conditions cannot be used for section trace conditions.
- F400d: Too many identical events are registered ( $\geq 32767$ ).
- F400e: Specified event condition does not exist.
- F400f: Illegal event link condition.
- F4010: Function not found.
- F4012: Timer is being disabled.
- F4014: Can not use software break.
- F4015: Can not use event condition specifying address range.
- F4016: Can not change event condition.
- F4017: Can not access word at odd address.
- F4019: This feature is not supported.
- F401a: No Event.
- F401b: Can not use tag-event.
- F401d: Start event and end event of timer are not made to the same setup.
- F401e: Too many trace-events.

F401f: Path count cannot be set up.

F4020: Address range cannot be set up in event before execution.

F4021: Event conditions number overflow.

F4022: Software DMM conditions number overflow.

F4023: Real-time call conditions number overflow.

F4024: Software break call conditions number overflow.

F4025: Illegal snap condition.

F4026: Too many event conditions cannot be set as Phase1 and Phase2 of event link conditions.

F4027: Software break conditions number which can be set as internal ROM was overflow.

F4318: Illegal memory bank setting.

F5008: Can not open device file.

1) Necessary files may be damaged. Reinstall the device file.

F5009: Can not open ie703000.ie.

F500a: Specified device file is illegal version.

1) Necessary files may be damaged. Reinstall the device file.

F5308: Can not open specified database file.

1) Necessary files may be damaged. Reinstall ID850NWC.

F5309: Specified database file is illegal version.

1) Necessary files may be damaged. Reinstall the ID850NWC, and the device file.

F6000: Current function does not exist.

F6001: Illegal symbol name.

F6002: Illegal condition.

F6003: Illegal function name.

F6004: Overflowed output buffer size.

F6005: Illegal expression.

F7000: Illegal mode.

F7001: User program is running.

F7002: User program has been stopped.

F7003: Trace enabled.

F7004: Trace memory is not set.

F7005: Function return address does not exist, can not do step execution.

F7801: End waiting state of step execution was canceled.

F7802: End waiting state of step execution was canceled.

F7f00: Aborted step execution.

F7f02: Suspended step execution.

F7f04: Can not execute non-mapped area.

F7f05: This feature is not supported.

F8000: Specified file was not found.

F8001: Illegal line number.

F8002: Current information is not set.

F8003: Illegal address.

F8004: This feature is not supported.

F9002: Illegal value.

F9005: This feature is not supported.

Fa001: Illegal expression.

Fa002: Start address is bigger than the end address.

Fa003: Illegal source path.

Fa004: Too long expression.

Fa006: Illegal argument.

Fa007: Illegal program number.

Fa008: Source path is not set.

Fa009: File not found.

Fa00a: Can not open file.

1) The file is damaged or does not exist. Recreate the file.

Fa00d: Not source file of load module.

Fa00e: Illegal line number.

Fa00f: Variable does not exist.

Fa011: Can not access register.

Fa012: Can not access memory.

Fa014: It was going to open the binary file.

Fa015: Can not get temporary path.

1) The disk is full. Delete or move unnecessary files and increase the available memory in the disk.

Fa016: Can not create temporary file.

1) The disk is full. Delete or move unnecessary files and increase the available memory in the disk.

Fa017: Can not remove temporary file.

Fa020: This feature is not supported.

Fa021: Symbol assigned to register cannot be specified.

Fb000: Illegal command line.

Fb001: Program information does not exist in specified load module file.

Fb002: File not found.

Fb003: Function not found.

Fb004: Selected load module different from kind(Chip) was loaded.

Fb005: Symbol not found.

1) The address could not be found. Specify a location holding address information.

Fb008: Illegal expression.

Fb00a: Illegal symbol in load module file.

Fb00b: Current program does not exist.

Fb00c: Current file does not exist.

Fb012: Too large line number.

Fb01b: Too long load module file name.

Fb01d: Address not found.

Fb01f: Can not find structure member.

Fb020: Can not find value.

Fb021: No debug information exists in load module file.

1) To create a load module with appended debug information, execute build in build mode of Debug Build.

Fb022: Illegal line number.



Fb026: Too many array dimensions (> 4).

Fb027: Found end of file.

1) The specified file may be damaged. Recreate the file.

Fb028: This feature is not supported.

Fb029: Illegal address.

Fb02b: Can not stack trace with current PC value.

Fb02c: Too many blocks for one function.

Fb02d: Illegal argument.

Fb02e: The file does not exist in the SOURCE PATH.

1) On stopping the program, the source that the ID850NWC tried to display could not be found. Check if the path connects to the source (using "Source Path (source path specification area)" in the Debugger Option dialog box), or check if the source is in the same directory as the out file. Refer to the Assemble window on which the error message is displayed, and check if the corresponding path connects.

Fb02f: Information has been deleted because of optimization.

Fb034: Return execution from present PC position cannot be performed.

Fb037: Too Many Line-Numbers Information.

Fb038: Compiler version mismatch.

1) Recreate the load module with the latest compiler.

Fb040: Specified file is not load module.

1) This is not a linker output file. Source debug cannot be executed with the load module before output from the linker. Specify the load module output from the linker.

Fb32e: Illegal port number.

Fb32f: Illegal port name.

Fb330: Illegal port position.

Fb331: Illegal increment number.

Fb332: Port for memory bank is not set.

Fb333: Illegal bank number.

Fb334: Area for memory bank is not set.

Fc001: Can not open file.

1) The file is damaged or does not exist. Recreate the file.

Fc005: Illegal file type.

Fc006: Kind(Chip) of load module is illegal.

Fc007: Specified file is not load module.

1) This is not a linker output file. Source debug cannot be executed with the load module before output from the linker. Specify the load module output from the linker.

Fc008: Specified load module file (ELF) is old version.

Fc00a: No mapped address was accessed.

Fc00b: Load module is not loaded.

Fc00c: Illegal argument.

Fc00d: User program is running.

Fc00e: User program is being traced.

Fc00f: Interrupted.

Fc011: Illegal load module file format.

Fc012: Check sum error.

Fc013: Too wide address range to upload (> 1M byte).

Fc014: Failed in writing file.

1) The file is damaged or does not exist. Recreate the file.

Fc015: Illegal program number.

Fc016: Load information is full.

Fc018: Specified file is not load module.

- 1) This is not a linker output file. Source debug cannot be executed with the load module before output from the linker. Specify the load module output from the linker.

Fc019: Failed in writing memory.

Fc01b: Programmable-IOR address not specified.

- 1) Necessary files may be damaged. Reinstall ID850NWC.

Fc100: This feature is not supported.

Fd004: Can not find Dynamic Link Library.

Fe000: Illegal argument.

Fe001: Illegal start address.

Fe002: Illegal end address.

Fe003: Too large size.

Fe004: Can not open file.

- 1) The file is damaged or does not exist. Recreate the file.

Fe005: Failed in reading file.

- 1) The file is damaged or does not exist. Recreate the file.

Fe006: Reading of file went wrong.

Fe007: Failed in writing file.

- 1) The file is damaged or does not exist. Recreate the file.

Fe009: Illegal file format.

Fe00a: Verification error.

Fe010: This feature is not supported.

Ff000: Not enough memory.

Ff001: [XXX] not found.

Ff004: Missing parameter.

Ff005: Illegal function name.

Ff006: Illegal number.

Ff007: Start address is bigger than end address.

Ff008: Illegal symbol or expression.

Ff009: [XXX] This file is illegal type.

Ff100: Disk cannot write or full.

Ff101: File not found.

Ff102: File not Create.

Ff103: Old project file version.

Ff104: Illegal project file format.

Ff105: This file is a project file for [XXX].Please select a correct file.

Ff201: Memory mapping error.

Ff202: Verify error.

- 1) External memory could not be accessed, as it is not set. Change the register values necessary for accessing the external memory using the IOR window or Hook Procedure before download (Refer to "To change the value of a register required for access of an external memory").

Ff301: The symbol being used on the event condition can't be evaluated.

Ff306: This name is too long.

- Ff307: There is the same name in other kinds.
- Ff308: An address can't be omitted.
- Ff309: Illegal address mask.
- Ff30a: Illegal data mask.
- Ff30b: Illegal ext probe mask.
- Ff30c: Illegal ext probe data.
- Ff30d: Illegal pass count.
- Ff30e: Illegal register name.
- Ff310: Illegal delay count.
- Ff312: [XXX] is already there.
- Ff313: Event number already exist.
- Ff314: Event name is not set.
- Ff315: [XXX] is already there.
- Ff316: Max number of enabled [XXX] event is over. Please disable other enabled [YYY] event.
- Ff317: Max number of set event is over.
- Ff31e: Illegal start address.
- Ff31f: Illegal end address.
- Ff350: There is a phase which event are not in the middle.
- Ff351: The same event is contained in Link and Disable.
- Ff352: An event isn't specified.
- Ff400: Coverage mapping error.

Ff500: Illegal symbol.

Ff501: Illegal value.

Ff502: Illegal parameter.

Ff503: Max number of symbol is over.

Ff504: This variable cannot be set as a break.

1) Break cannot be set for the following variables.

- Local variables, static variables
- Array variables, member variables of structures/unions
- Register/peripheral IOR
- Variable expressions

Ff603: Incorrect ID Code.

Ff900: Illegal I/O port name.

Ff901: Memory mapping error.

1) The specification of the address is illegal. The addresses that can be specified are in either the Target area or IOR area. Check in "Address: (I/O port address specification area)" in the Add I/O Port dialog box.

Ff902: Illegal access size.

Ff903: Illegal access type.

Ff904: There is the same name.

Ffa00: The [XXX] function of current program on PC position not found.

1) The symbol specified in main() label: in "Startup Routine (start-up symbol setting area)" in the Debugger Option dialog box could be found. Set a symbol of the main routine of the program. Default is `_main`.

Ffc00: Help window cannot be started. Please install HTML Help environment with reference to a users manual.

**E. 1. 5 Warning message or question message**

A list of the messages that are output when warnings or questions are detected while the debugger is executing processing is shown below in numerical order.

W0c22: Memory unready status was canceled.

W1014: Data is not exist.

W2005: I/O register of Read Protect attribute was specified.

W200d: No initialize data for IOR.

W4013: Access size is different from its mapped bus size.

W401c: Software break can not be set on this area.

W500b: Specified device file does not relocate IRAM.

W7010: No source information exists.

W7011: Unknown result of step execution.

Wb01e: No debug information (not compiled in Debug Build mode).

Wb036: Out of variable region.

Wb042: Symbol module is not initialized.

Wb335: Too long symbol name.

Wc017: Symbol information is duplicated, please reset symbols.

Wc01a: BSS area is assigned to non-mapped area.

- 1) When the program is executed, a non-map break may occur. Either allocate the BSS area to the internal RAM by using a CA link directive, or map the emulation memory or target memory to the BSS area using the Configuration dialog box of the ID850NWC.

Wc01c: Programmable IOR address mismatch.

- 1) Necessary files may be damaged. Reinstall the ID850NWC.

Wc01d: Selected load module different from kind(Chip) was loaded.

- Wf002: Not found [XXX]. Search from the beginning?
- Wf003: Already exceed search region.
- Wf106: CPU in the Project File was Changed. You must exit the debugger for the new CPU. Do you exit the Debugger?
- Wf107: CPU in the Project File was Changed. Do you start the Debugger with this CPU?
- Wf108: Selected project file different [YYY] from chip [XXX] was opened. Does it open, although the chip cannot be changed?
- Wf109: Project Manager cannot be used with the debugger of this version. Please use PMplus.
- Wf200: No difference encountered.
- Wf203: When a program is running, while rewriting a memory, program execution stops for a moment. Do you wish to rewrite a memory?
- Wf300: Would you like to save the changes made in [XXX]?
- Wf302: Delete: [XXX]
- Wf303: [XXX] is edited. Delete: [YYY]?
- Wf304: [XXX] is edited. Save: [YYY]?
- Wf305: [XXX] is already exist. Do you replace it?
- Wf311: Only one [XXX] can be enabled. Do you make this [YYY] to enable?
- Wf325: User program is running. Do you want to stop user program for a moment and set it?
- Wf326: User program is running. Do you want to stop user program for a moment and delete it?
- Wf401: Clear coverage?
- Wf600: Save project file?
- Wf601: When connecting the target system, please turn on the target system.



1) When a target is not connected, simply click the <OK> button.

Wf602: Please change a MODE mask condition or connect the target system.

Wf700: Do you want to download Load Module File?

Wf905: [XXX] is already exist. Do you replace it?

Wf906: Would you like to register the change made in [XXX]?

Wfb00: User program is running. Do you want to stop user program?

## APPENDIX F KEY FUNCTION LIST

Debugging can be efficiently executed by using special function keys.

Note that, in the following explanation, ordinary key representation (generic key representation) is used because the key representation differs depending on the type of keyboard.

Table F-1 through Table F-8 list the functions of the respective keys.

### F.1 Function List of Special Function Keys

Table F-1 Function List of Special Function Keys

Key	Function
BackSpace	Deletes one character before the cursor and moves the cursor to the position of the deleted character. At this time, the character string following the cursor moves forward.
Delete	Deletes one character after the cursor and move the character string following the cursor forward. Deletes a various event condition selected in the Event Manager or each event dialog box. Deletes the data selected in the Watch window.
Insert	Alternately selects the insert mode and overwrite mode in the Source window and Assemble window. However, this key is invalid in the Memory, Register, and IOR windows, and only the overwrite mode can be used as an input mode.
PrintScreen	Loads the entire display screen to the clipboard as a bitmap image (function of Windows).
Esc	Closes the pull-down menu. Closes the modal dialog box. Restores the input data.
Alt	Moves the cursor to the menu bar.
End	Moves the cursor to the end of the line.
Home	Moves the cursor to the beginning of the line.
PageUp	Scrolls the screen one screen up. The cursor also moves up to the top of the screen.
PageDown	Scrolls the screen one screen down. The cursor also moves up to the top of the screen.
Space	Inserts one blank character.
Tab	Moves the cursor to the next item.
Up arrow key	Moves the cursor up. If the cursor is at the bottom of the screen, scrolls the screen up one line at a time.
Down arrow key	Moves the cursor down. If the cursor is at the top of the screen, scrolls the screen down one line at a time.
Right arrow key	Moves the cursor to the left. If the cursor is at the left most position on the screen, scrolls the screen one column to the right.

Table F-1 Function List of Special Function Keys

Key	Function
Left arrow key	Moves the cursor to the right. If the cursor is at the right most position on the screen, scrolls the screen one column to the left.
Enter	Sets the input data. Presses the default push button.

## F.2 Function List of Function Keys

Table F-2 Function List of Function Keys

Key	Function
F1	Opens the Help window.
F2	Forcibly stops program execution. Same function as [Run] -> [Stop] on the menu bar.
F3	Resets the CPU. Same function as [Run] -> [CPU Reset] on the menu bar.
F4	Resets the CPU and executes the program. Same function as [Run] -> [Restart] on the menu bar.
F5	Executes the program. Same function as [Run] -> [Go] on the menu bar.
F6	Executes the program to the cursor position in the Source or Assemble window. Same function as [Run] -> [Come Here] on the menu bar
F7	The user program is real-time executed until execution returns. Same function as [Run] -> [Return Out] on the menu bar
F8	Step execution. Same function as [Run] -> [Step In] on the menu bar.
F9	Sets a breakpoint at cursor position in Source or Assemble window. Same function as [Run] -> [Break Point] on the menu bar.
F10	Next step execution. Same function as [Run] -> [Next Over] on the menu bar.
F11	Sets or deletes a software breakpoint. Same function as [Run] -> [Software Break Point] on the menu bar.

## F.3 Function List of Special Function Keys (Shift + key)

Table F-3 Function List of Special Function Keys (Shift + key)

Key	Function
End	Expands the selection range to the end of the line.
Home	Expands the selection range to the beginning of the line.
Left arrow key	Expands the selection range one character to the left.
Right arrow key	Expands the selection range one character to the right.

**F.4 Function List of Function Keys (Shift + key)**

Table F-4 Function List of Function Keys (Shift + key)

Key	Function
F6	Executes the program from the cursor position in the Source or Assemble window. Same function as [Run] -> [Start From Here] on the menu bar.
F9	Resets the CPU. Same function as [Run] -> [CPU Reset] on the menu bar.

**F.5 List of Special Function Keys (Ctrl + key)**

Table F-5 List of Special Function Keys (Ctrl + key)

Key	Function
End	Displays the last line. The cursor will also move to the last line.
Home	Displays the first line. The cursor will also move to the first line.
Left arrow key	Moves the cursor one word to the left. If the cursor at the left most position on the screen, scrolls the screen one column to the right.
Right arrow key	Moves the cursor one word to the right. If the cursor at the right most position on the screen, scrolls the screen one column to the left.

**F.6 Function List of Function Keys (Ctrl + key)**

Table F-6 Function List of Function Keys (Ctrl + key)

Key	Function
F5	Ignores break points being set, and executes the program. Same function as [Run] -> [Ignore break points and Go] on the menu bar.
F9	Sets the address at the cursor position in the Source window or Assemble window to the PC. Same function as [Run] -> [Change PC] on the menu bar.

**F.7 Function List of Control Keys (Ctrl + key)**

Table F-7 Function List of Control Keys (Ctrl + key)

Key	Function
A	Selects all the events registered to the Event Manager. Same function as [View] -> [Select All Event] on the menu bar in the Event Manager.
C	Copies a selected character string and saves it to the clipboard buffer.

Table F-7 Function List of Control Keys (Ctrl + key)

Key	Function
D	Disassembles and displays the results from the jump destination address specified by the data value selected in the current window. Opens the Assemble window. Same function as [Jump] -> [Assemble] on the menu bar.
E	Opens the source file displayed in the active Source window with the editor specified by the PM plus when the PM plus is running. Same function as [Edit] -> [Edit Source] on the menu bar.
G	Performs a search. Opens the search dialog box corresponding to the current window. Same function as [View] -> [Search...] on the menu bar.
J	Moves the display position. Opens the Source Move, Address Move, or Trace Move dialog box, depending on the current window. Same function as [View] -> [Move...] on the menu bar.
M	Displays the memory contents from the jump destination address specified by the data value selected in the current window. Opens the Memory window. Same function as [Jump] -> [Memory...] on the menu bar.
O	Loads a view file, source file, or text file. Opens the View File Load dialog box. The operation will differ depending on the extension of the file. view file: Displays the file in the corresponding window. Others: Displays the file in the Source window. Same function as [File] -> [Open...] on the menu bar.
S	Saves the data displayed in the current window to the view file. Same function as [View] -> [Save...] on the menu bar.
U	Displays the corresponding source text and source line, using the data value selected in the current window as the jump destination address. Opens the Source window. Same function as [Jump] -> [Source Text] on the menu bar.
V	Pastes the contents of the clipboard buffer to the text cursor position.
W	Temporarily displays the contents of the specified data. Opens the Quick Watch dialog box. Same function as [View] -> [Quick Watch...] on the menu bar.
X	Cuts a selected character string and saves it to the clipboard buffer. Same function as [Edit] -> [Cut...] on the menu bar.

**F.8 List of Special Function Keys (Ctrl + Shift + key)**

Table F-8 List of Special Function Keys (Ctrl + Shift + key)

Key	Function
Left arrow key	Expands the selection range one word to the left.
Right arrow key	Expands the selection range one word to the right.

## APPENDIX G INDEX

### Symbols

[Come Here] function ... 117  
[run] menu ... 29  
[Start from Here] function ... 117

### A

About dialog box ... 220  
active status ... 39  
Add I/O Port dialog box ... 186  
Add Watch dialog box ... 159  
address ... 229  
Address Move dialog box ... 104  
ASCII display ... 134  
assemble ... 230  
Assemble Search dialog box ... 129  
Assemble window ... 124  
auto load ... 27, 74, 84  
auto save ... 74, 82

### B

balloon watch function ... 34, 113, 154  
batch ... 231  
Break dialog box ... 212  
break function ... 31  
breakpoint ... 232  
breakpoint for read/write access ... 155  
breakpoint setting/deletion function ... 111  
Browse dialog box ... 99  
build mode ... 24

### C

cache ... 234  
callback procedure ... 225  
Change Watch dialog box ... 163  
character set ... 276  
color of character on event mark ... 191  
color of character on mark ... 191  
command ... 17, 221  
command line ... 222  
command list ... 222  
command reference ... 222  
compulsion read ... 178  
Configuration dialog box ... 62  
Console window ... 221  
contents saved to project file ... 80  
current window ... 281

### D

data display format ... 152  
dbgexit ... 236  
Debugger Option dialog box ... 71  
delimiter ... 281  
device file ... 18  
disable condition ... 209  
display color of breakpoint ... 111  
displaying event details ... 193  
DMM function ... 136  
download ... 237

Download dialog box ... 91  
drag & drop function ... 119

### E

error messages at startup ... 284  
event condition ... 33  
Event dialog box ... 199  
event display mark ... 112  
event function ... 33  
event link condition ... 33  
Event Link dialog box ... 206  
Event Manager ... 189  
Exit Debugger dialog box ... 218  
EXPANSION ... 268  
expansion window ... 268  
expressions ... 278  
Extended Option dialog box ... 68  
extwin ... 238

### F

file ... 281  
finish ... 239  
flash memory ... 29, 64  
Flow of debugging operations ... 37  
font ... 73  
Font dialog box ... 78  
Format of file that can be downloaded ... 91  
format of specifiable file ... 95  
function ... 282

### G

-g option ... 18  
go ... 240

### H

help ... 241  
hook ... 242  
hook procedure ... 226  
host machine ... 18

### I

I/O port ... 178, 186  
I/O protect area ... 28  
ID code ... 18, 64  
ie ... 243  
in-circuit emulator ... 18  
information files ... 36  
input conventions ... 276  
installation ... 19  
IOR Select dialog box ... 183  
IOR window ... 178

### J

jump ... 244  
jump function ... 39

### K

key function list ... 312

**L**

Load module list dialog box ... 97  
 load/save function ... 35  
 Local Variable window ... 166

**M**

Main window ... 44  
 map ... 245  
 mapping function ... 28  
 mapping unit ... 67  
 marks of event icon ... 191  
 mask ... 203  
 mdi ... 246  
 memory ... 247  
 Memory Compare dialog box ... 144  
 Memory Compare Result dialog box ... 146  
 Memory Copy dialog box ... 142  
 Memory Fill dialog box ... 140  
 memory manipulation function ... 34  
 Memory Search dialog box ... 137  
 Memory window ... 132  
 menu bar ... 47  
 messages ... 283  
 mixed display ... 116  
 module ... 248

**N**

next ... 249  
 normal mode ... 200, 207  
 N-Wire Checker ... 261

**O**

offset setting ... 93  
 online assembly ... 124  
 operating environment ... 18  
 operating system ... 18  
 operators ... 278

**P**

pass count ... 209  
 PM plus ... 24  
 point mark area ... 111, 125  
 program ... 281  
 program code ... 112, 114  
 program counter setting ... 117  
 program execution function ... 29  
 program register ... 174  
 programmable IOR ... 34  
 project file ... 29, 74, 218  
 Project File Load dialog box ... 83  
 Project File Save dialog box ... 80

**Q**

Quick Watch dialog box ... 156

**R**

range of radices ... 280  
 refresh ... 250  
 register ... 251  
 register manipulation function ... 34

Register Select dialog box ... 176  
 Register window ... 173  
 registering debugger ... 24  
 reset ... 216, 252  
 Reset Debugger dialog box ... 216  
 run ... 253

**S**

select mode ... 200, 204, 207, 210  
 selecting debugger ... 24  
 setting debugging environment ... 29  
 setting event condition in each area ... 208  
 setting files ... 36  
 show offset ... 76  
 software break ... 32  
 Software Break Manager ... 196  
 Source Move dialog box ... 101  
 source path ... 72  
 Source Search dialog box ... 121  
 Source window ... 109  
 specification of scope ... 160  
 specifying symbols ... 107  
 stack frame ... 170, 282  
 start-up symbol ... 75  
 static status ... 39  
 status bar ... 59  
 step ... 254  
 stop ... 255  
 storing program in ROM ... 18  
 structure ... 282  
 Symbol To Address dialog box ... 106  
 symbols ... 277  
 system register ... 174

**T**

tab size ... 75  
 terminology ... 281  
 toolbar ... 44

**U**

uninstalling ... 20  
 upload ... 256  
 Upload dialog box ... 95

**V**

verify check ... 70  
 version ... 257  
 view file ... 35  
 View File Load dialog box ... 88  
 View File Save dialog box ... 85

**W**

watch ... 258  
 watch function ... 34, 118  
 watch symbol setting ... 76  
 Watch window ... 150  
 where ... 259  
 window lists ... 41  
 window reference ... 41  
 wish ... 260