

To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

User's Manual

TW850 Ver. 2.00

Performance Analysis Tuning Tool

Target Device V850 Series

Document No. U17241EJ1V0UM00 (1st edition)

Date Published October 2004 CP(K)

© NEC Electronics Corporation 2004

Printed in Japan

[MEMO]

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.

Java and all trademarks and logos related to Java are trademarks of Sun Microsystems, Inc.

• **The information in this document is current as of October, 2004. 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).

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.

[GLOBAL SUPPORT]

<http://www.necel.com/en/support/support.html>

NEC Electronics America, Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65030

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318

- **Sucursal en España**
Madrid, Spain
Tel: 091-504 27 87

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-558-3737

- **Succursale Française**
Vélizy-Villacoublay, France
Tel: 01-30-67 58 00

NEC Electronics Shanghai Ltd.
Shanghai, P.R. China
Tel: 021-5888-5400

- **Filiale Italiana**
Milano, Italy
Tel: 02-66 75 41

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377

- **Branch The Netherlands**
Eindhoven, The Netherlands
Tel: 040-244 58 45

NEC Electronics Singapore Pte. Ltd.
Novena Square, Singapore
Tel: 6253-8311

- **Tyskland Filial**
Taebby, Sweden
Tel: 08-63 80 820

- **United Kingdom Branch**
Milton Keynes, UK
Tel: 01908-691-133

J04.1

[MEMO]

PREFACE

Target Readers	This manual is intended for user engineers who design and develop application systems of the V850 Series.												
Purpose	This manual is intended for users to understand the functions of the TW850 in the organization below.												
Organization	<p>This manual consists of the following chapters:</p> <ul style="list-style-type: none">• GENERAL• INSTALLATION• STARTING AND EXITING• FUNCTIONS• METHOD OF OPERATION• WINDOW REFERENCE												
How to Use This Manual	<p>It is assumed that the readers of this manual have general knowledge of electrical engineering, logic circuits, microcontrollers, C language, and assemblers.</p> <p>To understand the functions of the V850 Series → Refer to Hardware User's Manual for each product.</p> <p>To understand the instruction functions of the V850 Series → Refer to V850 Series Architecture User's Manual or V850E/MS1 Architecture User's Manual.</p>												
Conventions	<table><tr><td>Data significance:</td><td>Higher digits on the left and lower digits on the right</td></tr><tr><td>Note:</td><td>Footnote for item marked with Note in the text</td></tr><tr><td>Caution:</td><td>Information requiring particular attention</td></tr><tr><td>Remark:</td><td>Supplementary information</td></tr><tr><td>Numerical representation:</td><td>Binary ... XXXX or XXXXB Decimal ... XXXX Hexadecimal ... XXXXH</td></tr><tr><td>Prefix indicating the power of 2 (address space, memory capacity):</td><td>K (Kilo): $2^{10} = 1024$ M (Mega): $2^{20} = 1024^2$ G (Giga): $2^{30} = 1024^3$</td></tr></table>	Data significance:	Higher digits on the left and lower digits on the right	Note:	Footnote for item marked with Note in the text	Caution:	Information requiring particular attention	Remark:	Supplementary information	Numerical representation:	Binary ... XXXX or XXXXB Decimal ... XXXX Hexadecimal ... XXXXH	Prefix indicating the power of 2 (address space, memory capacity):	K (Kilo): $2^{10} = 1024$ M (Mega): $2^{20} = 1024^2$ G (Giga): $2^{30} = 1024^3$
Data significance:	Higher digits on the left and lower digits on the right												
Note:	Footnote for item marked with Note in the text												
Caution:	Information requiring particular attention												
Remark:	Supplementary information												
Numerical representation:	Binary ... XXXX or XXXXB Decimal ... XXXX Hexadecimal ... XXXXH												
Prefix indicating the power of 2 (address space, memory capacity):	K (Kilo): $2^{10} = 1024$ M (Mega): $2^{20} = 1024^2$ G (Giga): $2^{30} = 1024^3$												

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, V850/SA1, V850/SB1, V850/SB2, V850/SC1, V850/SC2, V850/SC3, V850/SF1, V850/SV1)	U11595E	
IE-V850E-MC, IE-V850E-MC-A (In-circuit emulator for V850E/MA1, V850E/MA2, V850E/IA1, V850E/IA2, V850E/SV2)	U14487E	
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)	U13875E	
IE-703102-MC-EM1, IE-703102-MC-EM1-A (In-circuit emulator option board for V850E/MS1)	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	
IE-703166-MC-EM1 (In-circuit emulator option board for V850E/SV2)	U16688E	
CA850 Ver. 3.00 C Compiler Package	Operation	U17293E
	C Language	U17291E
	Assembly Language	U17292E
	Link Directives	U17294E
PM plus Ver.6.00		U17178E
ID850 Ver. 2.50 Integrated Debugger	Operation	U16217E
ID850NW Ver. 2.51 Integrated Debugger	Operation	U16454E
ID850QB Ver. 2.80 Integrated Debugger	Operation	U16973E
SM plus System Simulator	Operation	U17246E
	User Open Interface	U17247E
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. 3.15 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
TW850 Ver. 2.00 Performance Analysis Tuning Tool		This manual

CONTENTS

CHAPTER 1 GENERAL	15
1.1 Terminology	15
1.2 Functional Outline	16
1.3 Features	16
1.4 Operating Environment	17
1.4.1 Hardware environment	17
1.4.2 Software environment	17
CHAPTER 2 INSTALLATION	18
2.1 Installation Procedure	18
2.2 Uninstallation Procedure	19
2.3 Folder Configuration	20
CHAPTER 3 STARTING AND EXITING	22
3.1 Starting Method	23
3.1.1 Starting from PM plus	23
3.1.2 Start options	24
3.2 Exiting Method	25
3.2.1 Exiting form menu bar	25
3.2.2 Exiting form title bar	25
CHAPTER 4 FUNCTIONS	26
4.1 Performance Analysis Functions	26
4.1.1 Performance analysis target	26
4.1.2 Methods for acquiring performance analysis information	28
4.1.3 Details of performance analysis functions	30
4.1.4 Notes on performance analysis processing	32
4.2 Tuning Functions	33
4.2.1 Tuning target	33
4.2.2 Details of tuning functions	34
4.2.3 Notes on tuning processing	36
4.3 GUI Functions	37
4.3.1 Details of GUI functions	37
4.4 Timer Control Code Block	38
4.4.1 Creating timer control code block	38
4.4.2 Inspecting timer control code block	40
4.4.3 Notes on time measurement processing	41
4.5 System Library for TW850	42
4.6 Notes on Assembly Source File	43
4.6.1 Detection of functions subject to performance analysis and tuning	43
4.6.2 Detection of location calling function	45
4.6.3 Description format of assembly source file	47
CHAPTER 5 METHOD OF OPERATION	54
5.1 Starting TW850	57
5.1.1 Starting TW850	57
5.1.2 Reading project	57
5.2 Starting Wizard	58
5.2.1 Starting wizard	58

5.2.2	Acquiring project information	59
5.2.3	Selecting analysis source file	60
5.2.4	Acquiring execution module information	61
5.3	Selecting Execution Item	62
5.4	When [Use trace data for analyze.] Is Selected	63
5.4.1	Selecting trace analysis option	63
5.5	When [Use probe code for analyzing (for original execution module only).] Is Selected	64
5.5.1	Selecting probe analysis option	64
5.6	When Only [Analyze an original execution module.] Is Selected	65
5.6.1	Selecting execution analysis range	65
5.6.2	Selecting trace event	66
5.6.3	Selecting instruction cache	67
5.6.4	Selecting allocation destination	68
5.7	When [Tune and analyze a tuned execution module.] Is Selected	69
5.7.1	Selecting execution analysis range	69
5.7.2	Selecting trace event	70
5.7.3	Selecting instruction cache	71
5.7.4	Selecting fast access memory area	72
5.7.5	Selecting tuning method	73
5.8	When [Default Tuning] Is Selected	74
5.8.1	Selecting allocation destination	74
5.9	When [Set By User] Is Selected	75
5.9.1	Selecting tuning method for fast access memory area	75
5.9.2	Selecting tuning method of instruction cache	76
5.9.3	Selecting callpair correction option	77
5.9.4	Selecting priority allocation function	78
5.9.5	Selecting allocation destination	79
5.10	Executing Performance Analysis/Tuning	80
5.10.1	Checking before execution	80
5.10.2	Executing performance analysis/tuning	81
5.10.3	Checking completion of execution	82
5.11	Checking Result	83
5.11.1	History information	83
5.11.2	Execution module information	84
5.11.3	Callgraph information	85
5.11.4	Function information table	86
5.11.5	Detailed function information	87
5.11.6	Function transition information	88
5.12	Exiting TW850	89
CHAPTER 6 WINDOW REFERENCE		90
6.1	Input Rules	93
6.2	Window Explanation	94
	Main Window	95
	Callgraph Window	103
	Function Information Table Window	106
	Function Transition Information Window	109
	Execution Module Information Window	114
	Function Detail Information Window	116
	Initial Screen Panel	120
	Project Information Acquisition Panel	122
	Analysis Source File Selection Panel	123
	Execution Module Information Acquisition Panel	125
	Execution Item Selection Panel	126

Trace Data Analysis Method Option Selection Panel	128
Probe Code Insertion Analysis Method Option Selection Panel	130
Execution Analysis Range Selection Panel	133
Trace Event Selection Panel	137
Instruction Cache Selection Panel	141
Fast Access Memory Area Selection Panel	144
Tuning Method Selection Panel	146
Fast Access Memory Area Tuning Selection Panel	149
Instruction Cache Tuning Selection Panel	151
Callpair Correction Option Selection Panel	154
Priority Allocation Function Selection Panel	156
Allocation Destination Selection Panel	159
Pre-execution Confirmation Panel	163
Performance Analysis/Tuning Execution Panel	164
Execution Completion Confirmation Panel	166
Select Display Function Dialog Box	167
Search Function Transition Information Dialog Box	171
Option Dialog Box	175
Version Information Dialog Box	178
File Selection Dialog Box	180
Message Dialog Box	183
Error Dialog Box	184
Verification Dialog Box	186
Input Dialog Box	188
TW850 Options Dialog Box	190
APPENDIX A MESSAGES	193
A.1 Display Format	193
A.2 Information Messages	194
A.3 Warning Messages	196
A.4 Error Messages	197
APPENDIX B RESTRICTION AND LIMIT	214
B.1 Restriction	214
B.2 Limit	214
APPENDIX C INTERMEDIATE FILES	215
INDEX	220

LIST OF FIGURES

Figure 2-1	Folder Configuration	20
Figure 3-1	Main Window	23
Figure 3-2	Verification Dialog Box	25
Figure 4-1	Image of Updating RTOS Information	31
Figure 5-1	Flow of Wizard Processing (Performance Analysis and Tuning)	54
Figure 5-2	Procedure for Operating the TW850/Checking the Results of Performance Analysis	55
Figure 5-3	Main Window	57
Figure 5-4	Initial Screen Panel	58
Figure 5-5	Project Information Acquisition Panel	59
Figure 5-6	Analysis Source File Selection Panel	60
Figure 5-7	Execution Module Information Acquisition Panel	61
Figure 5-8	Execution Item Selection Panel	62
Figure 5-9	Trace Data Analysis Method Option Selection Panel	63
Figure 5-10	Probe Code Insertion Analysis Method Option Selection Panel	64
Figure 5-11	Execution Analysis Range Selection Panel	65
Figure 5-12	Trace Event Selection Panel	66
Figure 5-13	Instruction Cache Selection Panel	67
Figure 5-14	Allocation Destination Selection Panel	68
Figure 5-15	Execution Analysis Range Selection Panel	69
Figure 5-16	Trace Event Selection Panel	70
Figure 5-17	Instruction Cache Selection Panel	71
Figure 5-18	Fast Access Memory Area Selection Panel	72
Figure 5-19	Tuning Method Selection Panel	73
Figure 5-20	Allocation Destination Selection Panel	74
Figure 5-21	Fast Access Memory Area Tuning Selection Panel	75
Figure 5-22	Instruction Cache Tuning Selection Panel	76
Figure 5-23	Callpair Correction Option Selection Panel	77
Figure 5-24	Priority Allocation Function Selection Panel	78
Figure 5-25	Allocation Destination Selection Panel	79
Figure 5-26	Pre-execution Confirmation Panel	80
Figure 5-27	Performance Analysis/Tuning Execution Panel	81
Figure 5-28	Execution Completion Confirmation Panel	82
Figure 5-29	History Display Area	83
Figure 5-30	Execution Module Information Window	84
Figure 5-31	Callgraph Window	85
Figure 5-32	Function Information Table Window	86
Figure 5-33	Function Detail Information Window	87
Figure 5-34	Function Transition Information Window	88
Figure 5-35	Verification Dialog Box	89
Figure 6-1	Main Window	95
Figure 6-2	Callgraph Window	103
Figure 6-3	Function Information Table Window	106
Figure 6-4	Function Transition Information Window	109
Figure 6-5	Execution Module Information Window	114
Figure 6-6	Function Detail Information Window	116
Figure 6-7	Initial Screen Panel	120
Figure 6-8	Project Information Acquisition Panel	122
Figure 6-9	Analysis Source File Selection Panel	123
Figure 6-10	Execution Module Information Acquisition Panel	125

Figure 6-11	Execution Item Selection Panel	126
Figure 6-12	Trace Data Analysis Method Option Selection Panel	128
Figure 6-13	Probe Code Insertion Analysis Method Option Selection Panel	130
Figure 6-14	Execution Analysis Range Selection Panel	133
Figure 6-15	Trace Event Selection Panel	137
Figure 6-16	Instruction Cache Selection Panel	141
Figure 6-17	Fast Access Memory Area Selection Panel	144
Figure 6-18	Tuning Method Selection Panel	146
Figure 6-19	Fast Access Memory Area Tuning Selection Panel	149
Figure 6-20	Instruction Cache Tuning Selection Panel	151
Figure 6-21	Callpair Correction Option Selection Panel	154
Figure 6-22	Priority Allocation Function Selection Panel	156
Figure 6-23	Allocation Destination Selection Panel	159
Figure 6-24	Pre-execution Confirmation Panel	163
Figure 6-25	Performance Analysis/Tuning Execution Panel	164
Figure 6-26	Execution Completion Confirmation Panel	166
Figure 6-27	Select Display Function Dialog Box	167
Figure 6-28	Search Function Transition Information Dialog Box	171
Figure 6-29	Search Function Transition Information Dialog Box (When Expansion Search Condition Is Displayed)	171
Figure 6-30	Option Dialog Box	175
Figure 6-31	[System] Tab	176
Figure 6-32	Version Information Dialog Box	178
Figure 6-33	File Selection Dialog Box	180
Figure 6-34	Message Dialog Box	183
Figure 6-35	Error Dialog Box (When Starting TW850)	184
Figure 6-36	Error Dialog Box (When Executing TW850)	184
Figure 6-37	Verification Dialog Box	186
Figure 6-38	Input Dialog Box	188
Figure 6-39	TW850 Options Dialog Box	190
Figure A-1	Message Display Area in Main Window	193
Figure A-2	Message Dialog Box	193
Figure A-3	Error Dialog Box	193

LIST OF TABLES

Table 2-1	TW850 Supply Format	18
Table 4-1	Restrictions on Assembly Source File	47
Table 6-1	Windows, Panels, and Dialog Boxes of TW850	90
Table B-1	Restrictions of TW850	214
Table B-2	Limits of TW850	214

CHAPTER 1 GENERAL

The TW850 is software (performance analysis tuning tool) that analyzes and tunes the execution performance of a user program that runs on NEC Electronics V850 Series of 32-bit microcontrollers for embedded control.

1.1 Terminology

The following terms are used in this manual to explain the features and functions of the TW850.

- Callpair

The relationship between a function and another function that is called by that function is called a "callpair".

At this time, the function that calls is called the parent function and the function that is called is called the child function.

- Hot path

The function transition path that takes the longest time in the program to execute is called a "hot path".

If the user program is tuned with priority given to the functions corresponding to a hot path, the performance of the program can be substantially improved in many cases.

- Execution module

The TW850 generates an execution module for original analysis, an execution module for profile, and a tuned execution module as necessary.

The execution module for original analysis is a module that inserts a label or file symbol at the beginning of a function to collect the static information of a function in the source program. The performance does not differ from when a label or file symbol is not inserted.

- Instruction cache miss

The instruction cache that can be accessed at high speeds is smaller than the program in most cases. If a program is executed by a processor equipped with an instruction cache, therefore, the program loaded to the cache before is removed and the program being executed at that time is loaded. If no program to be executed is found in the instruction cache at this time, this is called an "instruction cache miss".

The occurrence rate of instruction cache misses is calculated by the following expression.

$$\text{Number of instruction cache misses} / (\text{Number of instruction cache hits} + \text{Number of instruction cache misses})$$

- Fast access memory area

With the TW850, a memory area of the target system that can be accessed at high speed is called a "fast access memory area".

The program can be sped up by allocating frequently used functions and functions that must be executed at high speeds to this fast access memory area.

- Execution engine

A program that executes the execution module in order to acquire the profile information of the module, such as a debugger or simulator, is called an "execution engine".

- Far jump calling

Calling via an instruction that can branch to the entire address space is called "far jump calling".

If the probe code insertion method is used, the probe code to be inserted may call a profile routine. If probe code and profile routine are located far away from each other, a compilation error may occur.

If far jump calling is used to call the profile routine in this case, a profile routine located far away can be called.

1.2 Functional Outline

The TW850 supplies the following performance analysis functions, tuning functions, and GUI functions. Refer to "CHAPTER 4 FUNCTIONS" for details of each function.

- **Performance Analysis Functions**
 - Acquiring memory allocation information
 - Acquiring callpair information
 - Acquiring call count information
 - Acquiring execution time information
 - Acquiring instruction cache information
 - Acquiring RTOS information
- **Tuning Functions**
 - Tuning of fast access memory area
 - Tuning of instruction cache
 - Correcting callpair information
 - Specifying function to be given priority and allocated
- **GUI Functions**
 - Setting in wizard format
 - Displaying performance analysis and tuning results

1.3 Features

The TW850 has the following features.

- Performance prediction and estimation
The performance of the user program or execution module can be predicted and estimated before the target system is evaluated, by using performance analysis functions supplied by the TW850.
- Easy performance analysis and tuning
Graphical guidance to the settings (such as the performance analysis zone and execution environment) required for using performance analysis and tuning functions supplied by the TW850 is provided in a wizard format so that the settings can be made easily and efficiently.
- Supply of detailed performance analysis information
A location that causes execution performance of an execution module to drop can be detected in function units.
- Automatic generation of tuned execution module
By making settings (inputting corresponding information) via graphical guidance enabled by a wizard format, an execution module that has been tuned (tuned execution module) is automatically generated by the TW850.

1.4 Operating Environment

1.4.1 Hardware environment

The following hardware is necessary for the TW850 to operate.

- Host machine
IBM PC/AT compatible machine
- CPU
Pentium 450 MHz or higher
- Main memory
128 Mbytes or more (256 Mbytes or more is recommended)

1.4.2 Software environment

The following software is necessary for the TW850 to operate.

- Basic software (required)
Windows 2000 Professional, Windows 98 Second Edition, Windows Me, Windows XP Professional, Windows XP Home Edition, Windows NT 4.0 Service Pack 5 or later

Remark It is recommended that the latest Service Pack be installed in any of the above OSs.

Caution The Windows XP user-switching function is not supported.

- Java execution environment (required)
Java2 Runtime Environment Standard Edition V1.4.2_04: Sun Microsystems, Inc.
- C compiler package (required)
CA850 Ver.3.00 or later: made by NEC Electronics
- Device file (required)
Device file of target device to be used: made by NEC Electronics
- Project manager (required)
PM plus Ver.6.00 or later: made by NEC Electronics
- RTOS(Real-time OS)
RX850 Ver.3.20: made by NEC Electronics
RX850 Pro Ver.3.20: made by NEC Electronics

Select any of the following(required)

- Debugger
ID850 Ver.3.00 or later: made by NEC Electronics
ID850NW Ver.3.00 or later: made by NEC Electronics

Caution When using the ICE server of Midas Lab, use the ID850NW with RTE for WIN32 (Ver.5.10 or later).

- Simulator
SM850 Ver.3.00 or later: made by NEC Electronics
SM plus for V850 Ver.2.00 or later: made by NEC Electronics

Remark Only ToolLink-supporting versions of SM plus for V850 can be used.

CHAPTER 2 INSTALLATION

This chapter explains the procedure of installing and uninstalling the files stored in the supply medium of the the TW850 in the user's development environment (host machine).

A TW850 supply medium is available for each type of host machine (Windows-based).

[Table 2-1](#) shows the supply format of the TW850.

Table 2-1 TW850 Supply Format

Host machine	Supply Format	Supply Medium
Windows-based - IBM PC/AT compatible machine	NEC Electronics tool chain-supporting version	CD-ROM

2.1 Installation Procedure

The procedure for installing the files stored in the supply medium of the TW850 in the host machine is as follows.

1) Start Windows

Turn on the power of the host machine and peripheral equipment and start Windows.

2) Set the supply medium

Setting the supply medium of the TW850 in the CD-ROM drive of the host machine is automatically executed the setup program.

After that, perform installation in accordance with the messages displayed on the monitor screen.

Caution If the setup program is not automatically executed, start Install.exe stored in the CD-ROM drive with Explorer.

3) Check files

Check that the files stored in the supply medium of the TW850 have been installed in the host machine by using a standard application of Windows, such as Explorer.

Refer to "[2.3 Folder Configuration](#)" for details of each folder.

Remarks 1. Before re-installing the TW850, uninstall the TW850.

2. The ToolLinkManager is also installed at the same time.

2.2 Uninstallation Procedure

The procedure for uninstalling the files installed in the host machine is as follows.

1) Start Windows

Turn on the power of the host machine and peripheral equipment and start Windows.

2) Open [Control Panel] window

Select [Settings] menu -> [Control Panel] from the <Start> button to open the [Control Panel] window.

3) Open [Add/Remove Programs Properties] window

Double-click the [Add/Remove Programs] icon on the [Control Panel] window to open the [Add/Remove Programs Properties] window.

Remark In the case of Windows XP, "Add/Delete Application" is replaced by "Add/Delete Program".

4) Delete the TW850

Select "NEC EL TW850 V2.00" from the list of installed software displayed on the [Install/Uninstall] tab in the [Add/Remove Programs Properties] window and click the <Add/Remove...> button.

Click the <Yes> button when the [Confirm File Deletion] window opens.

5) Check files

Check that the files installed in the host machine have been uninstalled using a standard application of Windows, such as Explorer.

Refer to "[2.3 Folder Configuration](#)" for details of each folder.

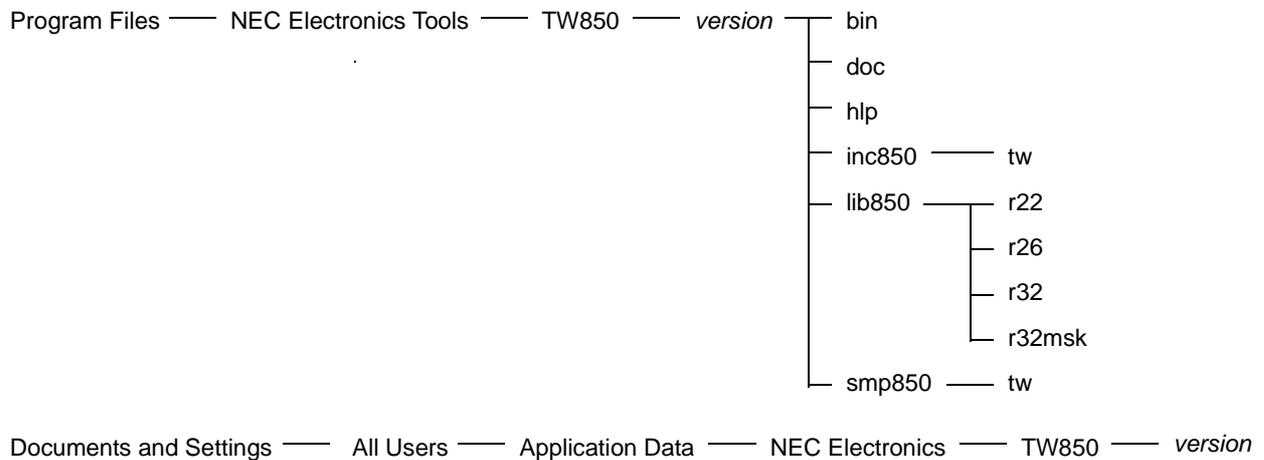
Remark The ToolLinkManager is not uninstalled. To uninstall the ToolLinkManager, use the [Control Panel] window in the same manner as the above.

2.3 Folder Configuration

Figure 2-1 shows the folder configuration that is generated when the files stored in the supply medium of the TW850 are installed in the host machine.

Figure 2-1 is an example that the TW850 is installed in the folder "Program Files\NEC Electronics Tools".

Figure 2-1 Folder Configuration



The overview of each folder is as follows.

- 1) Program Files\NEC Electronics Tools\TW850\version\bin
This folder stores the executable files of the TW850.
TW850.EXE: Executable file of the TW850
- 2) Program Files\NEC Electronics Tools\TW850\version\doc
This folder stores the documents of the TW850.
- 3) Program Files\NEC Electronics Tools\TW850\version\hlp
This folder stores the help of the TW850.
TW850.CHM: Help
- 4) Program Files\NEC Electronics Tools\TW850\version\inc850\tw
This folder stores the header files of the TW850.
- 5) Program Files\NEC Electronics Tools\TW850\version\lib850
This folder stores the system library for the TW850 and data files for performance analysis/tuning.
- 6) Program Files\NEC Electronics Tools\TW850\version\lib850\r22
This folder stores the system library for the TW850 (22-register mode) and data files for performance analysis/tuning (22-register mode).
- 7) Program Files\NEC Electronics Tools\TW850\version\lib850\r26
This folder stores the system library for the TW850 (26-register mode) and data files for performance analysis/tuning (26-register mode).
- 8) Program Files\NEC Electronics Tools\TW850\version\lib850\r32
This folder stores the system library for the TW850 (32-register mode) and data files for performance analysis/tuning (32-register mode).
- 9) Program Files\NEC Electronics Tools\TW850\version\lib850\r32msk
This folder stores the system library for the TW850 (32-register mask mode) and data files for performance analysis/tuning (32-register mask mode).

10) Program Files\NEC Electronics Tools\TW850\version\smp850\tw

This folder stores the sample program of the timer control code block.

For details of the timer control code block, refer to "[4.4 Timer Control Code Block](#)".

TWTIMER85.S: Timer control code block for V853 TM11

TWTIMER85E.S: Timer control code block for V850E/MS1 TM10

11) Documents and Settings\All Users\Application Data\NEC Electronics\TW850\version

This folder stores the files to save the common information in the TW850.

TW850.INI: The files to save the common information in the TW850.

Remark The folder "Documents and Settings" saves the setting information of each user. This folder differs depending on the OS to be used. (The above is the example in the case of Windows XP.)

CHAPTER 3 STARTING AND EXITING

This chapter explains how to start and exit the TW850.

Before starting the TW850, check the following matters.

- Java execution environment

A Java execution environment is necessary for starting the TW850.

The TW850 searches for a Java application start command (JAVAW.EXE) in the following sequence.

In the case of NEC Electronics tool-chain-supporting versions, however, the installation directory of the corresponding Java runtime environment is added to the beginning of the environment variable PATH in the default condition.

- 1) Folder specified by environment variable PATH
- 2) Windows system folder
- 3) Windows folder

- NEC Electronics tool chain

To start the TW850 with the NEC Electronics tool chain, the project that is used with the TW850 must be correctly built and debugged with the PM plus, a debugger or a simulator.

The device file to be used with the project that is used with the TW850 must be installed at the specified location by the device file installer.

3.1 Starting Method

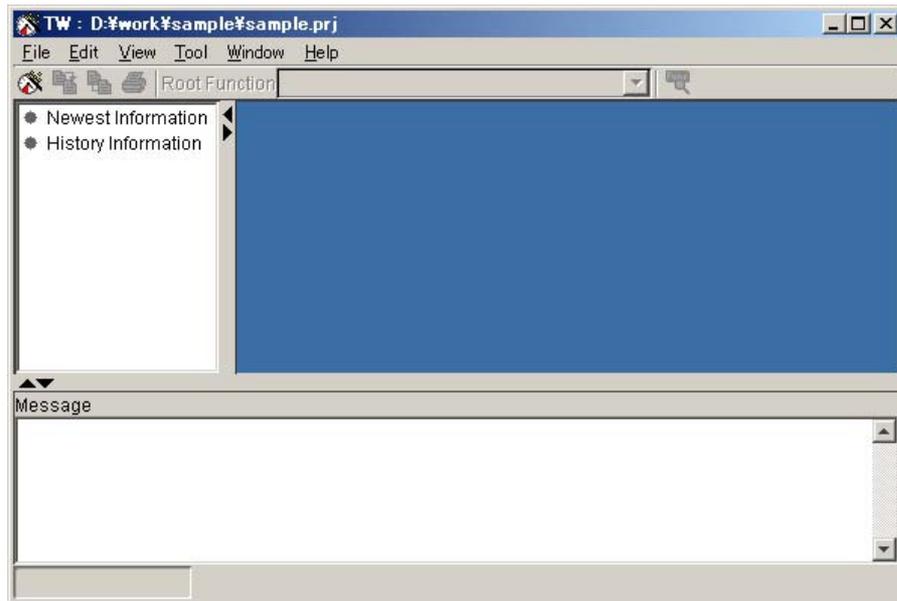
3.1.1 Starting from PM plus

This section explains how to start the TW850 from the menu bar of the PM plus.

When the TW850 has been correctly started, the [Main Window](#) shown in [Figure 3-1](#) is automatically opened.

- 1) Select the [Tool] menu -> [Startup TW850].

Figure 3-1 Main Window



- Remarks 1. Specify the start option in the [TW850 Options Dialog Box](#).
2. If the <Startup TW850> button on the tool bar of the PM plus is clicked, processing equivalent to the above operation is executed.

3.1.2 Start options

Specify the start options of the TW850 in the [Other Options] specification area of the [TW850 Options Dialog Box](#). The start options of the TW850 are as follows.

- `-Dtw.NWireRealTimeTrace`
Specify that the debugger is executed in the real-time mode when trace is set to on for N-Wire trace.
The TW850 executes processing assuming that non-real-time mode is specified.
- `-Dtw.DebuggerTimeout=DbgTmout`
Specify the response wait time (in seconds) during which the TW850 prepares for operation after debugger processing is started.
The value that can be specified as *DbgTmout* is limited to 1 to 3600.
If this option is omitted, the TW850 executes processing assuming that `-Dtw.DebuggerTimeout=300` is specified.
- `-Dtw.IgnoreJavaVersionWaning`
Specify that the [Message Dialog Box](#) that is displayed if the version of JavaVM is later than that supported by the TW850 not be displayed.
If this option is omitted, the TW850 executes processing assuming that `-Dtw.IgnoreJavaVersionWaning` is not specified.

3.2 Exiting Method

3.2.1 Exiting form menu bar

This section explains how to exit the TW850 from the menu bar of the [Main Window](#).
When the TW850 has been correctly exited, the [Verification Dialog Box](#) shown in [Figure 3-2](#) is automatically opened.

- 1) Select [File] menu -> [Exit...].
- 2) Click the <Yes> button on the [Verification Dialog Box](#).

Figure 3-2 Verification Dialog Box



3.2.2 Exiting form title bar

This section explains how to exit the TW850 from the title bar of the [Main Window](#).
When the TW850 has been correctly exited, the [Verification Dialog Box](#) is automatically opened.

- 1) Click the <Close> button.
- 2) Click the <Yes> button on the [Verification Dialog Box](#).

CHAPTER 4 FUNCTIONS

This chapter explains performance analysis, tuning, and GUI functions supplied by the TW850.

4.1 Performance Analysis Functions

This section describes the following items.

- [Performance analysis target](#)
- [Methods for acquiring performance analysis information](#)
- [Details of performance analysis functions](#)
- [Notes on performance analysis processing](#)

4.1.1 Performance analysis target

The function, project, and target environment subject to performance analysis are as follows.

- Function allocated to ".text" section

The TW850 analyzes the performance of a function allocated to the ".text" section.

The TW850 converts a user program in C language into an assembly language, and then acquires and analyzes the performance analysis information. Therefore, inline expansion functions in the user program in C language are not subject to performance analysis by the TW850. A function that is not assigned to a section is assigned to the ".text" section.

- Function in the first image area of the internal ROM

If the function subject to performance analysis is in the second or subsequent image area of the internal ROM, the TW850 cannot correctly analyze the performance.

- Project that a function subject to performance analysis exists

The TW850 does not handle a project that no function subject to performance analysis exists.

- Project that does not require the user operation

The TW850 does not handle a project that requires the user operation between when the debugger is started and when an execution module is executed because it automatically starts the debugger, and downloads and executes the execution module.

If there is a part that requires the user operation within the range subject to analysis, the result of analysis may change by the operation. Consequently, the project cannot be tuned correctly.

If the user operation can be executed by describing a hook procedure in the script file, however, the TW850 can handle the project.

The following is an example that the internal instruction RAM of the V850E/ME2 is used.

To use the internal instruction RAM of the V850E/ME2, it is necessary to set the internal instruction RAM in the write mode before the execution module is downloaded and set the internal instruction RAM in the read mode in the read mode after the CPU is reset.

Create the following script file (execution module name).tcl in the same folder as the execution module of the project to be generated. If (execution module name).tcl already exists, add the following script.

```
proc BeforeDownload {} {
    # Set the internal instruction RAM in the write mode before downloading.
    register IRAMM 0x3
}

proc AfterCpuReset {} {
    # Set the internal instruction RAM in the read mode after the CPU is reset.
    register IRAMM 0x0
}
```

The TW850 resets the CPU after the execution module of the project to be used has been downloaded. By using this script, therefore, the internal instruction RAM can be set in the write mode via a hook before downloading and in the read mode via a hook after the CPU has been reset.

In addition, the script file can be executed just when the TW850 is executed if the script file name is changed to tw.tcl.

- Project that does not require downloading of multiple execution modules

The TW850 does not handle a project that requires downloading of multiple execution modules to execute the execution module.

If multiple execution modules can be downloaded by describing a hook procedure in the script file, however, the TW850 can handle the project.

The following is an example that the execution module of the project is module.out and the execution module that must be downloaded along with module.out when the module is executed is submodule.out.

Create the following script file module.tcl in the same folder as the execution module of the project to be generated. If module.tcl already exists, add the following script.

```
proc AfterCpuReset {} {
    # Download in addition after the CPU is reset
    download -append submodule.out
}
```

The TW850 resets the CPU after the execution module of the project to be used has been downloaded. By using this script, therefore, submodule.out can be downloaded in addition via a hook after the CPU has been reset.

In addition, the script file can be executed just when the TW850 is executed if the script file name is changed to tw.tcl.

- Target environment that operates correctly when the CPU is reset

The TW850 cannot correctly analyze the performance in a target environment that does not operate correctly if the CPU is reset because it resets the CPU after the execution module has been downloaded.

If the target environment can be operated correctly after the CPU is reset by describing a hook procedure in the script file, however, the TW850 can handle the project.

Examples are shown below.

Create the following script file (execution module name).tcl described the initialization processing necessary for execution after the CPU is reset in the same folder as the execution module of the project to be generated.

```
proc AfterCpuReset {} {
    # Describe here the initialization processing necessary for execution after the CPU is reset.
    .....
}
```

By using this script, the TW850 can execute the initialization processing necessary for execution after the CPU is reset via a hook after the CPU has been reset.

In addition, the script file can be executed just when the TW850 is executed if the script file name is changed to tw.tcl.

4.1.2 Methods for acquiring performance analysis information

The TW850 supplies the following two methods for acquiring performance analysis information.

- [Trace data analysis method](#)
- [Probe code insertion method](#)

1) Trace data analysis method

By trace data analysis method, performance analysis information is acquired by analyzing the hardware trace data output by an execution engine (such as a debugger or simulator) when an execution module is executed. In addition, this method can tune based on this information.

By this method, the execution engine must be able to acquire all trace data or N-Wire trace data. When the N-Wire trace data is acquired (except when `-Dtw.NwireRealTimeTrace` is specified as the start option of the TW850 with a CPU that can set a real-time mode to acquire the N-Wire trace data), the execution speed of the program decreases while the trace data is output, as a functional characteristic of the CPU. As a result, the execution time of the TW850 is longer than when the trace data is not output.

Cases that performance analysis cannot be performed by using the trace data analysis method are explained below.

- If the correct operation cannot be guaranteed when the trace buffer of the execution engine is full and when execution of the execution module is temporarily stopped at the performance analysis start position and end position
The TW850 temporarily stops execution of an execution module and acquires, analyzes the performance analysis information when the trace buffer of the execution engine is full (except when a trace event is specified) and at the performance analysis start and end positions. Therefore, the performance of an execution module that cannot guarantee correct operation if execution is temporarily stopped cannot be analyzed by using the trace data analysis method.

2) Probe code insertion method

By probe code insertion method, performance analysis information (the number of times of execution and the execution time of the function) is acquired by executing an execution module with measurement probe code inserted to acquire the function information at the location that the function subject to performance analysis is called. In addition, this method can tune based on this information.

This method increases the code size of the program and requires a memory area on the target that stores the analysis information of functions.

Compared with the trace data analysis method, the information that can be acquired decreases. However, this method can be used even in a target environment that trace data cannot be acquired. In addition, the program is not stopped in the analysis zone. Therefore, a zone that program execution must not be stopped can be specified as an analysis zone. Further, performance analysis and tuning can be performed faster than when the trace data analysis method is used.

Cases that performance analysis cannot be performed by using the probe code insertion method are explained below.

- The TW850 cannot recognize a function subject to performance analysis
The TW850 cannot output the callpair information of a function that it cannot recognize because the TW850 inserts a probe code in the function call part.
- If performance of a tuned execution module is analyzed
If a probe code is inserted in a tuned execution module, the performance cannot be analyzed in function units because the performance fluctuates. In the tuned execution module, only the performance of items related to the entire tuned execution module and that can be acquired by static analysis can be analyzed.
- If the contents of the register used in the probe code are lost by an interrupt
If the contents of the register used in the probe code are not saved or restored, and thus lost by an interrupt, the performance cannot be correctly analyzed because the code for profile does not operate correctly.
- If performance is analyzed before a stack pointer is initialized
The performance cannot be analyzed before a stack pointer is initialized because the stack is saved to a register in the probe code.

- If a sufficient text area and data area cannot be allocated

When the probe code is inserted, the data area that stores profile data increases. If a sufficient text area and data area cannot be allocated, therefore, the probe code insertion method cannot be used for performance analysis.

- If the correct operation cannot be guaranteed when execution of the execution module is temporarily stopped at the performance analysis start position and end position

The TW850 temporarily stops execution of the execution module at the performance analysis start and end positions, and acquires and analyzes the performance analysis information. Therefore, the performance of an execution module that cannot guarantee correct operation if execution is temporarily stopped cannot be analyzed by using the probe code insertion method.

- If branching of an instruction that cannot branch to the entire address space exceeds the branch limit of the instruction

Branching of an instruction that cannot branch to the entire address space may exceed the branch limit of the instruction by an increase in the text size by the probe code inserted by the TW850. Therefore, an execution module that calls such a function cannot be analyzed by using the probe code insertion method.

This can be solved by using far jump for a function call, declaring the calling function and called function close together, or shortening the branch distance. With a C source file, a compilation option that uses far jump for a function call can be specified.

With the probe code insertion method, the number of times of calling callpairs and the execution time of each function are stored in the profile table on the memory. The memory capacity (unit: bytes) to be used when the probe code insertion method is used is shown below.

Remark Because the hole size is not included in the calculated size, the memory capacity actually occupied increases by the hole size.

- The memory capacity of the stack is calculated by the following expression.

Up to $24 + 20 * \text{Interrupt nesting level of program}$

Remark The interrupt nesting level is calculated as 0 if there are no interrupts, as 1 if an interrupt that does not enable multiple interrupts is used, and as 2 or more if an interrupt that enables multiple interrupts is used.

- The method of calculating the memory to be used is shown below.

[When only the number of times of execution is measured]

Memory to be used =
 Number of callpairs * 4 +
 Number of functions with dynamic call * Number of functions +
 14 to 3200^{Note}

[When RTOS is not used and the number of times of execution and the time are measured]

Memory to be used =
 Number of static callpairs * 4 +
 Number of functions with dynamic call * Number of functions * 4 +
 Number of functions * 4 +
 26 to 3200^{Note}

If system library functions for the TW850 is not used in the program that the function main calls func1 and func2, the memory capacity is calculated as follows.

Memory to be used = $2 * 4 + 0 * 3 * 4 + 3 * 4 + 26 = 46$

[When RX850 is used and the number of times of execution and the time are measured]

Memory to be used =
 Number of static callpairs * 4 +
 Number of functions with dynamic call * Number of functions * 4 +
 Number of functions * 4 +
 (User-specified maximum task ID + 1) * 2 +
 102 to 3200^{Note}

[When RX850 Pro is used and the number of times of execution and the time are measured]

Memory to be used =

Number of static callpairs * 4 +
 Number of functions with dynamic call * Number of functions * 4 +
 Number of functions * 4 +
 (User-specified maximum task ID + 1) * 2 +
 70 to 3200^{Note}

Note The memory to be used differs depending on how system libraries for the TW850 are used.

- Remarks 1. The number of static callpairs includes the system library for the TW850, except callpairs whose parent function has a dynamic call.
2. The number of functions includes the system library for the TW850.

4.1.3 Details of performance analysis functions

The TW850 supplies the following performance analysis functions.

- [Acquiring memory allocation information](#)
- [Acquiring callpair information](#)
- [Acquiring call count information](#)
- [Acquiring execution time information](#)
- [Acquiring instruction cache information](#)
- [Acquiring RTOS information](#)

1) Acquiring memory allocation information

Memory allocation information is acquired in function units by statically analyzing a user program or an execution module.

The memory allocation information that has been acquired can be referenced from each window (such as the [Function Detail Information Window](#)).

2) Acquiring callpair information

Callpair information (calling relationship between functions) is acquired in function units by statically analyzing a user program or an execution module.

The callpair information that has been acquired can be referenced from each window (such as the [Callgraph Window](#) or [Function Detail Information Window](#)).

3) Acquiring call count information

Call count information (which function has called which function how many times) is acquired in function units by analyzing hardware trace data output by an execution engine (such as a debugger or simulator) when an execution module is executed, or by executing an execution module with probe code for measurement inserted at the calling location of a function subject to performance analysis.

The call count information that has been acquired can be referenced from each window (such as the [Callgraph Window](#), [Function Information Table Window](#), and [Function Detail Information Window](#)).

4) Acquiring execution time information

Execution time information is acquired in function units by analyzing hardware trace data output by an execution engine (such as a debugger or simulator) when an execution module is executed, or by executing an execution module with probe code for measurement inserted at the calling and returning location of a function subject to performance analysis.

The execution time information that has been acquired can be referenced from each window (such as the [Function Information Table Window](#) or [Function Detail Information Window](#)).

5) Acquiring instruction cache information

Instruction cache information (such as the number of instruction cache hits or misses) is acquired in function units by analyzing the hardware trace data (all trace data) output by an execution engine (such as a debugger or simulator) when an execution module is executed.

The instruction cache information that has been acquired can be referenced from each window (such as the [Execution Module Information Window](#), [Function Information Table Window](#), and [Function Detail Information Window](#)).

6) Acquiring RTOS information

RTOS information (such as the number of times of execution and the execution time) is acquired by analyzing hardware trace data output by an execution engine (such as a debugger or simulator) when an execution module is executed, or by executing an execution module with measurement probe code inserted at the calling and returning location of the function supplied by the RTOS (RX850 or RX850 Pro).

- Number of times of execution

The TW850 counts up the number of times of execution when execution has passed the following locations.

- Entrance of extended SVC handler
- Entrance of system call

- Execution time

The TW850 starts measuring the execution time when execution has passed the following locations.

- Entrance of extended SVC handler
- Entrance of system call
- Entrance of scheduling processing

The TW850 stops measuring the execution time when execution has passed the following locations.

- Exit of system call
- Entrance of interrupt handler
- Exit of interrupt handler
- Entrance of idle handler

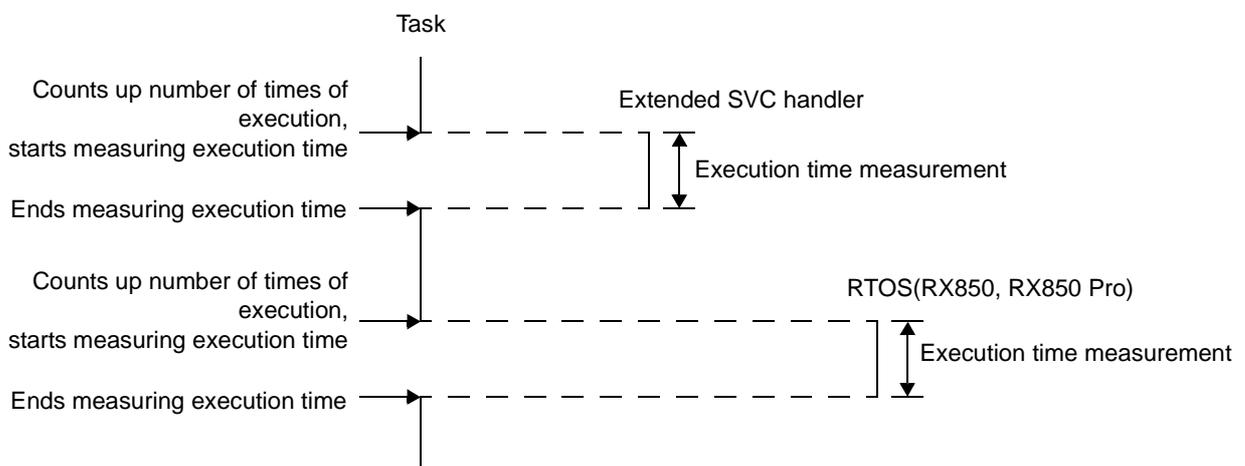
- GROSS time

The execution time of a system call is not added to the GROSS time of a task during trace analysis.

The RTOS information that has been acquired can be referenced from the [Function Information Table Window](#).

[Figure 4-1](#) shows the image of updating the RTOS information.

Figure 4-1 Image of Updating RTOS Information



- Cautions 1. If the probe code insertion method is specified to acquire performance analysis information when a user program or an execution module using the functions supplied by the RTOS is analyzed, the performance of the function that was called before initialization processing of the RTOS has been completed cannot be analyzed.
2. Inside the extended SVC handler is not subject to trace analysis.

4.1.4 Notes on performance analysis processing

The following points must be noted when executing performance analysis processing.

- When the performance is analyzed by the probe code insertion method, the TW850 generates the following segments, to which it allocates items such as a routine for profile.

TWROUTINE: Segment for profile routine
 TWTABLE: Segment for profile table

- To perform "time measurement processing using the probe code insertion method" when the performance of the user program or an execution module is analyzed, a processing routine (timer control code block) that initializes the timer/counter so that it is used with the TW850 is necessary.

For details of the timer control code block, refer to "[4.4 Timer Control Code Block](#)".

- When a function is dynamically called, the TW850 generates a callpair only for a function whose calling relationship is actually established.

Because the callpair is erased if the project is changed, however, the number of times of execution may be 0 even if a callpair exists.

For example, when func2 is dynamically called by func1 the first time the performance is analyzed, and that func2 is not called from func1 the second time, a callpair that calls func2 from func1 exists, but the number of times of execution is 0.

- Generating a callpair depends on the method to acquire performance analysis information.

[With trace data analysis method]

If processing is transferred from a function that the TW850 recognizes as executing a dynamic call to the beginning of another function, a callpair is generated.

[With probe code insertion method]

If a function calls another function using a method that the TW850 can recognize a dynamic call (such as by using an interrupt), a callpair is generated.

- If an ICE server is used, the correct execution time of a trace frame immediately after program execution has been resumed cannot be correctly acquired. If several frames are missed when tracing is stopped, the correct execution time of that part cannot be obtained.
- If the performance of an assembly source file is to be analyzed, the TW850 may not recognize some descriptions. Refer to the "[4.6.3 Description format of assembly source file](#)" for details on an assembly source subject to performance analysis.

4.2 Tuning Functions

This section describes the following items.

- [Tuning target](#)
- [Details of tuning functions](#)
- [Notes on tuning processing](#)

4.2.1 Tuning target

The function, and project subject to tuning are as follows.

- Function allocated to ".text" section

The TW850 tunes a function allocated to the ".text" section.

The TW850 converts a user program in C language into an assembly language, and then acquires and tunes the tuning information. Therefore, inline expansion functions in the user program in C language are not subject to tuning by the TW850. A function that is not assigned to a section is assigned to the ".text" section.

- Function that the size is smaller than the vacant size of the fast access memory area or the size of the instruction cache memory

If the size of a function is larger than the vacant size of the fast access memory area or the size of the instruction cache memory, that function is not tuned.

Such a function can be tuned if its size is reduced by making a subroutine of the function or dividing the function.

- Function in the first image area of the internal ROM

If the function subject to tuning is in the second or subsequent image area of the internal ROM, the TW850 cannot correctly tune.

- Project that a function subject to tuning

The TW850 does not handle a project that no function subject to tuning exists.

- Project that does not require the user operation

The TW850 does not handle a project that requires the user operation between when the debugger is started and when an execution module is executed because it automatically starts the debugger, and downloads and executes the execution module.

If there is a part that requires the user operation within the range subject to analysis, the result of analysis may change by the operation. Consequently, the project cannot be tuned correctly.

If the the TW850 the user operation can be executed by describing a hook procedure in the script file, however, the the TW850 can handle the project.

The following is an example that the internal instruction RAM of the V850E/ME2 is used.

To use the internal instruction RAM of the V850E/ME2, it is necessary to set the internal instruction RAM in the write mode before the execution module is downloaded and set the internal instruction RAM in the read mode in the read mode after the CPU is reset.

Create the following script file (execution module name).tcl in the same folder as the execution module of the project to be generated. If (module name).tcl already exists, add the following script.

```

proc BeforeDownload {} {
    # Set the internal instruction RAM in the write mode before downloading.
    register IRAMM 0x3
}

proc AfterCpuReset {} {
    # Set the internal instruction RAM in the read mode after the CPU is reset.
    register IRAMM 0x0
}

```

The TW850 resets the CPU after the execution module of the project to be used has been downloaded. By using this script, therefore, the internal instruction RAM can be set in the write mode via a hook before downloading and in the read mode via a hook after the CPU has been reset.

In addition, the script file can be executed just when the TW850 is executed if the script file name is changed to tw.tcl.

4.2.2 Details of tuning functions

The TW850 supplies the following tuning functions.

- [Tuning of fast access memory area](#)
- [Tuning of instruction cache](#)
- [Correcting callpair information](#)
- [Specifying function to be given priority and allocated](#)

Remark If "tuning of instruction cache" and "tuning of fast access memory area" are specified at the same time, the TW850 tunes the user program and execution memory in the following order.

Tuning of fast access memory area > Tuning of instruction cache

1) Tuning of fast access memory area

With the TW850, a memory area of the target system that can be accessed at high speed is called a fast access memory area. The program can be sped up by allocating frequently used functions and functions that must be executed at high speeds to this fast access memory area. However, manually selecting that function increases the burden on the user and becomes more difficult as the program scale increases.

The TW850 has a tuning function that the execution performance of an execution module is improved by giving priority to and allocating a function that causes the execution performance of an execution module to drop to a fast access memory area (such as an internal ROM).

By using this tuning function, programming can be performed without the user having to be aware of the existence of the fast access memory area. In addition, the method of allocating the function to the fast access memory can be easily changed, so that various allocation methods can be used to try and increase the speed.

The TW850 supplies the following three methods for allocating functions to the fast access memory area (a method giving priority to a function that often calls a callpair, one giving priority to a function that takes time to be executed, and one giving priority to a function with many instruction cache misses).

- Function allocation method giving priority to a function that often calls a callpair

Of the functions that call another function or are called many times (calling function or called function), the one with the smallest code size is given priority and allocated to the fast access memory area, based on the callpair information of each function (calling function or called function).

- Function allocation method giving priority to a function that takes time to be executed

Based on execution time information of each function, a function that takes time to execute one instruction is given priority and allocated to the fast access memory area.

This function allocation method cannot be specified if the "probe code analysis method" is specified and "measurement of execution time of a function" is not specified when the performance of the user program or an execution module is analyzed.

- Function allocation method giving priority to a function with many instruction cache misses

This option gives priority to the function with many instruction cache misses when allocating functions to the fast access memory area, based on the instruction cache information of each function (the number of instruction cache hits and the number of instruction cache misses).

This function allocation method cannot be specified if the "probe code analysis method" is specified to acquire performance analysis information when the performance of the user program or an execution module is analyzed, or if no instruction cache miss information can be acquired.

Remark When the TW850 tunes a fast access memory area, functions are not allocated to a fast access memory area already used by a program or data. If functions are allocated to a fast access memory area by a link directive file, etc., they are allocated to a fast access memory area in a range that starts from the function immediately after the one closest to the last address and finishes at the end of the fast access memory area. If an NMI interrupt vector area, security ID code storage area, or flash mask option area is included in the range that functions are to be allocated, however, functions are allocated to a range from immediately after the end of that area to the end of the fast access memory area. If the fast access memory area cannot be efficiently tuned, decrease the functions allocated to the fast access memory area before using the TW850. Then the TW850 can execute tuning by effectively using the vacant fast access memory area. If no function is allocated to the fast access memory area, the TW850 can execute tuning using the fast access memory area more effectively.

2) Tuning of instruction cache

The instruction cache that can be accessed at high speeds is smaller than the program in most cases. If a program is executed by a processor equipped with an instruction cache, therefore, the program loaded to the cache before is removed and the program being executed at that time is loaded. If no program to be executed is found in the instruction cache at this time, this is called an instruction cache miss. It takes a long time to exchange the program on the instruction cache and it takes the programmer a lot of effort to allocate functions in order to decrease instruction cache misses.

The TW850 has a tuning function that the execution performance of an execution module can be improved by allocating each function so that the number of instruction cache miss decreases.

By using this tuning function, programmer can easily changed the function allocation algorithm and the option related to the algorithm, so that various allocation methods can be used to try and increase the speed.

The TW850 supplies the following two methods for allocating functions to the instruction cache (coloring algorithm and grouping algorithm).

- Coloring algorithm

The coloring algorithm allocates functions, based on the callpair information (calling relationship between functions) of each function, so that instruction cache misses among all callpairs can be reduced in as a wide range as possible.

- Grouping algorithm

The grouping algorithm groups functions in a close calling relationship and allocates them so that instruction cache misses are decreased within the group, with the size of the instruction cache as the upper-limit value of the group size, based on the callpair information (calling relationship between functions) of each function.

The TW850 supplies the following two options to correct the priority of allocation to the instruction cache (an option giving priority to the number of times of calling and an option giving priority to depth).

- Option giving priority to the number of times of calling

This option gives priority to the function that has been called most times when allocating functions to the instruction cache, based on the callpair information of each function (calling relationship between functions).

- Option giving priority to the depth

This option gives priority to the function closest to the end of the calling relationship when allocating functions to the instruction cache, based on the callpair information of each function (calling relationship between functions).

Remark With the TW850, an option "considering the number of ways" can be also specified if the cache mechanism has two or more ways.

If the "considering the number of ways" option is specified, overlapping of two or more functions on the cache can be permitted, so that functions can be ideally allocated, compared with an option that "gives priority to the number of times of calling" and an option that "gives priority to the depth".

3) Correcting callpair information

The improvement rate of tuning processing (tuning of a fast access memory area and tuning of an instruction cache) is raised by correcting a callpair information (calling relationship between functions) of each function. The TW850 supplies the following three methods (float runtime library, function corresponding to a hot path, and function in loop processing) for correcting the callpair information.

- Float runtime library

The execution performance of an execution module is improved by correcting the number of times that a callpair related to the float runtime library functions (library functions used by the compiler when a floating-point operation is performed) is called.

- Function corresponding to a hot path

The execution performance of an execution module is improved by correcting the number of times a callpair related to the function, based on execution time (GROSS time) information including the child function of the function.

This correction method cannot be specified if "trace data analysis method" is not specified when the performance of the user program or an execution module is analyzed.

- Function in loop processing

The execution performance of an execution module is improved by correcting the number of times a callpair related to the same loop processing is called.

Remark If "float runtime library", "function corresponding to a hot path", and "function in loop processing" are specified at the same time, the TW850 corrects callpair information in the following order.

Float runtime library > Function corresponding to a hot path > Function in loop processing

4) Specifying function to be given priority and allocated

The improvement rate of tuning processing (tuning of a fast access memory area and tuning of an instruction cache) is raised if the user specifies a function that is given priority and tuned.

If "a function to be given priority and allocated" is specified, the TW850 basically assumes this function as a function with the highest priority for tuning of a fast access memory area.

Remark If the TW850 cannot allocate the function to the fast access memory area, it treats it as a function given the highest priority when tuning the instruction cache.

4.2.3 Notes on tuning processing

The following points must be noted when executing tuning processing.

- Branching of an instruction that cannot branch to the entire address space may exceed the branch limit of the instruction because the TW850 does not take into consideration the distance of a function when it relocates the function. Therefore, an execution module that calls such a function cannot be tuned.

This can be solved by using far jump for a function call. With a C source file, a compilation option that uses far jump for a function call can be specified.

- Unless the user specifies a segment, the TW850 generates the following segments and it allocates the functions subject to relocation to the segment.

TWTEXT: Segment for function allocation

TWFAMEM: Segment for fast access memory function allocation

If the TWTEXT segment is in the link directive file of a project, the TW850 can allocate a function to that segment.

If the TWFAMEM segment exists, a tuned execution module cannot be built.

- If an assembly source file is to be tuned, the TW850 may not recognize some descriptions.

Refer to the "[4.6.3 Description format of assembly source file](#)" for details on an assembly source.

4.3 GUI Functions

This section describes the following items.

- [Details of GUI functions](#)

4.3.1 Details of GUI functions

The TW850 supplies the following GUI functions.

- [Setting in wizard format](#)
- [Display of performance analysis and tuning results](#)

1) **Setting in wizard format**

Graphical guidance to the settings (such as the performance analysis zone and execution environment) required for using performance analysis and tuning functions supplied by the TW850 is provided in a wizard format so that the settings can be made easily and efficiently.

2) **Display of performance analysis and tuning results**

By graphically displaying results of performance analysis and tuning by the TW850 so that they can be checked through simple perusal, these results can be easily compared and evaluated.

The results of performance analysis and tuning can be referenced from each window (such as the ([Execution Module Information Window](#), [Callgraph Window](#), [Function Information Table Window](#), [Function Transition Information Window](#), and [Function Detail Information Window](#))).

4.4 Timer Control Code Block

The TW850 uses one timer/counter on the target system for time measurement processing using the probe code insertion method.

To perform "time measurement processing using the probe code insertion method" when the performance of the user program or an execution module is analyzed, a processing routine (timer control code block) that initializes the timer/counter so that it is used with the TW850 is necessary.

Caution If time measurement is not performed when the probe code insertion method is used, the timer control code block is not necessary because the timer/counter on the target system is not used.

This section describes the following items.

- [Creating timer control code block](#)
- [Inspecting timer control code block](#)
- [Notes on time measurement processing](#)

4.4.1 Creating timer control code block

Create a timer control code block in accordance with the target used.

Remark The TW850 supplies a sample program of the timer control code block.

- Program Files\NEC Electronics Tools\TW850\version\smp850\tw\TWTIMER85.S
This is the timer control code block for V853 TM11.
- Program Files\NEC Electronics Tools\TW850\version\smp850\tw\TWTIMER85E.S
This is the timer control code block for V850E/MS1 TM10.

The configuration of the timer control code block is shown below.

```
.section ".tw_profile_routine", text

.globl tw_init_timer_timing_address
.globl tw_clock_count
.globl tw_init_timer
.globl tw_start_timer
.globl tw_stop_and_clear_timer
.globl tw_get_timer_count

.option new_fcall

.align 0x4
tw_init_timer_timing_address:
.word <Address that timer is initialized>

.align 0x4
tw_clock_count:
.word <Number of CPU clocks per timer count>

tw_init_timer:
<Initialization processing code of timer>

tw_start_timer:
<Start processing code of timer count>

tw_stop_and_clear_timer:
<Stop and clear processing code of timer count>

tw_get_timer_count:
<Acquisition processing code of timer count value>
```

Only the timers that satisfy the following conditions can be specified by the TW850.

- It can count up.
- The count can be cleared (to 0).
- It can start and stop counting up.

Even a counter that the count cannot be cleared or that counting up cannot be stopped can be specified by saving its count value to memory when counting up is started and by calculating the difference between the current count value and the saved count value when a count value is acquired.

The labels being used in timer control code block are listed below .

- `tw_init_timer_timing_address`

Specify the address the timer is initialized.

Data type: 4-byte integer format

- Remarks
1. If this label is omitted, the timer is initialized immediately before the position that performance analysis is to be started.
 2. This label is ignored when RX850 or RX850 Pro is used.

Caution Specify the address to be always executed from the execution start position to the analysis start position. In addition, specify the address that is different from the execution start position and analysis start position.

- `tw_clock_count` (required)

Specify the number of CPU clocks per timer count.

Data type: 4-byte integer format

- `tw_init_timer` (required)

Describe the initialization processing code of the timer.

Input register: lp(return address)

Output register: None

Register whose contents can be destroyed: r10, r11, r12

- Cautions
1. Instructions expanded by the assembler may destroy r1.
 2. If the timer can set division rate, make sure that the value as `tw_clock_count` matches the setting of division rate when it is initialized.

- `tw_start_timer` (required)

Describe the start processing code of the timer count.

Input register: lp(return address)

Output register: None

Register whose contents can be destroyed: r10, r11

Caution Instructions expanded by the assembler may destroy r1.

- `tw_stop_and_clear_timer` (required)

Describe the stop and clear processing code of the timer count.

Input register: lp(return address)

Output register: None

Register whose contents can be destroyed: r10, r11

Caution Instructions expanded by the assembler may destroy r1.

- `tw_get_timer_count` (required)

Describe the acquisition processing code of the timer count value.

Input register: lp(return address)

Output register: r10(count value)

Register whose contents can be destroyed: r11

- Cautions
1. Instructions expanded by the assembler may destroy r1.
 2. Make sure that a value assigned to an output register is not negative.
For example, if the target is the V850 and if the timer/counter is 2 bytes, a value must be loaded to the register by the ld.h instruction to extend the sign by the ld.h instruction, and then the register must be masked with 0xffff by using the andi instruction.
 3. If an event (such as a function call or RTOS system call) does not occur and the timer overflows or rounds during time measurement, time measurement cannot be performed correctly.

4.4.2 Inspecting timer control code block

By creating an execution module by linking the created timer control code block as the object for inspection, and by executing that module, the timer control code block can be inspected.

The following items can be inspected by this object for inspection.

- Illegal operation of timer (not operating, not stopped, or not cleared)
- Destruction of the contents of registers in each processing block (timer initialization, count start, count stop/clear, count value acquisition), other than a register whose contents can be destroyed

Remark Destruction of the register that stores the stack pointer and return address is not inspected.

- Clear setting error in the number of CPU clocks per timer count

Inspect the timer control code block in accordance with the following procedure.

1) Create an execution module for inspection

As necessary, change the contents of the link directive file, link the timer control code block and object for inspection, and the startup file to create an execution module.

To create an execution module for inspection, the following files stored in "Program Files\NEC Electronics Tools\TW850\version\smp850\tw" is necessary.

TWTMCHK85.O:	Inspection object for V850
TWTMCHK85E.O:	Inspection object for V850E
TWTMCHK85.DIR:	Sample using internal ROM/RAM of a link directive file for CA850 V850
TWTMCHK85E.DIR:	Sample using internal ROM/RAM of a link directive file for CA850 V850E
TWTMCHK85.MAK:	Sample of a make file for V850
TWTMCHK85E.MAK:	Sample of a make file for V850E
TWTMCHK85.PRJ:	Sample of a PM plus project file for V850
TWTMCHK85E.PRJ:	Sample of a PM plus project file for V850E

Cautions 1. The following processing must be described in a file when the startup file is customized.

- Allocating 512 bytes or more of the stack area
 - Initialization of stack pointer
 - Environment setting and calling _main function after the above processing
2. When using the sample of a PM plus project file, if the folder of the sample is used as is, the sample of the make file is overwritten during building. Therefore, copy the necessary files to another folder.

Remark The 32-bit Motorola S type hex format is specified as the format of a hex file in the sample of a PM plus project file for the V850E. This is to prevent display of a warning message that the address runs short with the 16-bit the default Intel expanded hex format.

2) Execute an execution module for inspection

Execute the created execution module for inspection and then inspect the timer control code block in the following procedure.

- (1) Download the execution module for inspection to a debugger or simulator.
- (2) Set a breakpoint at the label "tw_timer_check".
- (3) Execute the execution module for inspection to the label "tw_timer_check".
- (4) Check that the execution of the execution module for inspection has been stopped at the label "tw_timer_check".

- (5) Check the value of register (r1) of register No.1.
An error code is stored in the register of register No.1. The result of inspecting the timer control code block can be checked by the error code.

The error codes are listed below.

0x00000000:	Normal completion
0x000011XX:	Register XX (XX indicates the register number) other than a register whose contents can be destroyed is destroyed in the initialization processing of the timer. Make sure that the corresponding register is not destroyed in the initialization processing of the timer.
0x000012XX:	Register XX (XX indicates the register number) other than a destructible register is destroyed in the start processing of the timer count. Make sure that the corresponding register is not destroyed in the start processing of the timer.
0x000013XX:	Register XX (XX indicates the register number) other than a destructible register is destroyed in the stop and clear processing of the timer count. Make sure that the corresponding register is not destroyed in the stop and clear processing of the timer.
0x000014XX:	Register XX (XX indicates the register number) other than a destructible register is destroyed in the acquisition processing of the timer count value. Make sure that the corresponding register is not destroyed in the acquisition processing of the timer count value.
0x00002201:	The timer is not operated. Check the contents of the initialization processing of the timer, start processing of timer count, acquiring processing of the timer count value, and whether the number of CPU clocks per timer count has been set correctly.
0x00002301:	The timer is not stopped. Check the contents of the stop processing of the timer count in the stop and clear processing of the timer.
0x00002302:	The timer/counter is not cleared. Check the contents of the clear processing of the timer count in the stop and clear processing of the timer.
0x00003001:	The number of CPU clocks per timer count is 0. Clear the specified value of the number of CPU clocks per timer count.

- (6) If the inspection has not been correctly completed, correct the corresponding location and execute the above procedure again.

4.4.3 Notes on time measurement processing

The following points must be noted when executing time measurement processing.

- The TW850 inserts a probe code for time measurement in the return portion of a function call when the probe code insertion method is used. If a function that does not return to the calling source is called, the time of the call source and the call destination may not be correctly measured.
For example, the execution time of `longjmp` cannot be measured even if the functions `setjmp` and `longjmp` are used and `libtw` is used in a C source file. In addition, the time is added, in principle, to the function that calls `setjmp`. If `libtw` is not used, part of the execution time of the function calling `longjmp` is added to the execution time of the function calling `setjmp`.
- Time measurement is performed by calling the profile routine each time an event such as calling a function, returning, or system call hooking occurs. Therefore, if analysis is started in the middle of a function and is completed without any event occurring, it cannot be identified which function has been executed. Consequently, the execution time of the function is regarded as 0.

- The following points must be noted if an execution engine that the timer is not stopped is used when the execution has been stopped.
 - The execution time of a function specified to be analyzed or a function that analysis start and end positions belong to cannot be correctly measured. In this case, if a function calling the function is specified as a function to be analyzed, the execution time of that function can be correctly measured.
 - If another function is called in an assembly source file by a function calling instruction the TW850 cannot recognize, from a function specified to be analyzed or from the analysis start position, and an event occurs in the function that is called, the execution time of the called function cannot be correctly measured.
 - If execution returns (branches) by an instruction the TW850 cannot recognize, to a function specified to be analyzed or to the analysis end position in an assembly source file and no event occurs until the function specified to be analyzed is completed or before the analysis end position, the execution time of the recovery source function cannot be measured correctly.
- The execution time may differ each time it is measured even in the same analysis zone in the same program if the timing of counting up the timer and the timing of executing the program differ.

4.5 System Library for TW850

The TW850 relocates not only the user program but also the standard library functions (included in libc.a and libm.a) during tuning. Therefore, it supplies libtw.a instead of libc.a and libm.a. This libtw.a is necessary for generating an execution module with the TW850.

4.6 Notes on Assembly Source File

This section describes the following items.

- [Detection of functions subject to performance analysis and tuning](#)
- [Detection of location calling function](#)
- [Description format of assembly source file](#)

4.6.1 Detection of functions subject to performance analysis and tuning

Among the label declarations of the functions in the range subject to performance analysis and tuning, the TW850 recognizes the label declarations that satisfy any of the following conditions as functions subject to performance analysis or tuning.

The functions the TW850 could not recognize are not displayed on the [Function Information Table Window](#).

1) Global label declaration

The TW850 recognizes a function declared by a global label in the range subject to performance analysis or tuning as a function subject to performance analysis or tuning.

```

.globl  _func                -- global label declaration
_func:

```

2) Label declaration with function attribute

The TW850 recognizes a function declared by a label with a function attribute in the range subject to performance analysis or tuning as a function subject to performance analysis or tuning.

```

.frame  _func, 0x4          -- label declaration with function attribute
_func:

```

Even if the above conditions (global label declaration or label declaration with function attribute) are satisfied, the function declared by the label may not be recognized as a function subject to performance analysis or tuning, depending on the result of analyzing the instruction immediately before the label declaration.

1) Instruction immediately before label declaration (label declaration)

The TW850 does not recognize the function declared by the label as a function subject to performance analysis or tuning if the instruction immediately before the label declaration is a label declaration.

```

.text
.globl  _func1
_func1:

.globl  _func2
_func2:                -- Not recognized as target function

```

In this case, change the label in the function from a global label to a local label and do not set a function attribute, so that the label is not detected as a function label.

In the following example, delete the global declaration line (*) of the return label.

```

        .text
        .globl  _func1
_func1:
        mov    #_func2, r15
        mov    #_ret_label, lp
        jmp    [r15]

        .globl  _ret_label          -- Global label declaration(*)of return label
_ret_label:                          -- Return label
        jmp    [lp]

        .globl  _func2
_func2:                                -- Not recognized as target function
        jmp    [lp]

```

- 2) Instruction immediately before label declaration (other than quasi directive or unconditional branch instruction)
 The TW850 does not recognize the function declared by the label as a function subject to performance analysis or tuning if the instruction immediately before the label declaration is not a quasi directive or unconditional branch instruction.

```

        .text
        .globl  _func1
_func1:
        nop

        .globl  _func2
_func2:                                -- Not recognized as target function

```

4.6.2 Detection of location calling function

The TW850 recognizes the location of calling function that satisfies any of the following conditions as a location of calling a function when a function in the range subject to performance analysis or tuning is called.

The function calls that the TW850 could not recognize are not displayed in the [Callgraph Window](#).

1) Issuance of branch instruction (jarl)

The TW850 recognizes the location of issuing branch instruction `jarl` in the range subject to performance analysis or tuning as a location of calling a function.

```

.globl  _func1
_func1:
    mov   lp, r14
    jarl  _func2, lp           -- Issuance of branch instruction jarl

```

2) Matching of instruction patterns (arithmetic operation instruction `movea` and branch instruction `jmp`)

The TW850 recognizes the location declaring a label immediately after arithmetic operation instruction `movea` or branch instruction `jmp` in the range subject to performance analysis or tuning as a location calling a function.

```

.globl  _func
_func:
    movea xxx, xxx, r31       -- Issuance of arithmetic instruction movea
    jmp   [REGISTER]         -- Issuance of branch instruction jmp

_label1:                       -- Label declaration
    movea xxx, xxx, lp       -- Issuance of arithmetic instruction movea
    jmp   [REGISTER]         -- Issuance of branch instruction jmp

_label2:                       -- Label declaration

```

Caution `REGISTER` must be a register other than `r31` or `lp`.

3) Matching of instruction patterns (arithmetic operation instruction `mov` and branch instruction `jmp`)

The TW850 recognizes the location declaring a label immediately after arithmetic operation instruction `mov` or branch instruction `jmp` in the range subject to performance analysis or tuning as a location calling a function.

```

.globl  _func
_func:
    mov   xxx, xxx, r31       -- Issuance of arithmetic instruction mov
    jmp   [REGISTER]         -- Issuance of branch instruction jmp

_label1:                       -- Label declaration
    mov   xxx, xxx, lp       -- Issuance of arithmetic instruction mov
    jmp   [REGISTER]         -- Issuance of branch instruction jmp

_label2:                       -- Label declaration

```

Caution `REGISTER` must be a register other than `r31` or `lp`.

If the label declaration is not immediately after arithmetic operation instruction `mov` or branch instruction `jmp` in the range subject to performance analysis or tuning, the TW850 does not recognize the location as a location of calling function.

```
.text
.globl  _func1
_func1:
    mov    lp, r14
    jarl   _func2, lp
    mov    #_ret_label, lp
    mov    #_func2, r15
    jmp    [r15]                -- Issuance of branch instruction jmp
_ret_label:
    mov    #_func_end - 4, lp
    jmp    [r15]                -- Branch instruction jmp is not issued
    mov    lp, r14
    jmp    [lp]
_func_end:

.globl  _func2
_func2:
    jmp    [lp]
```

4.6.3 Description format of assembly source file

The TW850 cannot recognize assembly source files described in specific formats as subject to performance analysis or tuning.

Table 4-1 shows the assembly source files described in specific formats the TW850 can and cannot recognize as subject to performance analysis (trace data analysis method and probe code insertion method) or tuning.

Table 4-1 Restrictions on Assembly Source File

Description Format	Trace	Probe	Tuning
Accessing higher address of function label by offset	OK	NG	NG
Assembly quasi directive with changing conditions	NG	NG	NG
Formal parameter outside target range	NG	NG	NG
Defined macro call	NG	NG	NG
Multiple expansion of macro definition exceeding 32 stages	NG	NG	NG
Numeric expression or operation result in quasi directive outside valid range	-	-	-
Block declaration straddling over blocks	NG	NG	NG
Boundary of function	-	-	-
Branch to outside of function referencing label in function	OK	NG	NG
Recognition of post-declaration function label with two or more function entrances	NG	NG	NG
Offset branch instruction straddling function calling instructions	OK	NG	OK
Reference/branch of label outside function in instruction that cannot straddle sections	OK	OK	NG
Address calculation using label in different functions	OK	OK	NG
Non-initialization of stack pointer	OK	NG	OK
Use of register that is saved in case of interrupt	OK	NG	OK
Use of I/O register for usage other than that assumed	NG	NG	NG
Concatenation operator for label declaration, conditional assemble quasi directive, or symbol definition quasi directive	NG	NG	NG
Use of macro reserved by system	NG	NG	NG
Specification of function allocation sequence or function allocation address	OK	NG	NG
Referencing PSW(program status word) that straddles functions	OK	NG	OK
Using \$ macro operator	NG	NG	NG

Remark "Function" is the function that satisfies the conditions described in "4.6.1 Detection of functions subject to performance analysis and tuning".
 "Function call" is the function call that satisfies the conditions described in "4.6.2 Detection of location calling function".

Below are descriptions of the description formats in Table 4-1.

1) Accessing higher address of function label by offset

The TW850 cannot execute performance analysis processing (probe code insertion method) and tuning processing on an assembly source file that describes an access to the higher address (address close to 0x0) of a function label by an offset.

```
.text
.word    0x0
.globl   _func
_func:
mov     #_func, r12
ld.w   -4 [r12], r13           -- Accessing higher address of function label by offset
```

2) Assembly quasi directive with changing conditions

The TW850 cannot execute performance analysis processing (trace data analysis method or probe code insertion method) and tuning processing on an assembly source file that describes an assembly quasi directive (such as .if and .elseif) whose condition is changed if the instruction is repeated in the range of a repeat quasi directive (such as .repeat to .endm and .rept to .endr).

```
.set    VAR, 0x0

.repeat 0x10
.if     ( VAR == 0x5 )           -- Assembly quasi directive with changing condition
      nop
      .endif
      .set    VAR, VAR + 0x1
.endm
```

3) Formal parameter outside target range

The TW850 cannot execute performance analysis processing (trace data analysis method or probe code insertion method) and tuning processing on an assembly source file that describes a formal parameter in a macro definition quasi directive with parameters and outside the range subject to a repeat quasi directive (such as outside .macro to .endm and .repeat to .endm).

```
.macro  MCR1  ARG
      mov   ARG, r12
.endm

MCR1   0x1

.if    ARG
      nop
      .endif           -- Formal parameter outside target range
```

4) Defined macro call

The TW850 cannot execute performance analysis processing (trace data analysis method or probe code insertion method) and tuning processing on an assembly source file that describes macro call with the same name as a defined macro in the range subject to a macro definition quasi directive (.macro to .endm).

```
.macro MCR1
    .macro MCR2
        nop
    .endm

    MCR2
    nop
.endm

.macro MCR2
    MCR1                -- Defined macro call
    nop
.endm

MCR2
```

5) Multiple expansion of macro definition exceeding 32 stages

The TW850 cannot execute performance analysis processing (trace data analysis method or probe code insertion method) and tuning processing on an assembly source file in which multiple macro definitions are expanded exceeding 32 stages.

```
.macro MCR33
    nop
.endm
.macro MCR32
    MCR33
.endm
.macro MCR31
    MCR32
.endm

.....

.macro MCR3
    MCR4
.endm
.macro MCR2
    MCR3
.endm
.macro MCR1
    MCR2
.endm

MCR1                -- Multiple macro expansions exceeding 32 stages
```

6) Numeric expression or operation result in quasi directive outside valid range

With an assembly source file with a numeric expression or operation result in a quasi directive outside the valid range, the TW850 ignores the part outside the valid range (numeric expression: ignores bits other than valid bits, operation result: rounds bits other than valid bits) and executes performance analysis processing (trace data analysis method or probe code insertion method) and tuning processing.

```
.if      ( 0xffffffff == 0xffff )      -- Numeric expression in quasi directive is outside valid range
        nop
.endif
```

7) Block declaration straddling over blocks

The TW850 cannot execute performance analysis processing (trace data analysis method or probe code insertion method) and tuning processing on an assembly source file in which a specified block straddles over other blocks in a block declaration (such as a conditional assembly block declaration, repetitive definition block declaration, and macro definition block declaration).

```
.set     EVAL, 0x1

.if      EVAL
        .repeat EVAL                -- Repetitive definition block declaration straddling over other
        blocks
.endif

nop

.if      EVAL
        .endm
.endif
```

8) Boundary of function

The TW850 executes performance analysis processing (trace data analysis method or probe code insertion method) and tuning processing on an assembly source file in which an align specification quasi directive or symbol attribute setting quasi directive is described immediately before a label declaration (function symbol), assuming that the quasi directive is included in a function declared by label.

```
.align   0x4                        -- Boundary of function
.globl   _func
_func:
        nop
```

9) Branch to outside of function referencing label in function

The TW850 cannot execute performance analysis processing (probe code insertion method) and tuning processing on an assembly source file in which processing to branch outside a function by referencing a label in a function is described.

```
_func1: .globl   _func1
        jr     _func1 + 0x6        -- Branch outside function referencing label in function
        jmp    [!p]

_func2: .globl   _func2
        jmp    [!p]
```

10) Recognition of post-declaration function label with two or more function entrances

The TW850 executes performance analysis processing (trace data analysis method or probe code insertion method) and tuning processing on an assembly source file with two or more function entrances (function labels), recognizing only the function label declared first.

```

.globl  _func1
_func1:
    nop

.globl  _func2
_func2:                                -- Post-declaration function label that is not recognized
    jmp  [lp]

```

11) Offset branch instruction straddling function calling instructions

The TW850 cannot execute performance analysis processing (probe code insertion method) on an assembly source file in which an offset branch instruction straddling a function call instruction (including a function call instruction between a reference label address and branch destination address) is described.

```

.globl  _func1
_func1:
    br   _func1 + 0x6                -- Offset branch instruction straddling function call instruction
    jarl _func2, r11
    jmp  [lp]

.globl  _func2
_func2:
    jmp  [r11]

```

12) Reference/branch of label outside function in instruction that cannot straddle sections

The TW850 cannot execute tuning processing on an assembly source file using a reference/branch outside a function label in an instruction that cannot straddle a section.

```

.globl  _func1
_func1:
    jbr  #_func2                    -- Branch to label outside function
    jmp  [lp]

.globl  _func2
_func2:
    jmp  [lp]

```

13) Address calculation using label in different functions

The TW850 cannot execute tuning processing on an assembly source file in which an address calculation using a label in different functions is described.

```

_func1:  .globl  _func1
         mov   #data - #_func2, r11    -- Address calculation using label in different functions
         jmp   [lp]

         .globl  _func2
_func2:  jmp   [lp]
data:    .word  0x0

```

14) Non-initialization of stack pointer

The TW850 cannot execute performance analysis processing (probe code insertion method) on an assembly source file that is executed at a location the TW850 recognizes as a function call before the stack pointer is initialized.

15) Use of register that is saved in case of interrupt

The TW850 cannot execute performance analysis processing (probe code insertion method) on an interrupt service routine that is started without saving registers r10 to r14, which are used by the profile routine when an interrupt occurs.

16) Use of I/O register for usage other than that assumed

The TW850 limits the conditions of using the macro for I/O registers defined in the device file to the operand block of the ld/st instruction and bit manipulation instructions.

Therefore, the TW850 cannot execute performance analysis processing (trace data analysis method or probe code insertion method) and tuning processing on an assembly source file that uses the macro for I/O registers for a usage other than the above.

17) Concatenation operator for label declaration, conditional assemble quasi directive, or symbol definition quasi directive

Therefore, the TW850 cannot execute performance analysis processing (trace data analysis method or probe code insertion method) and tuning processing on an assembly source file that uses the macro for I/O registers for a usage other than the above.

```

.macro  labeldef ARG
        .globl  _func~ARG
        _func~ARG:
        -- Concatenation operator for label declaration
.endm

```

18) Use of macro reserved by system

The TW850 treats words starting with "tw_" and "__e_sysfnc" (when RX850 or RX850 Pro is used) as system-reserved macros.

Therefore, the TW850 cannot execute performance analysis processing (trace data analysis method or probe code insertion method) and tuning processing on an assembly source file that uses these words.

"_AZMON_BaseTable" (when RX850 or RX850 Pro is used) is treated as system-reserved macros.

The TW850 cannot execute a project that these words exist.

```

        .globl  tw_func                -- Use of system-reserved macro
tw_func:
        jmp   [lp]

```

19) Specification of function allocation sequence or function allocation address

The TW850 cannot execute performance analysis (probe code insertion method) and tuning processing on an assembly source file whose operation cannot be guaranteed if the allocation sequence or allocation address of a function is changed.

```

.globl  _func1
_func1:
    mov    #_func, r11          -- Specification of function allocation address
    cmp    0x10000, r11
    je     _func2
    halt

_func2:
    jmp    [!p]

```

20) Referencing PSW(program status word) that straddles functions

The TW850 cannot execute performance analysis (probe code insertion method) and tuning processing on an assembly source file that references a PSW straddling the function call destination and the function call source when a function is called or when execution is restored from a function call.

```

.globl  _func1
_func1:
    cmp    r11, r12            -- Comparison (Specification of PSW)
    jarl   _func2, r13
    jmp    [!p]

.globl  _func2
_func2:
    bne    _func2_ret         -- Refer to comparison result (PSW) after function call
    mov    r0, r11

_func2_ret:
    jmp    [r13]

```

21) Using \$ macro operator

Therefore, the TW850 cannot execute performance analysis processing (trace data analysis method or probe code insertion method) and tuning processing on an assembly source file that uses a macro using the \$ macro operator.

```

.macro  mac1    x
        mov    x, r10
.endm

.macro  mac2
        .set   value, 10
        mac1   $value    -- Specification of $value
.endm
mac2

```

CHAPTER 5 METHOD OF OPERATION

This chapter explains how to operate the TW850 and check the results of performance analysis.

Figure 5-1 shows the flow of wizard processing (performance analysis and tuning) executed by the TW850.

Figure 5-1 Flow of Wizard Processing (Performance Analysis and Tuning)

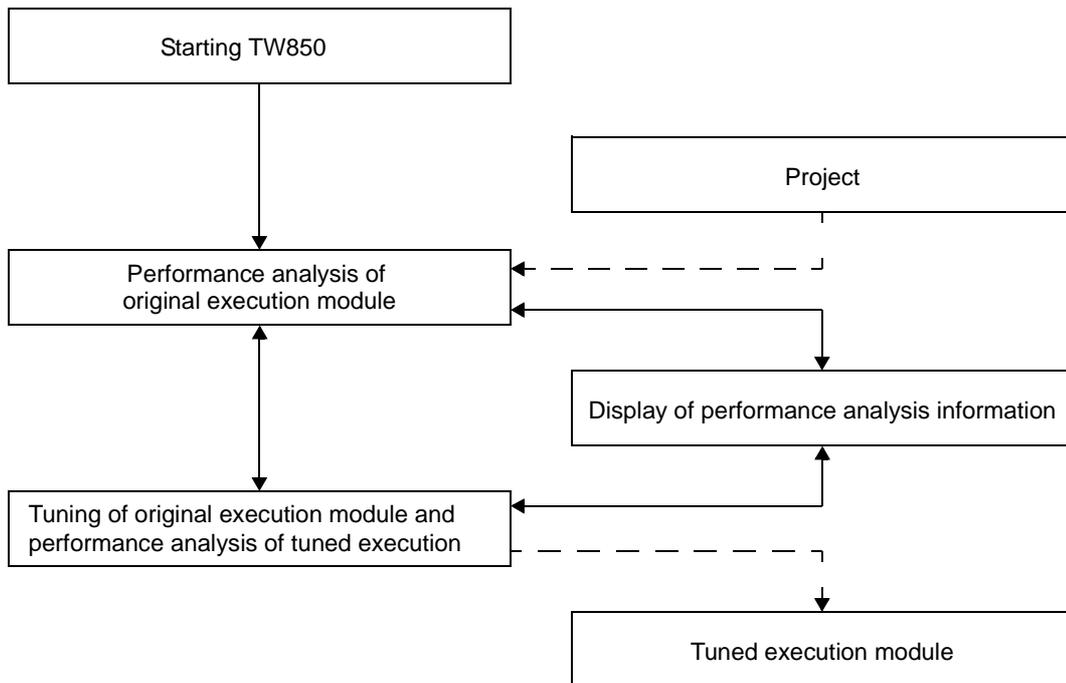
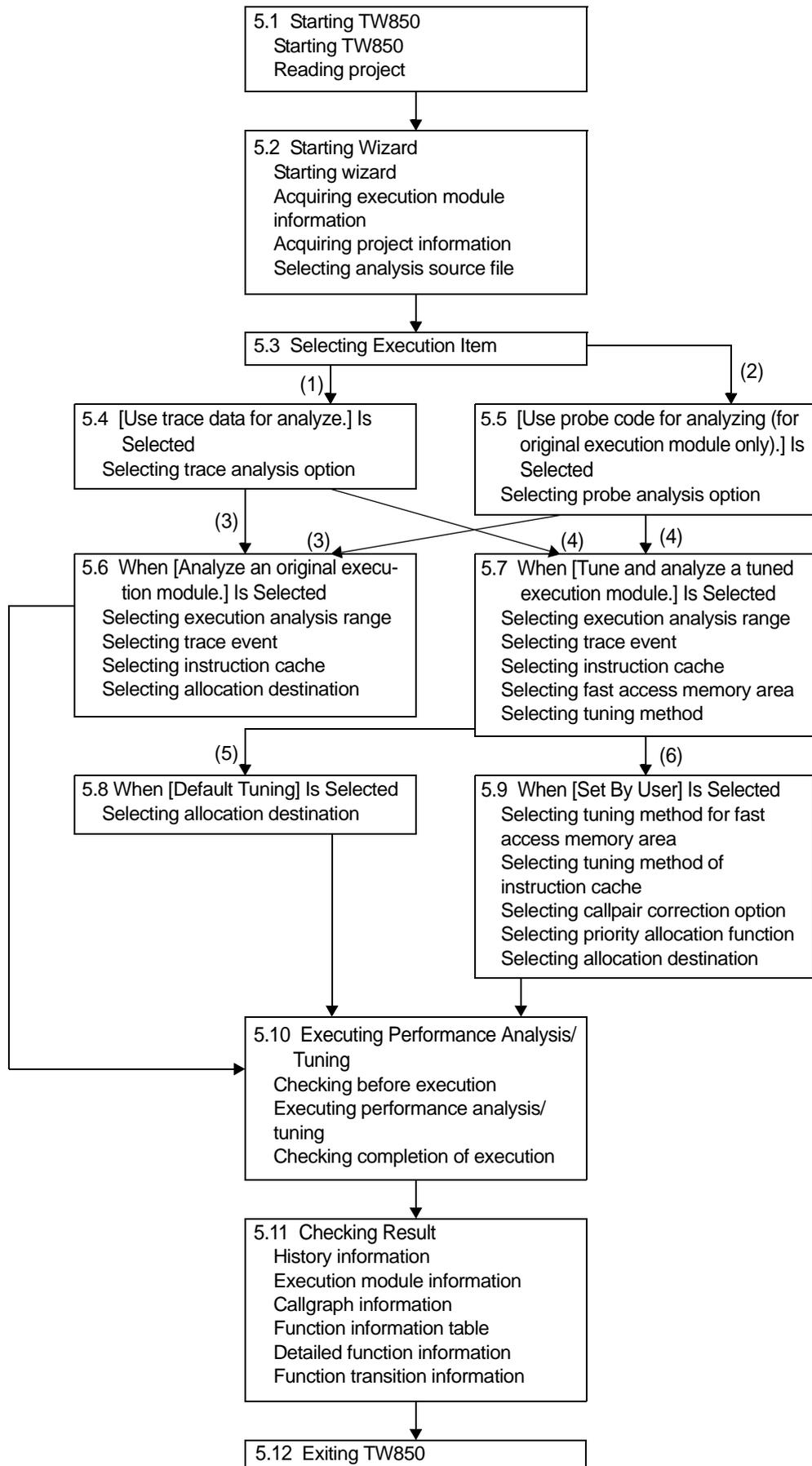


Figure 5-2 shows the procedure for operating the TW850 and checking the results of performance analysis.

Figure 5-2 Procedure for Operating the TW850/Checking the Results of Performance Analysis



- (1) If the [Use trace data for analyze.] radio button is checked when selecting an execution item
- (2) If the [Use probe code for analyzing (for original execution module only).] radio button is checked when selecting an execution item
- (3) If only the [Analyze an original execution module.] check box is checked when selecting an execution item
- (4) If the [Tune and analyze a tuned execution module.] check box is checked when selecting an execution item
- (5) If the [Default Tuning] radio button is checked when selecting a tuning method
- (6) If the [Default Tuning] radio button is checked when selecting a tuning method

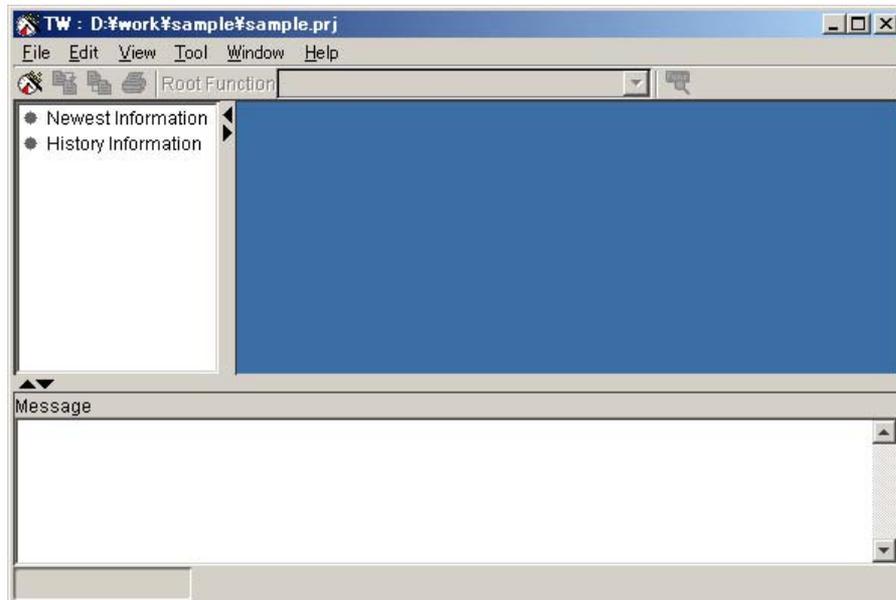
5.1 Starting TW850

5.1.1 Starting TW850

Refer to "3.1 Starting Method" for details of starting the TW850.

When the TW850 has been correctly started, the following [Main Window](#) is automatically opened.

Figure 5-3 Main Window



5.1.2 Reading project

The TW850 recognizes a project (project file) that is active in the PM plus as the project subject to performance analysis or tuning.

If a TW850 project file for the project subject to performance analysis or tuning exists when the project is changed or when the TW850 is started, that project file is read and the information (such as the execution module information) of the TW850 is restored.

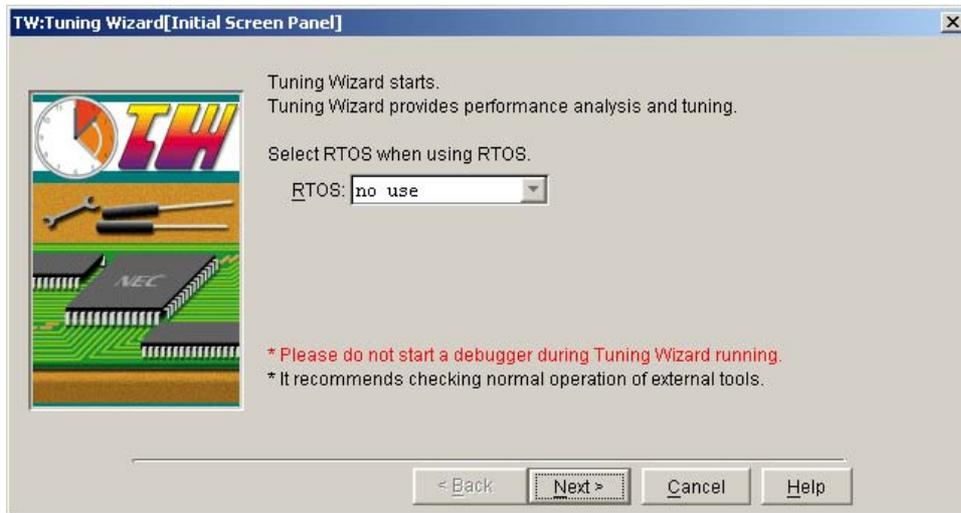
Caution If a TW850 project file of a version different from that of the TW850 currently being used is read, the [Verification Dialog Box](#) is opened so that it can be specified whether the file is to be saved as a backup file. However, a TW850 project file of later cannot be read by the TW850 of an earlier version.

5.2 Starting Wizard

5.2.1 Starting wizard

Click the Tuning Wizard button or select [Tool] menu -> [Tuning Wizard...] on the [Main Window](#).
When the TW850 has been correctly started, the following [Initial Screen Panel](#) is automatically opened.

Figure 5-4 Initial Screen Panel



To continue the wizard processing (performance analysis or tuning), check a message to report that the TW850 has started the wizard processing (performance analysis or tuning) is displayed in the message display area, and click the [<Next>](#) button.

Refer to [Initial Screen Panel](#) of WINDOW REFERENCE for details of each item.

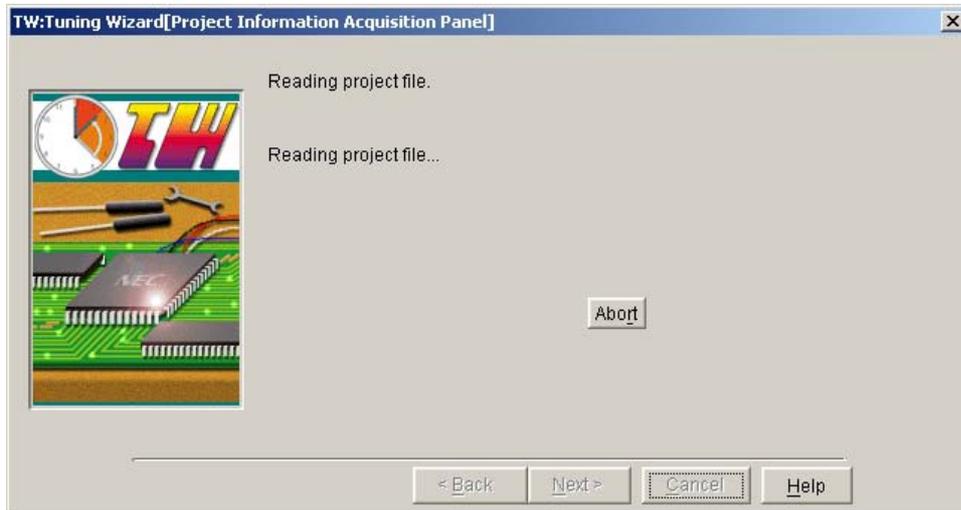
5.2.2 Acquiring project information

Read a project file and acquire project information (such as composition of source files).

However, the user does not have to be aware of this phase because it is automatically performed by the TW850 when the <Next> button on the [Initial Screen Panel](#) is clicked in "5.2.1 Starting wizard".

The [Project Information Acquisition Panel](#) that is opened while processing to acquire project information is in progress is shown below.

Figure 5-5 Project Information Acquisition Panel

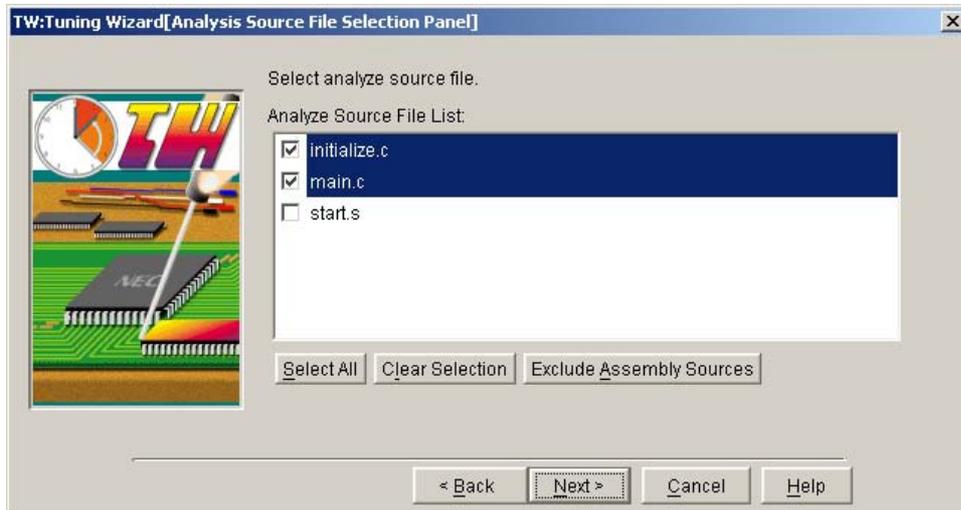


5.2.3 Selecting analysis source file

Specify source files subject to performance analysis or tuning during wizard processing (performance analysis or tuning).

The following [Analysis Source File Selection Panel](#) is automatically opened when processing to acquire project information that is executed in "5.2.2 Acquiring project information" has been correctly completed.

Figure 5-6 Analysis Source File Selection Panel



To continue wizard processing (performance analysis or tuning), specify each item, and click the <Next> button.

Refer to [Analysis Source File Selection Panel](#) of WINDOW REFERENCE for details of each item.

5.2.4 Acquiring execution module information

Acquire execution module information necessary for the TW850 to execute performance analysis processing from a user program or an execution module (execution module for original analysis) described in the project (project file) read in "5.2.1 Starting wizard".

However, the user does not have to be aware of this phase because it is automatically performed by the TW850 when the <Next> button on the [Analysis Source File Selection Panel](#) is clicked in "5.2.3 Selecting analysis source file".

The [Execution Module Information Acquisition Panel](#) that is opened while processing to acquire execution module information is in progress is shown below.

Figure 5-7 Execution Module Information Acquisition Panel

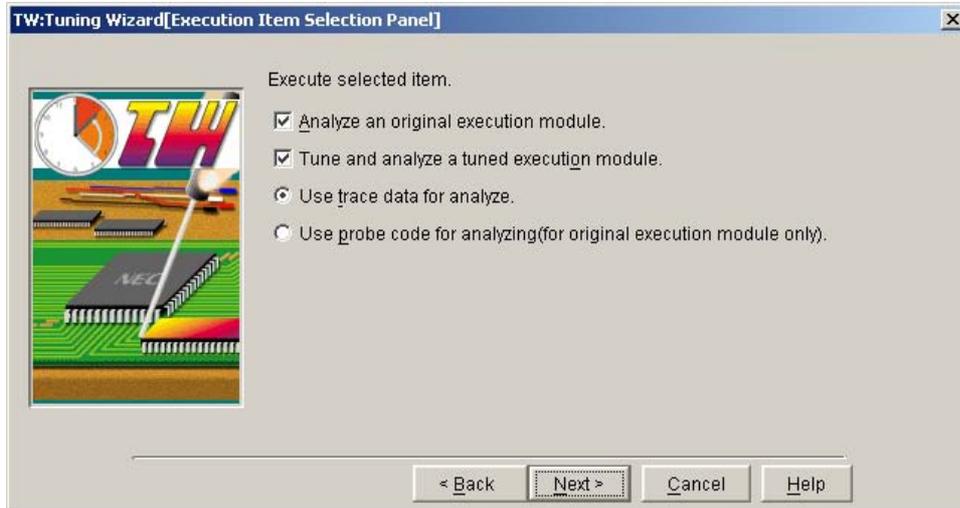


5.3 Selecting Execution Item

Specify execution items for wizard processing (performance analysis or tuning).

The following [Execution Item Selection Panel](#) is automatically opened when processing to acquire execution module information that is executed in "[5.2.4 Acquiring execution module information](#)" has been correctly completed.

Figure 5-8 Execution Item Selection Panel



To continue wizard processing (performance analysis or tuning), specify each item, and click the <Next> button.

Refer to [Execution Item Selection Panel](#) of WINDOW REFERENCE for details of each item.

5.4 When [Use trace data for analyze.] Is Selected

This section explains the procedure if the [Use trace data for analyze.] radio button is checked in "5.3 Selecting Execution Item".

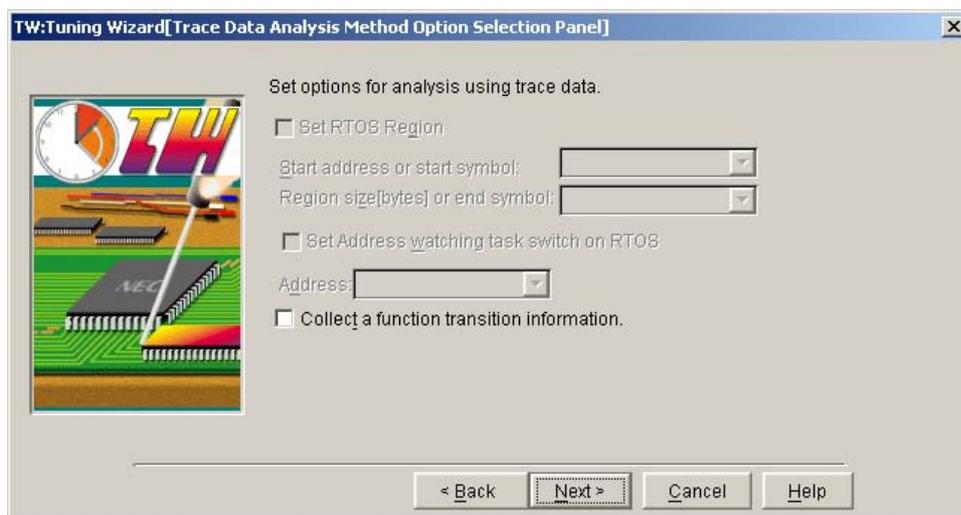
For the operating procedure if the [Use probe code for analyzing (for original execution module only).] radio button is checked in "5.3 Selecting Execution Item", refer to "5.5 When [Use probe code for analyzing (for original execution module only.) Is Selected".

5.4.1 Selecting trace analysis option

Specify an option using the trace data analysis method to acquire performance analysis information for wizard processing (performance analysis).

The following [Trace Data Analysis Method Option Selection Panel](#) is opened when each item has been specified in "5.3 Selecting Execution Item" and then <Next> button has been clicked.

Figure 5-9 Trace Data Analysis Method Option Selection Panel



To continue wizard processing (performance analysis), specify each item, and click the <Next> button.

Refer to [Trace Data Analysis Method Option Selection Panel](#) of WINDOW REFERENCE for details of each item.

5.5 When [Use probe code for analyzing (for original execution module only).] Is Selected

This section explains the procedure if the [Use probe code for analyzing (for original execution module only).] radio button is checked in "5.3 Selecting Execution Item".

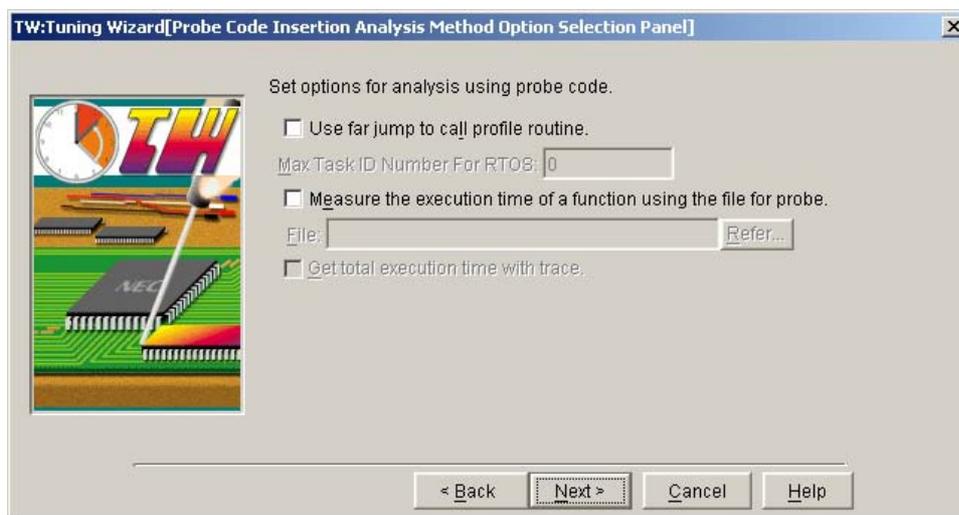
For the operating procedure if the [Use trace data for analyze.] radio button is checked in "5.3 Selecting Execution Item", refer to "5.4 When [Use trace data for analyze.] Is Selected".

5.5.1 Selecting probe analysis option

Specify an option using the probe code insertion method to acquire performance analysis information for wizard processing (performance analysis).

The following [Probe Code Insertion Analysis Method Option Selection Panel](#) is opened when each item has been specified in "5.3 Selecting Execution Item" and then <Next> button has been clicked.

Figure 5-10 Probe Code Insertion Analysis Method Option Selection Panel



To continue wizard processing (performance analysis), specify each item, and click the <Next> button.

Refer to [Probe Code Insertion Analysis Method Option Selection Panel](#) of WINDOW REFERENCE for details of each item.

5.6 When Only [Analyze an original execution module.] Is Selected

This section explains the procedure if only the [Analyze an original execution module.] check box on the "5.3 Selecting Execution Item" is checked.

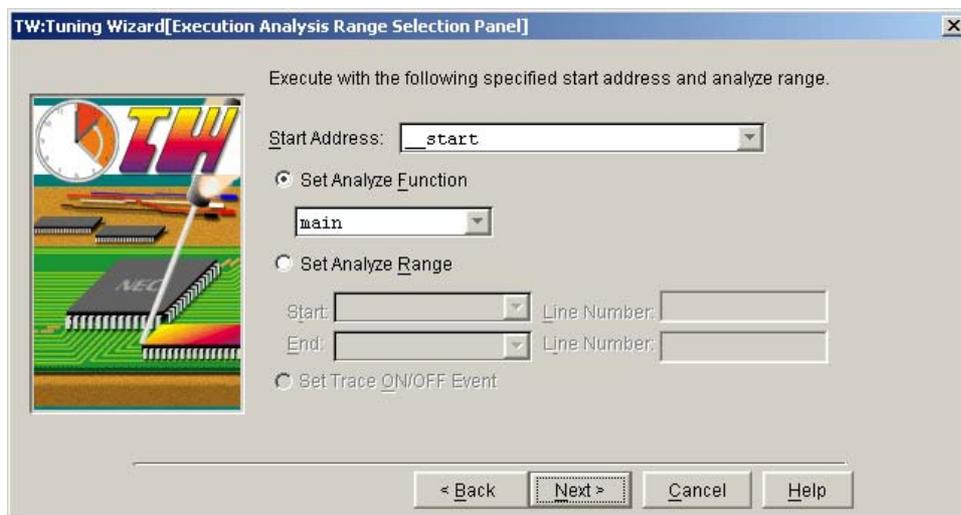
For the operating procedure if the [Tune and analyze a tuned execution module.] check box is checked in "5.3 Selecting Execution Item", refer to "5.7 When [Tune and analyze a tuned execution module.] Is Selected".

5.6.1 Selecting execution analysis range

Specify an execution start position of a user program or an execution module and range of performance analysis for wizard processing (performance analysis).

The following Execution Analysis Range Selection Panel is opened when each item has been specified in "5.3 Selecting Execution Item" and then <Next> button has been clicked.

Figure 5-11 Execution Analysis Range Selection Panel



To continue wizard processing (performance analysis), specify each item, and click the <Next> button.

Refer to Execution Analysis Range Selection Panel of WINDOW REFERENCE for details of each item.

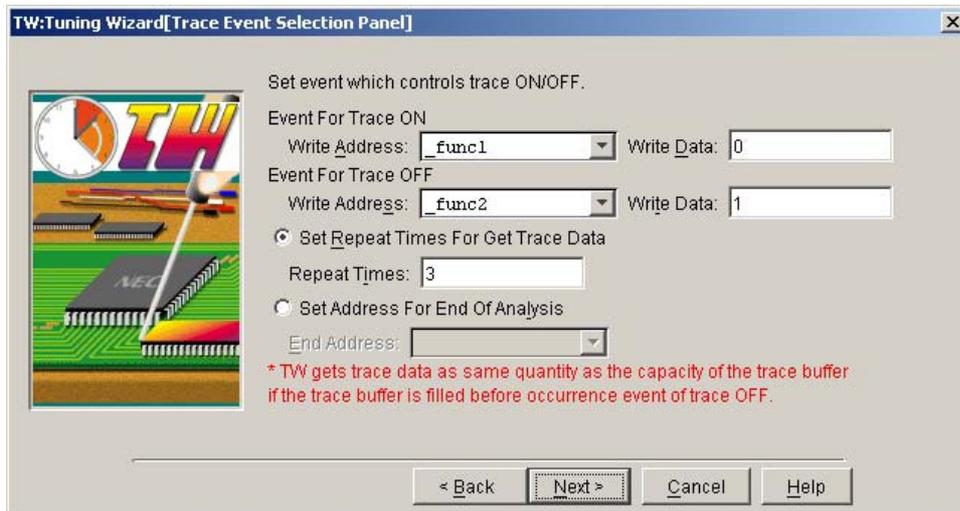
5.6.2 Selecting trace event

Specify range subject to performance analysis for wizard processing (performance analysis) by using a trace event. The following [Trace Event Selection Panel](#) is opened when the [Set Trace ON/OFF Event] radio button is selected in "5.6.1 [Selecting execution analysis range](#)" and then <Next>> button has been clicked.

Caution This panel is not opened if the following condition is satisfied.

- If the ID850NW is not used as the debugger

Figure 5-12 Trace Event Selection Panel



To continue wizard processing (performance analysis), specify each item, and click the <Next>> button.

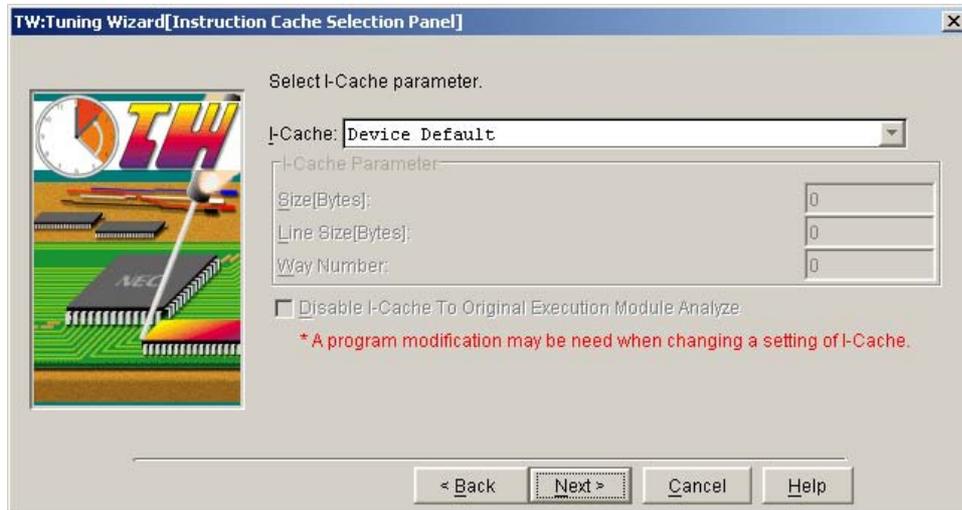
Refer to [Trace Event Selection Panel](#) of WINDOW REFERENCE for details of each item.

5.6.3 Selecting instruction cache

Specify a type of an instruction cache and parameters used for wizard processing (performance analysis).

The following [Instruction Cache Selection Panel](#) is opened when the [Set Trace ON/OFF Event] radio button is not checked in "5.6.1 [Selecting execution analysis range](#)" and each item has been specified and then <Next> button has been clicked, or when each item has been specified in "5.6.2 [Selecting trace event](#)" and then <Next> button has been clicked.

Figure 5-13 Instruction Cache Selection Panel



To continue wizard processing (performance analysis), specify each item, and click the <Next> button.

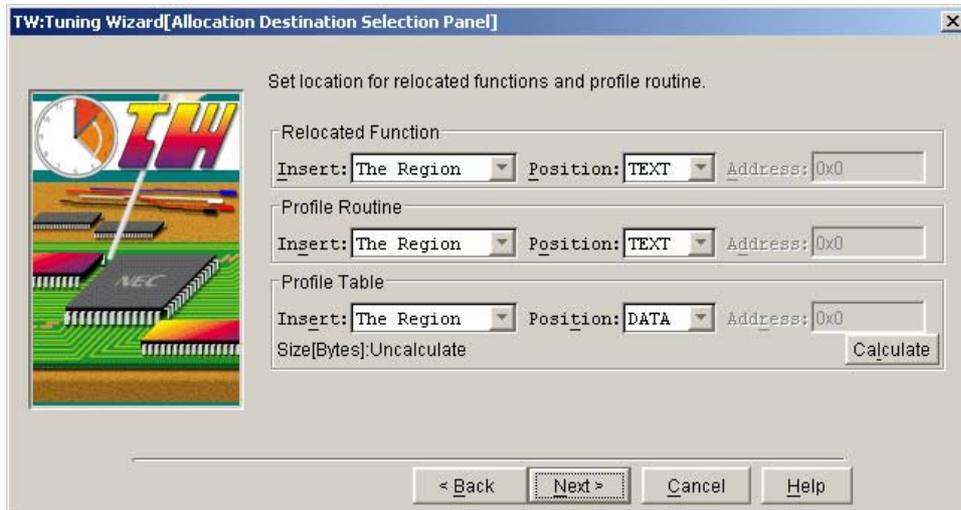
Refer to [Instruction Cache Selection Panel](#) of WINDOW REFERENCE for details of each item.

5.6.4 Selecting allocation destination

Specify a memory area for allocating a profile routine or a profile table output as an intermediate file during performance analysis.

The following [Allocation Destination Selection Panel](#) is opened when each item has been specified in "5.6.3 Selecting instruction cache" and then <Next> button has been clicked.

Figure 5-14 Allocation Destination Selection Panel



To continue wizard processing (performance analysis), specify each item, and click the <Next> button.

Refer to [Allocation Destination Selection Panel](#) of WINDOW REFERENCE for details of each item.

5.7 When [Tune and analyze a tuned execution module.] Is Selected

This section explains the procedure if the [Tune and analyze a tuned execution module.] check box on the "5.3 Selecting Execution Item" is checked.

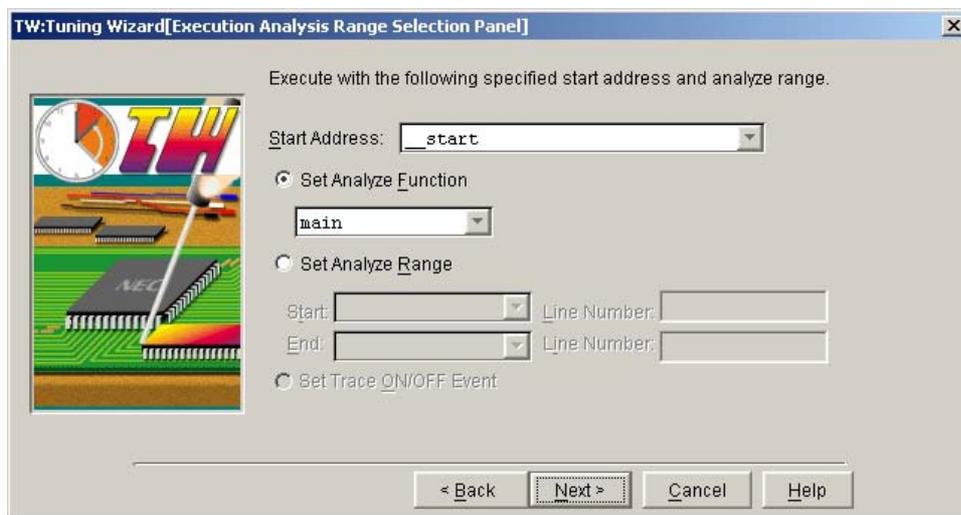
For the operating procedure if the [Analyze an original execution module.] check box is checked in "5.3 Selecting Execution Item", refer to "5.6 When Only [Analyze an original execution module.] Is Selected".

5.7.1 Selecting execution analysis range

Specify an execution start position of a user program or an execution module and range of performance analysis for wizard processing (performance analysis).

The following [Execution Analysis Range Selection Panel](#) is opened when each item has been specified in "5.3 Selecting Execution Item" and then <Next> button has been clicked.

Figure 5-15 Execution Analysis Range Selection Panel



To continue wizard processing (performance analysis), specify each item, and click the <Next> button.

Refer to [Execution Analysis Range Selection Panel](#) of WINDOW REFERENCE for details of each item.

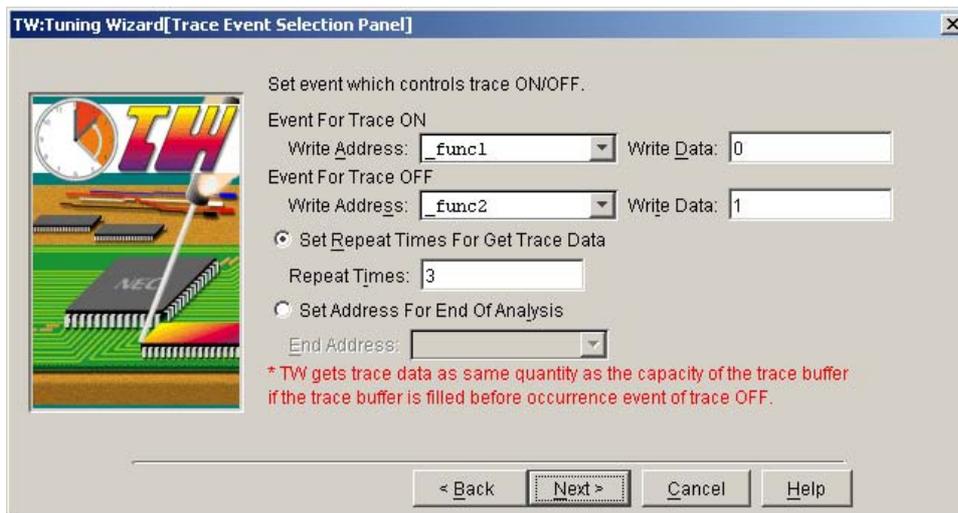
5.7.2 Selecting trace event

Specify range subject to performance analysis for wizard processing (performance analysis) by using a trace event. The following [Trace Event Selection Panel](#) is opened when the [Set Trace ON/OFF Event] radio button is selected in "5.7.1 [Selecting execution analysis range](#)" and then <Next>> button has been clicked.

Caution This panel is not opened if the following condition is satisfied.

- If the ID850NW is not used as the debugger

Figure 5-16 Trace Event Selection Panel



To continue wizard processing (performance analysis), specify each item, and click the <Next>> button.

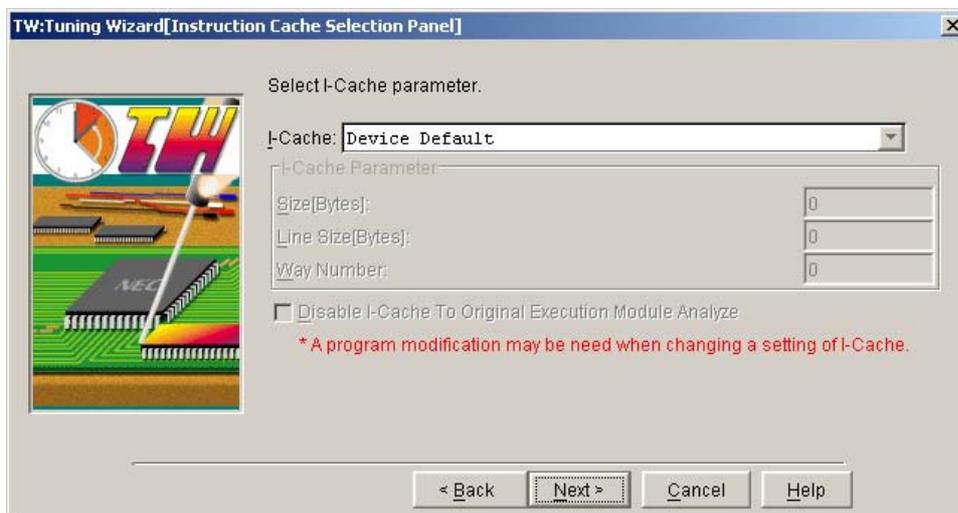
Refer to [Trace Event Selection Panel](#) of WINDOW REFERENCE for details of each item.

5.7.3 Selecting instruction cache

Specify a type and parameters of an instruction cache used for wizard processing (performance analysis or tuning).

The following [Instruction Cache Selection Panel](#) is opened when each item has been specified in "5.3 [Selecting Execution Item](#)" and then <Next> button has been clicked.

Figure 5-17 Instruction Cache Selection Panel



To continue wizard processing (performance analysis or tuning), specify each item, and click the <Next> button.

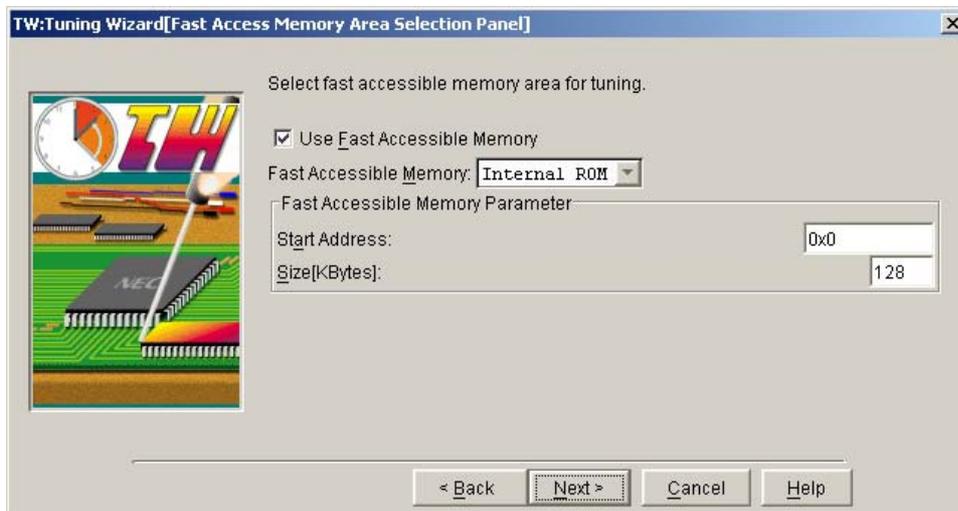
Refer to [Instruction Cache Selection Panel](#) of WINDOW REFERENCE for details of each item.

5.7.4 Selecting fast access memory area

Specify a type and parameters of a fast access memory area used for wizard processing (tuning).

The following [Fast Access Memory Area Selection Panel](#) is opened when each item has been specified in "5.7.3 Selecting instruction cache" and then <Next> button has been clicked.

Figure 5-18 Fast Access Memory Area Selection Panel



To continue wizard processing (performance analysis or tuning), specify each item, and click the <Next> button.

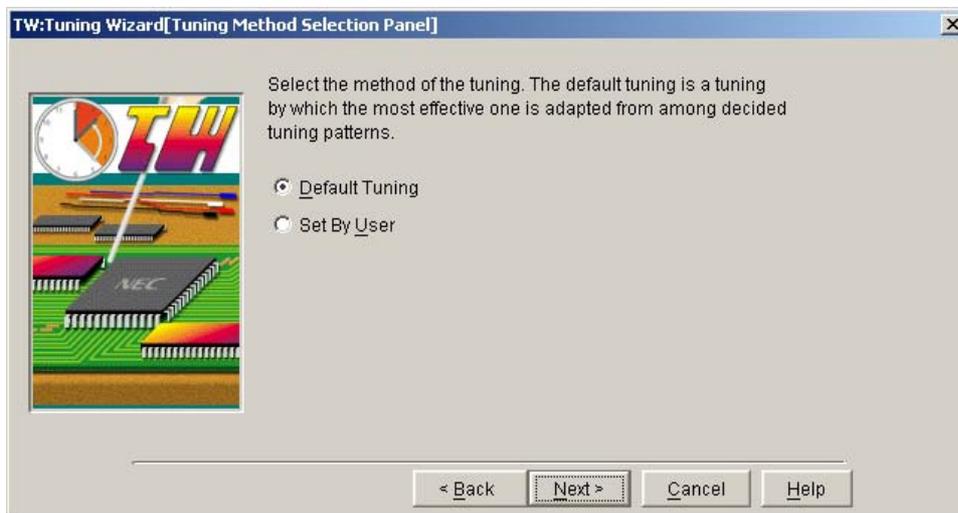
Refer to [Fast Access Memory Area Selection Panel](#) of WINDOW REFERENCE for details of each item.

5.7.5 Selecting tuning method

Specify a method of tuning a user program or an execution module (standard pattern or user-specified pattern). Refer to "4.2 Tuning Functions" for details of tuning functions.

The following [Tuning Method Selection Panel](#) is opened when each item has been specified in "5.7.4 Selecting fast access memory area" and then <Next> button has been clicked.

Figure 5-19 Tuning Method Selection Panel



To continue wizard processing (performance analysis or tuning), specify each item, and click the <Next> button.

Refer to [Tuning Method Selection Panel](#) of WINDOW REFERENCE for details of each item.

5.8 When [Default Tuning] Is Selected

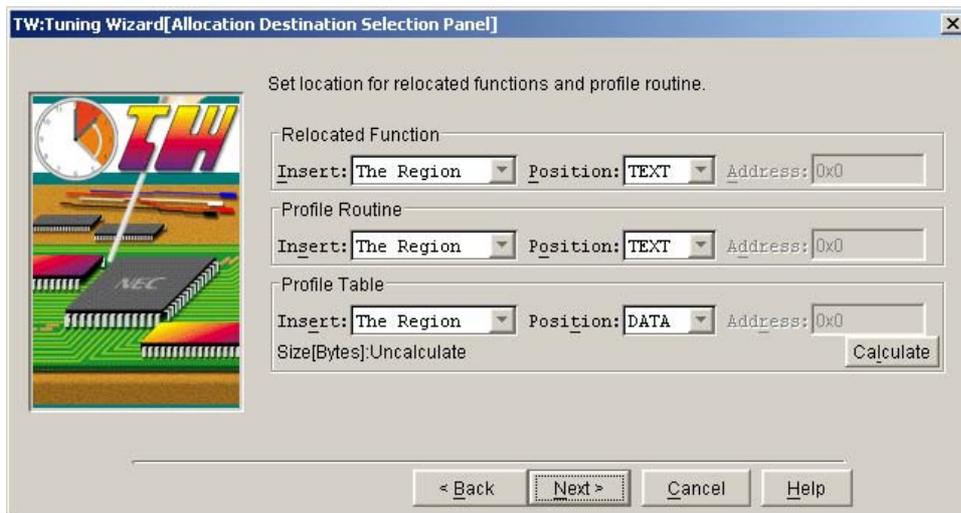
This section explains the procedure if the [Default Tuning] radio button is checked in "5.7.5 Selecting tuning method". For the operating procedure if the [Set By User] radio button is checked in "5.7.5 Selecting tuning method", refer to "5.9 When [Set By User] Is Selected".

5.8.1 Selecting allocation destination

Specify a memory area in which to relocate a function that is subject to relocation during tuning, and a memory area where a profile routine or a profile table output as intermediate files during tuning are to be allocated.

The following Allocation Destination Selection Panel is opened when [Default Tuning] check button is checked in "5.7.5 Selecting tuning method" and then <Next>> button has been clicked.

Figure 5-20 Allocation Destination Selection Panel



To continue wizard processing (performance analysis or tuning), specify each item, and click the <Next>> button.

Refer to Allocation Destination Selection Panel of WINDOW REFERENCE for details of each item.

5.9 When [Set By User] Is Selected

This section explains the procedure if the [Set By User] radio button is checked in "5.7.5 Selecting tuning method".

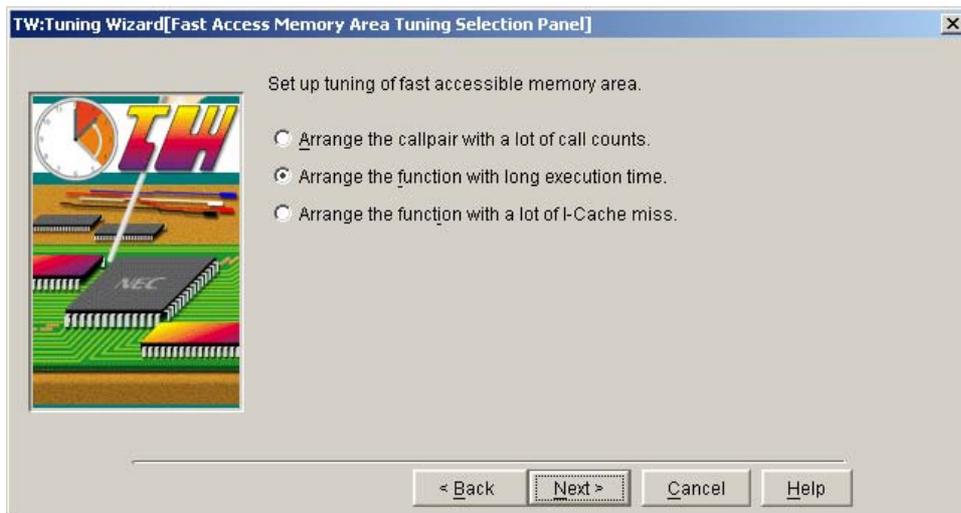
For the operating procedure if the [Default Tuning] radio button is checked in "5.7.5 Selecting tuning method", refer to "5.8 When [Default Tuning] Is Selected".

5.9.1 Selecting tuning method for fast access memory area

Specify a method of allocating functions in the fast access memory area (such as internal ROM), based on callpair information that was acquired when the performance of a user program or an execution module was analyzed.

The following [Fast Access Memory Area Tuning Selection Panel](#) is opened when [Set By User] check button is checked in "5.7.5 Selecting tuning method" and then <Next> button has been clicked.

Figure 5-21 Fast Access Memory Area Tuning Selection Panel



To continue wizard processing (performance analysis or tuning), specify each item, and click the <Next> button.

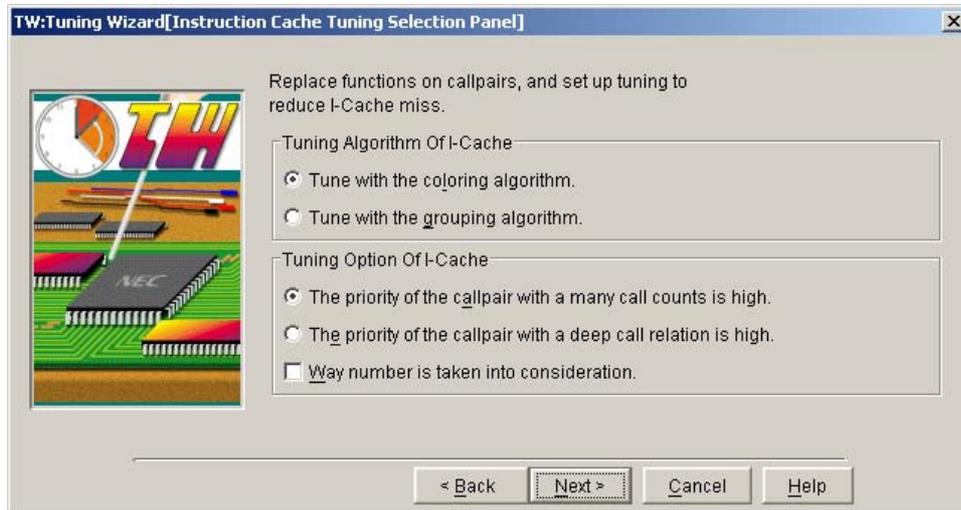
Refer to [Fast Access Memory Area Tuning Selection Panel](#) of WINDOW REFERENCE for details of each item.

5.9.2 Selecting tuning method of instruction cache

Specify a method of tuning an instruction cache (tuning algorithm and allocation priority correction option), based on callpair information that was acquired when the performance of a user program or execution module was analyzed.

The following [Instruction Cache Tuning Selection Panel](#) is opened when each item has been specified in "[5.9.1 Selecting tuning method for fast access memory area](#)" and then <Next> button has been clicked.

Figure 5-22 Instruction Cache Tuning Selection Panel



To continue wizard processing (performance analysis or tuning), specify each item, and click the <Next> button.

Refer to [Instruction Cache Tuning Selection Panel](#) of WINDOW REFERENCE for details of each item.

5.9.3 Selecting callpair correction option

Specify a method of tuning functions subject to be relocated (the callpair priority correction method in tuning of a fast access memory and an instruction cache), based on callpair information that was acquired when a performance of a user program or an execution module was analyzed.

The following [Callpair Correction Option Selection Panel](#) is opened when each item has been specified in "5.9.2 Selecting tuning method of instruction cache" and then <Next> button has been clicked.

Figure 5-23 Callpair Correction Option Selection Panel



To continue wizard processing (performance analysis or tuning), specify each item, and click the <Next> button.

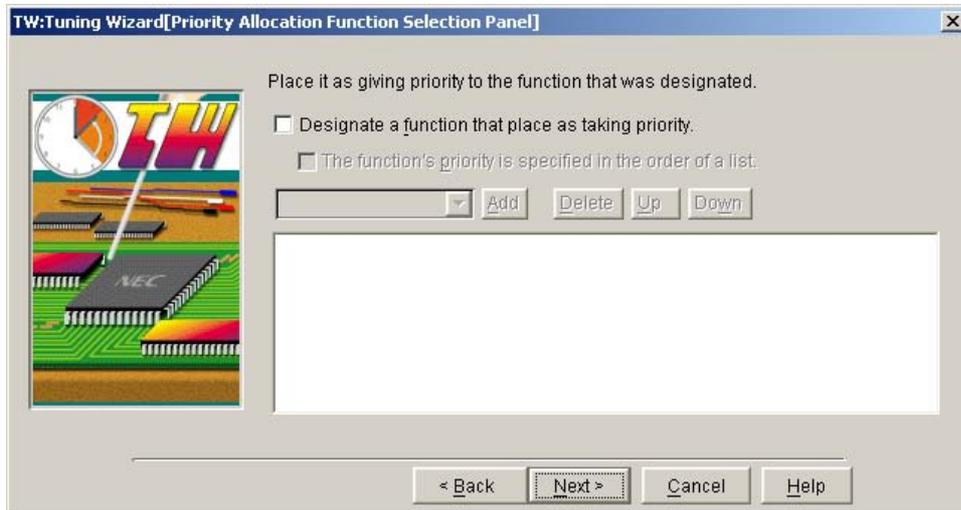
Refer to [Callpair Correction Option Selection Panel](#) of WINDOW REFERENCE for details of each item.

5.9.4 Selecting priority allocation function

Specify a function that is given priority and relocated to correct a callpair priority in tuning of a fast access memory and an instruction cache during wizard processing (tuning).

The following [Priority Allocation Function Selection Panel](#) is opened when each item has been specified in "5.9.3 [Selecting callpair correction option](#)" and then <Next> button has been clicked.

Figure 5-24 Priority Allocation Function Selection Panel



To continue wizard processing (performance analysis or tuning), specify each item, and click the <Next> button.

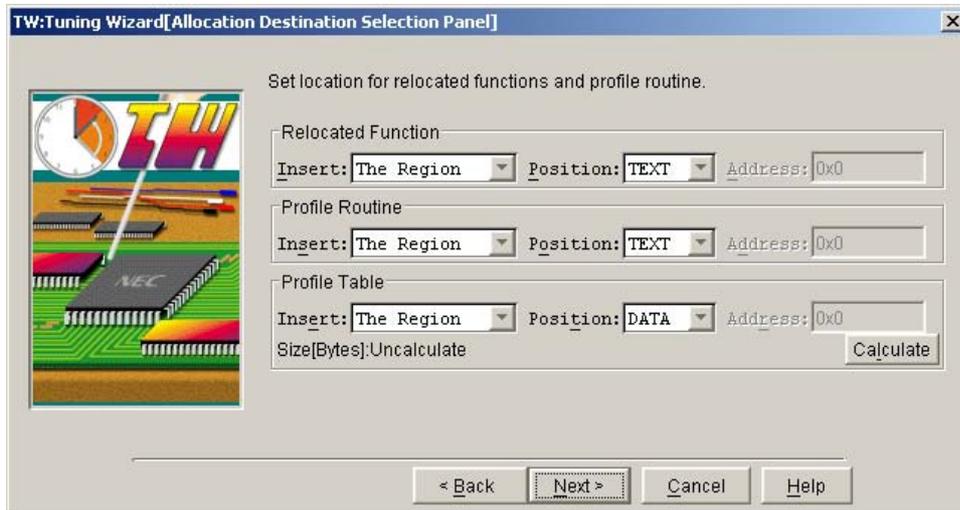
Refer to [Priority Allocation Function Selection Panel](#) of WINDOW REFERENCE for details of each item.

5.9.5 Selecting allocation destination

Specify a memory area in which to relocate a function that is subject to relocation during tuning, and a memory area where a profile routine or a profile table output as intermediate files during tuning are to be allocated.

The following [Allocation Destination Selection Panel](#) is opened when each item has been specified in "5.9.4 Selecting priority allocation function" and then <Next> button has been clicked.

Figure 5-25 Allocation Destination Selection Panel



To continue wizard processing (performance analysis or tuning), specify each item, and click the <Next> button.

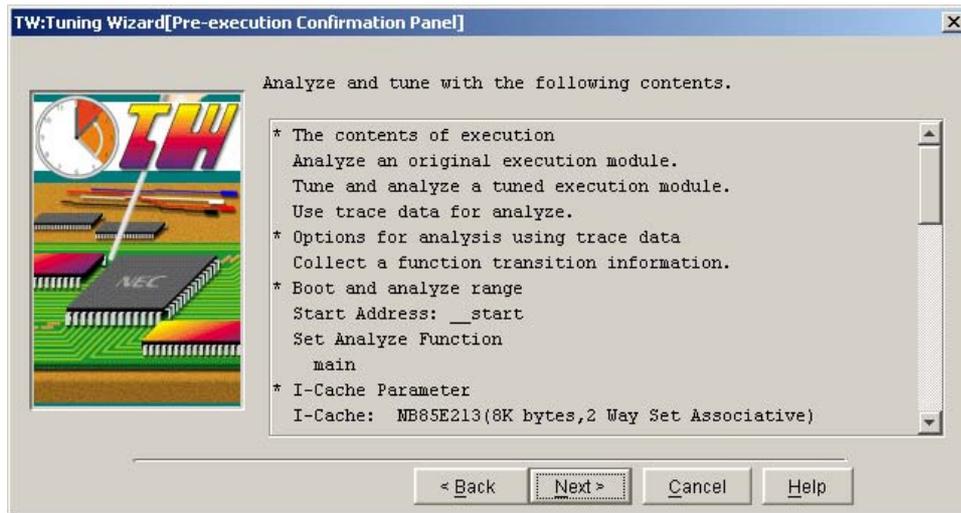
Refer to [Allocation Destination Selection Panel](#) of WINDOW REFERENCE for details of each item.

5.10 Executing Performance Analysis/Tuning

5.10.1 Checking before execution

The following [Pre-execution Confirmation Panel](#) is opened when setting of performance analysis or tuning items required for the TW850 to execute wizard processing (performance analysis or tuning) (i.e., setting operations up to "[5.6.4 Selecting allocation destination](#)", "[5.8.1 Selecting allocation destination](#)" or "[5.9.5 Selecting allocation destination](#)") has been correctly completed.

Figure 5-26 Pre-execution Confirmation Panel



To execute wizard processing (performance analysis or tuning), check a message report that setting of the performance analysis or tuning items required for the TW850 to execute wizard processing (performance analysis or tuning) has been completed is displayed in the message display area, and click the **<Next>** button.

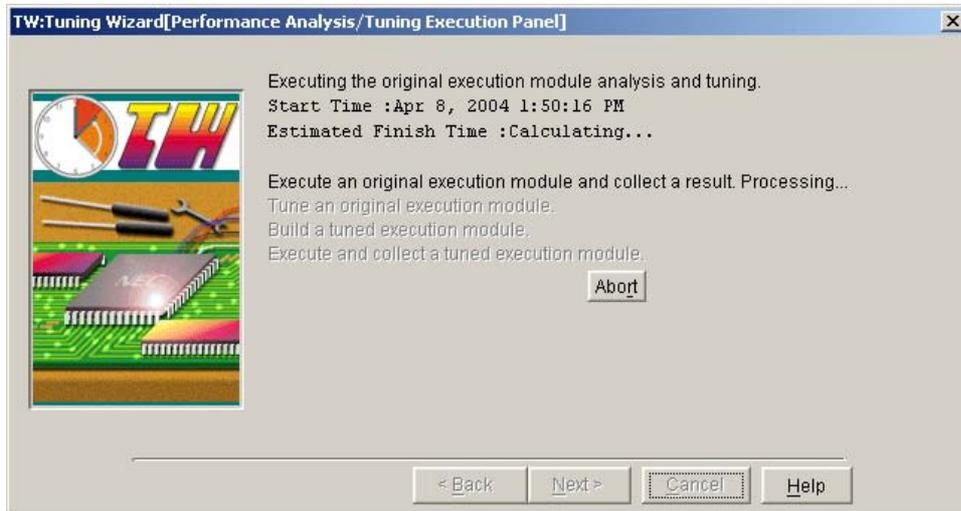
5.10.2 Executing performance analysis/tuning

Wizard processing (performance analysis or tuning) is executed in accordance with the performance analysis or tuning items displayed in "5.10.1 Checking before execution".

However, the user does not have to be aware of this phase because it is automatically performed by the TW850 when the message displayed in "5.10.1 Checking before execution" and the <Next> button is clicked.

The Performance Analysis/Tuning Execution Panel that is opened while wizard processing (performance analysis or tuning) of the TW850 is in progress is shown below.

Figure 5-27 Performance Analysis/Tuning Execution Panel



5.10.3 Checking completion of execution

Check that the TW850 has completed wizard processing (performance analysis or tuning).

The following [Execution Completion Confirmation Panel](#) is automatically opened when the TW850 has correctly completed wizard processing (performance analysis or tuning) in "[5.10.2 Executing performance analysis/tuning](#)".

Figure 5-28 Execution Completion Confirmation Panel



To complete wizard processing (performance analysis or tuning), check a message to report that the TW850 has completed wizard processing (performance analysis or tuning) is displayed in the message display area, and click the <Finish> button.

By clicking the <Finish> button, the wizard of the TW850 that was started in "[5.2.1 Starting wizard](#)" is exited.

Remark If the [Tune and analyze a tuned execution module.] check box is checked in "[5.3 Selecting Execution Item](#)", a tuned execution module is output to the "(original project file name).tw" folder.

5.11 Checking Result

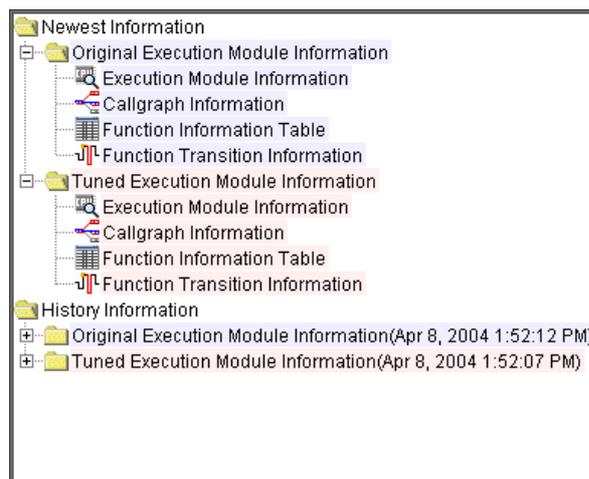
The results of performance analysis or tuning of a user program or an execution module can be checked on the [Main Window](#).

5.11.1 History information

The results of performance analysis or tuning of an original execution module and a tuned execution module can be checked in the history display area of the [Main Window](#).

Whether the following history display area is to be displayed or not can be specified by [View] menu -> [History] on the [Main Window](#).

Figure 5-29 History Display Area



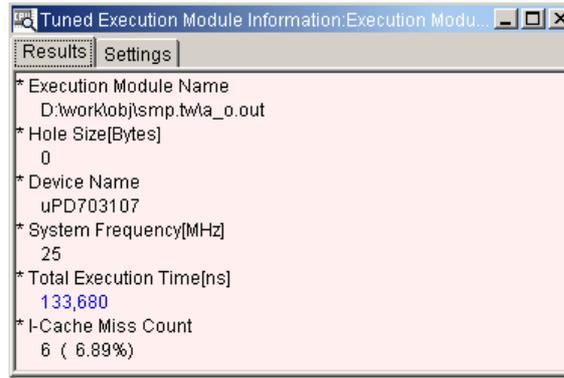
Refer to [Main Window](#) of WINDOW REFERENCE for details of this display area.

5.11.2 Execution module information

The execution module information of an execution module (the execution results and setting contents) is displayed.

The following [Execution Module Information Window](#) is opened when [Execution Module Information] is selected in the history display area of the [Main Window](#) and then the right mouse button menu -> [Open] has been selected.

Figure 5-30 Execution Module Information Window



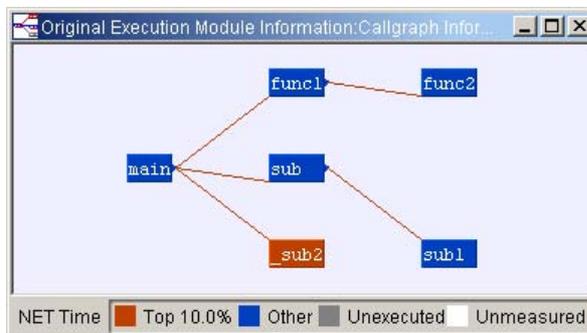
Refer to [Execution Module Information Window](#) of WINDOW REFERENCE for details of this window.

5.11.3 Callgraph information

Performance analysis information on an execution module (such as callpair information and call count information) is displayed in graph form.

The following [Callgraph Window](#) is opened when [Callgraph Information] is selected in the history display area of the [Main Window](#) and then the right mouse button menu -> [Open] has been selected.

Figure 5-31 Callgraph Window



Refer to [Callgraph Window](#) of WINDOW REFERENCE for details of this window.

5.11.4 Function information table

A list of function information of an execution module (such as the function name, number of times of execution, and NET time) is displayed in table form.

The following [Function Information Table Window](#) is opened when [Function Information Table] is selected in the history display area of the [Main Window](#) and then the right mouse button menu -> [Qopen] has been selected.

Figure 5-32 Function Information Table Window

Function Name	Execution Count	NET Time[ns]	Average NET Time[ns]	GROSS Time[ns]	Average GROSS Time[ns]	Unknown Function Time[ns]	I-Cache Miss Count
func2	5	10,200 (10.71%)	2,040 (2.14%)	10,200 (10.71%)	2,040 (2.14%)	0 (0.00%)	0
_sub2	1	48,520 (50.98%)	48,520 (50.98%)	53,600 (56.32%)	53,600 (56.32%)	5,080 (5.33%)	0
func1	1	2,320 (2.43%)	2,320 (2.43%)	12,520 (13.15%)	12,520 (13.15%)	0 (0.00%)	0
main	1	1,240 (1.30%)	1,240 (1.30%)	95,160 (100.00%)	95,160 (100.00%)	5,080 (5.33%)	0
sub	1	24,800 (26.06%)	24,800 (26.06%)	27,800 (29.21%)	27,800 (29.21%)	0 (0.00%)	0
sub1	1	3,000 (3.15%)	3,000 (3.15%)	3,000 (3.15%)	3,000 (3.15%)	0 (0.00%)	0
Unknown Function		5,080 (5.33%)					0

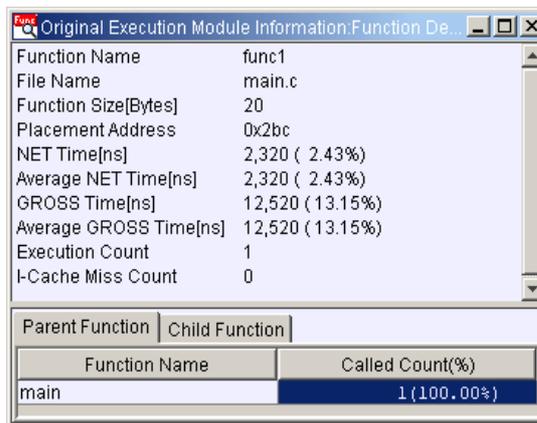
Refer to [Function Information Table Window](#) of WINDOW REFERENCE for details of this window.

5.11.5 Detailed function information

Detailed function information of an execution module (performance analysis information, parent function information, and child function information) is displayed.

The following [Function Detail Information Window](#) is opened when the Function Detail button on the [Main Window](#) is clicked after selecting a function name displayed in the callgraph display area of the [Callgraph Window](#), when the Function Detail button on the [Main Window](#) is clicked after selecting a function name displayed in the function name display area of the [Function Information Table Window](#), or when the Function Detail button on the [Main Window](#) is clicked after selecting a function name displayed on the [Parent Function] tab or [Child Function] tab in the [Function Detail Information Window](#).

Figure 5-33 Function Detail Information Window



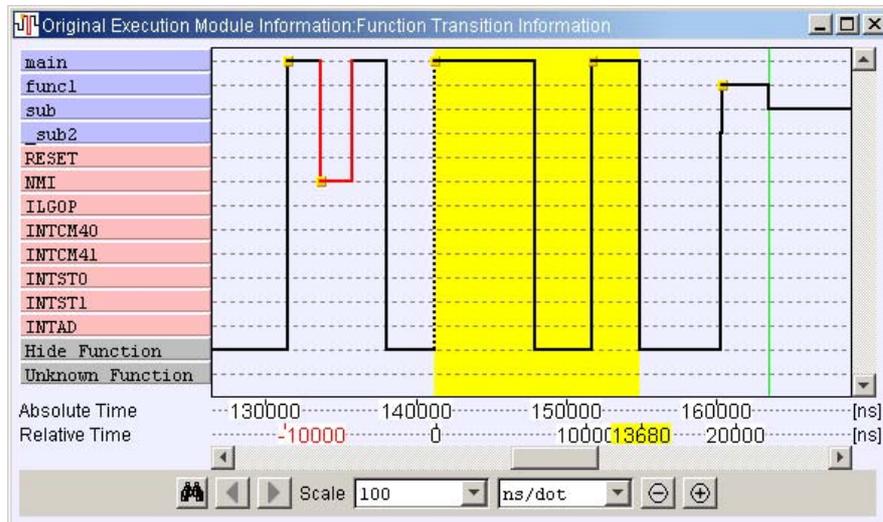
Refer to [Function Information Table Window](#) of WINDOW REFERENCE for details of this window.

5.11.6 Function transition information

Function transition information on an execution module is displayed in graph form.

The following [Function Transition Information Window](#) is opened when [Function Transition Information] is selected in the history display area of the [Main Window](#) and then the right mouse button menu -> [Open] has been selected.

Figure 5-34 Function Transition Information Window



Refer to [Function Transition Information Window](#) of WINDOW REFERENCE for details of this window.

5.12 Exiting TW850

Refer to "3.2 [Exiting Method](#)" for details of exiting the TW850.

If an exiting operation has been executed for the TW850, the following [Verification Dialog Box](#) is automatically opened.

Figure 5-35 Verification Dialog Box



To continue the exit operation for the TW850, check the "inquiry information" displayed in the message display area, and click the <Yes> button.

CHAPTER 6 WINDOW REFERENCE

This chapter explains details on the functions of the windows, panels and dialog boxes of the TW850.

Table 6-1 shows the list of the windows, panels, and dialog boxes of the TW850.

Table 6-1 Windows, Panels, and Dialog Boxes of TW850

Window name, Panel name, or Dialog Box name	Functional outline
Main Window	This window starts wizard processing (performance analysis or tuning) and displays performance analysis information (such as callpair information and call count information).
Callgraph Window	This window is used to display performance analysis information (such as callpair information and call count information) on an execution module in graph form.
Function Information Table Window	This window is used to display a list of function information (such as the function name, number of times of execution, and NET time) of an execution module in table form.
Function Transition Information Window	This window is used to display function transition information on an execution module in graph form.
Execution Module Information Window	This window is used to display execution module information (execution results and setting contents) of an execution module.
Function Detail Information Window	This window is used to display detailed function information (performance analysis information, parent function information, and child function information) of an execution module.
Tuning wizard Initial Screen Panel	This panel is used to report that the TW850 has started wizard processing (performance analysis or tuning) and specify the RTOS used for the program that executes performance analysis.
Tuning wizard Project Information Acquisition Panel	This panel is used to report that the TW850 is executing wizard processing (project information acquiring).
Tuning wizard Analysis Source File Selection Panel	This panel is used to specify source files subject to performance analysis or tuning during wizard processing (performance analysis or tuning) of the TW850.
Tuning wizard Execution Module Information Acquisition Panel	This panel is used to report that the TW850 is executing wizard processing (generation processing of an execution module for original analysis, and analysis processing of functions and an original execution module).
Tuning wizard Execution Item Selection Panel	This panel is used to specify execution items for wizard processing (performance analysis or tuning) of the TW850.
Tuning wizard Trace Data Analysis Method Option Selection Panel	This panel is used to specify an option using the trace data analysis method to acquire performance analysis information for wizard processing (performance analysis) of the TW850.
Tuning wizard Probe Code Insertion Analysis Method Option Selection Panel	This panel is used to specify an option using the probe code insertion method to acquire performance analysis information for wizard processing (performance analysis) of the TW850.

Window name, Panel name, or Dialog Box name	Functional outline
Tuning wizard Execution Analysis Range Selection Panel	This panel is used to specify an execution start position of a user program or an execution module and range of performance analysis for wizard processing (performance analysis).
Tuning wizard Trace Event Selection Panel	This panel is used to select a trace event in range of performance analysis for wizard processing (performance analysis) of the TW850.
Tuning wizard Instruction Cache Selection Panel	This panel is used to specify a type and parameters of an instruction cache used for wizard processing (performance analysis or tuning) of the TW850.
Tuning wizard Fast Access Memory Area Selection Panel	This panel is used to specify a type and parameters of a fast access memory area used for wizard processing (tuning) of the TW850.
Tuning wizard Tuning Method Selection Panel	This panel is used to specify a method of tuning a user program or an execution module (standard pattern or user-specified pattern).
Tuning wizard Fast Access Memory Area Tuning Selection Panel	This panel is used to specify a method of allocating functions in a fast access memory area (such as internal ROM), based on callpair information that was acquired when a performance of a user program or execution module was analyzed.
Tuning wizard Instruction Cache Tuning Selection Panel	This panel is used to specify a method of tuning an instruction cache (tuning algorithm and allocation priority correction option), based on callpair information that was acquired when a performance of a user program or an execution module was analyzed.
Tuning wizard Callpair Correction Option Selection Panel	This panel is used to specify a method of tuning functions subject to be relocated (the callpair priority correction method in tuning of a fast access memory and an instruction cache), based on callpair information that was acquired when a performance of a user program or an execution module was analyzed.
Tuning wizard Priority Allocation Function Selection Panel	This panel is used to specify a function that is given priority and relocated to correct a callpair priority in tuning of a fast access memory and an instruction cache during wizard processing (tuning) of the TW850.
Tuning wizard Allocation Destination Selection Panel	This panel is used to specify a memory area in which to relocate a function that is subject to relocation during tuning, and a memory area where a profile routine or a profile table output as intermediate files during tuning are to be allocated.
Tuning wizard Pre-execution Confirmation Panel	This panel is used to report that settings of performance analysis or tuning items required for the TW850 to execute wizard processing (performance analysis or tuning) has been completed.
Tuning wizard Performance Analysis/Tuning Execution Panel	This panel is used to report that the TW850 is executing wizard processing (performance analysis or tuning).
Tuning wizard Execution Completion Confirmation Panel	This panel is used to report that the TW850 has completed wizard processing (performance analysis or tuning).
Select Display Function Dialog Box	This dialog box is used to select functions or interrupts displayed on the Function Transition Information Window .
Search Function Transition Information Dialog Box	This dialog box is used to search a function execution or a function transition line on the Function Transition Information Window .

Window name, Panel name, or Dialog Box name	Functional outline
Option Dialog Box	This dialog box is used to specify how to print the Callgraph Window displayed in the internal window display area of the Main Window , or whether intermediate files (temporary files) the TW850 output while it was executing performance analysis or tuning processing are to be deleted.
Version Information Dialog Box	This dialog box is used to display version information (such as the version and copyright) of the TW850.
File Selection Dialog Box	This dialog box is used to specify files required by the TW850 to execute performance analysis or tuning processing (such as timer control code block).
Message Dialog Box	This dialog box is used to display information to be reported to a user (such as the progress of processing and occurrence of a warning) detected by the TW850 while it was executing performance analysis or tuning processing.
Error Dialog Box	This dialog box is used to display information to be reported to a user (such as the occurrence of a fatal error) detected by the TW850 when it was started or while it was executing performance analysis or tuning processing.
Verification Dialog Box	This dialog box is used to display inquiry information (such as the problem that prevents continuation of processing) detected by the TW850 while it was executing performance analysis or tuning processing.
Input Dialog Box	This dialog box is used to specify information (such as the color boundary, display magnification rate, and root function) requested by the TW850.
TW850 Options Dialog Box	This dialog box is used to specify start options of the TW850.

6.1 Input Rules

This section explains the rules for inputting characters (alphanumeric characters) in the windows, panels, or dialog boxes of the TW850.

- File name
 - File names are composed of a to z, A to Z, 0 to 9, ., _, +, and -.
 - File names cannot be prefixed by a period (.), +, or -.
 - File names cannot be suffixed by a period (.).
 - Uppercase characters and lowercase characters are not distinguished for file names.
 - The TW850 generates intermediate files with the original file name plus up to five characters while it is analyzing the performance of or tuning a user program or an execution module.
Therefore, the number of characters in the original file name that a user can specify is five fewer than the maximum number (259 characters) defined by a C compiler package.
- Path name
 - File names are composed of a to z, A to Z, 0 to 9, ., _, +, -, and one-byte spaces.
 - Path names cannot be prefixed by a period (.), +, -, or one-byte space.
 - Path names cannot be suffixed by a period (.), or one-byte space.
 - Uppercase characters and lowercase characters are not distinguished for file names.
 - The TW850 generates intermediate files in a folder that stores an original file, or a folder prescribed by the TW850 while it is analyzing the performance of or tuning a user program or an execution module.
Therefore, the number of characters in the path name that a user can specify is the maximum number (259 characters) defined by a C compiler package.
- Function name and symbol name
 - The TW850 outputs performance analysis information with the original function name or original symbol name plus up to 12 characters while it is analyzing the performance analysis of or tuning of a user program or an execution module.
Therefore, the number of characters in the original function name or original symbol name that a user can specify is 12 fewer than the maximum number (259 characters) defined by a C compiler package.
 - With the TW850, function names and symbol names starting with "TW_" and "tw_" are handled as function names and symbol names reserved by the TW850 and cannot be used.
- Section name
 - With the TW850, section names starting with "TW_" and "tw_" are handled as section names reserved by the TW850 and cannot be used.
- Macro name
 - With the TW850, words starting with "tw_" and "..", and "__e_sysfnc" are handled as system-reserved macros and cannot be used.
- Numeric value
 - The TW850 treats a word starting with a number (0 to 9) as a numeric value.
The TW850 usually distinguishes numeric values as follows.

XXXX:	Decimal
0xXXXX:	Hexadecimal

6.2 Window Explanation

This section explains the windows, panels, and dialog boxes of the TW850 in the following description format.

Window name, Panel name, or Dialog Box name
--

Explains the functional outline of the window, panel, or dialog box, and then shows the screen image.

Opening

Explains how to open the window, panel, or dialog box.

Explanation of each area

Enumerates the constituent elements of the window, panel, or dialog box, and then explains the detailed information of each element.

The function buttons of panels consist of the following buttons.

(1) <<Back> button

Returns to the previous wizard step.

This button is invalid on the [Initial Screen Panel](#) and [Execution Completion Confirmation Panel](#).

(2) <Next>> button

Opens the next step panel. (The next step panel to be opened differs depending on the selection.)

This button is invalid on the [Project Information Acquisition Panel](#), [Execution Module Information Acquisition Panel](#) and [Performance Analysis/Tuning Execution Panel](#).

On the [Execution Completion Confirmation Panel](#), the <Finish> button is displayed instead and used for closing the panel.

(3) <Cancel> button

Closes the panel without executing wizard processing.

This button is invalid on the [Execution Completion Confirmation Panel](#).

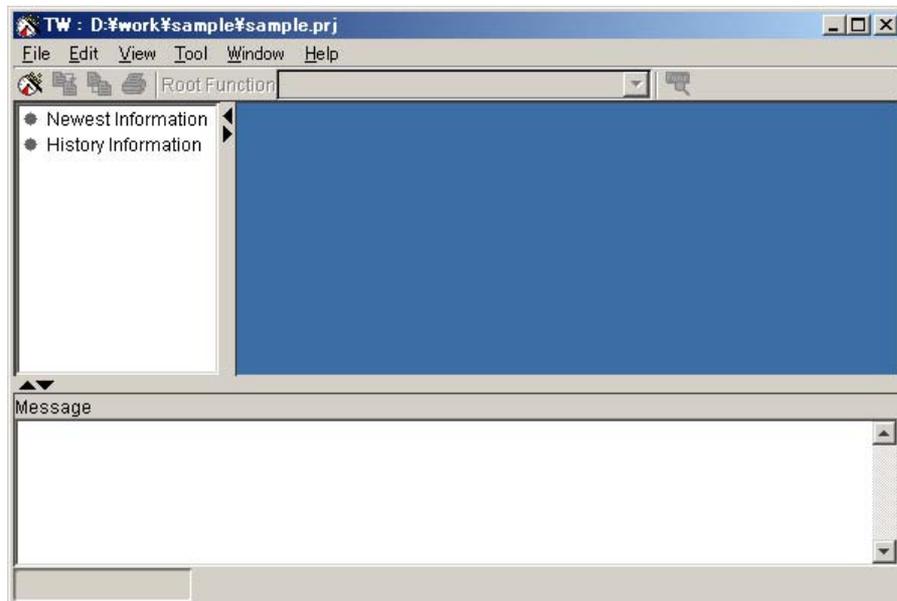
(4) <Help> button

Starts the help.

Main Window

This window starts wizard processing (performance analysis or tuning) and displays performance analysis information (such as callpair information and call count information).

Figure 6-1 Main Window



This section describes the following items.

- [Opening](#)
- [Explanation of each area](#)
- [Context menu](#)

Opening

This window is opened automatically when the TW850 is started.

Explanation of each area

This window consists of the following items.

- [Menu bar](#)
- [Tool bar](#)
- [History display area](#)
- [Internal window display area](#)
- [Message display area](#)
- [Status bar](#)

(1) Menu bar

The menu bar consists of the following menus.

(a) [File] menu

This menu consists of the following items.

- [Save TW Project]
Saves the execution status of the TW850.
- [Import...]
By selecting this item, a dialog box ([File Selection Dialog Box](#)) to specify the file used to read the function transition information data saved by using the [Export...] item is opened.
- [Export...]
By selecting this item after selecting an internal window (such as the [Function Information Table Window](#), [Function Transition Information Window](#), [Execution Module Information Window](#), or [Function Detail Information Window](#)), a dialog box ([File Selection Dialog Box](#)) to specify the file to save the selected window in text format is opened.
- [Print...]
By selecting this item after selecting an internal window (such as the [Callgraph Window](#), [Function Information Table Window](#), or [Function Transition Information Window](#)), the selected window is printed.
- [Exit...]
Opens the [Verification Dialog Box](#) to close this window.

(b) [Edit] menu

This menu consists of the following items.

- [Cut]
Cuts and pastes the text selected in the text input area to the copy buffer.

Remark This item is invalid if the following condition is satisfied.
 - If the TW850 is executing wizard processing
- [Copy]
Copies the text selected in the text input area to the copy buffer.

Remark This item is invalid if the following condition is satisfied.
 - If the TW850 is executing wizard processing
- [Paste]
Pastes the text stored (text cut or copied) in the copy buffer using the [Cut] or [Copy] item to the specified text input area.

Remark This item is invalid if the following condition is satisfied.
 - If the TW850 is executing wizard processing
- [Delete]
Deletes the text selected in the text input area.
If the focus is on the history display area, the selected history information item is deleted.

Remark This item is invalid if the following condition is satisfied.
 - If the TW850 is executing wizard processing
- [Search...]
Opens a dialog box ([Search Function Transition Information Dialog Box](#)) to search the transition of a function.

Remark This item is invalid if the following condition is satisfied.
 - If the [Function Transition Information Window](#) is not selected

- [Search Previous]

In the function transition information display area of the [Function Transition Information Window](#), searches the function execution line or function transition line of the condition (function or interrupt) specified in the [Function Transition Information Window](#) in the left direction, starting from the selected function execution line or function transition line.

Remark This item is invalid if either of the following conditions is satisfied.

 - If the [Function Transition Information Window](#) is not selected
 - If the specified search condition is invalid in the [Function Transition Information Window](#)
 - [Next]

In the function transition information display area of the [Function Transition Information Window](#), searches the function execution line or function transition line of the condition (function or interrupt) specified in the [Function Transition Information Window](#) in the right direction, starting from the selected function execution line or function transition line.

Remark This item is invalid if either of the following conditions is satisfied.

 - If the [Function Transition Information Window](#) is not selected
 - If the specified search condition is invalid in the [Function Transition Information Window](#)
 - [Add To History]

By selecting this item after selecting an item (such as [Original Execution Module Information] or [Tuned Execution Module Information]) displayed on [Newest Information] in the history display area, the selected item can be added and registered to [History Information].
 - [Restore From History...]

By selecting this item after selecting an item (such as [Original Execution Module Information] or [Tuned Execution Module Information]) displayed on [History Information] in the history display area, the setting of the selected item is reflected on the wizard.
- (c) [View] menu
- This menu consists of the following items.
- [Toolbar]

Specifies whether the toolbar (such as the Tuning Wizard button, Add To History button, and Restore From History button) is to be displayed.
If this item is checked, the toolbar is displayed.
 - [History]

Specifies whether the history display area (that displays the result history of wizard processing in tree form) is to be displayed.
If this item is checked, the history display area is displayed.
 - [Message]

Specifies whether the message display area (that displays the information to be reported to the user detected by the TW850) is to be displayed.
If this item is checked, the message display area is displayed.
 - [Open]

By selecting this item after selecting an item displayed in the history display area (such as the callgraph information, function information table, function transition information, or execution module information), the window corresponding to the selected item (such as the [Callgraph Window](#), [Function Information Table Window](#), [Function Transition Information Window](#), or [Execution Module Information Window](#)) is opened.
 - [Function Detail]

By selecting this item after selecting a function name displayed in the [Callgraph Window](#), [Function Information Table Window](#), [Function Transition Information Window](#), or [Function Detail Information Window](#), a window ([Function Detail Information Window](#)) that displays the detailed information of the execution module (performance analysis information, parent function information, child function information) is opened.

Remark This item is invalid if either of the following conditions is satisfied.

- If none of the [Callgraph Window](#), [Function Information Table Window](#), [Function Transition Information Window](#), and [Function Detail Information Window](#) is selected
- If any of the [Callgraph Window](#), [Function Information Table Window](#), [Function Transition Information Window](#), and [Function Detail Information Window](#) is selected and if a function is not selected on the selected window

- [Callgraph]

This item consists of the following items.

- [NET Time]

Specifies whether the NET time is used as a reference for coloring the function box that is displayed in the callgraph display area of the [Callgraph Window](#).

If this item is checked, the NET time is used as a reference for coloring the function box.

- [GROSS Time]

Specifies whether the GROSS time is used as a reference for coloring the function box that is displayed in the callgraph display area of the [Callgraph Window](#).

If this item is checked, the GROSS time is used as a reference for coloring the function box.

- [Execute Count]

Specifies whether the number of times of execution is used as a reference for coloring the function box that is displayed in the callgraph display area of the [Callgraph Window](#).

If this item is checked, the number of times of execution is used as a reference for coloring the function box.

- [I-Cache Miss]

Specifies whether the number of instruction cache misses is used as a reference for coloring the function box that is displayed in the callgraph display area of the [Callgraph Window](#).

If this item is checked, the number of instruction cache misses is used as a reference for coloring the function box.

- [Coloring Threshoud...]

Opens a dialog box ([Input Dialog Box](#)) to specify the coloring boundary value of the function box displayed in the callgraph display area of the [Callgraph Window](#).

The range of values that can be specified is from 0% to 100%. 10% is specified in the default condition.

- [Expand], [Shrink]

By selecting these items after selecting a function box displayed in the callgraph display area of the [Callgraph Window](#), the selected function box is expanded (displays the child function) or folded (not displays the child function).

- [Zoom...]

By selecting this item after selecting the [Callgraph Window](#), a window ([Input Dialog Box](#)) to change the display magnification of the callgraph display area is opened.

The selectable items are "200%", "150%", "100%", "75%", "50%", "Fit To Window".

Text can also be input.

The range of values that can be specified is from 10% to 100%.

- [Change Root Function...]

By selecting this item after selecting the [Callgraph Window](#), a dialog box ([Input Dialog Box](#)) to change the root function displayed in the callgraph display area is opened.

All the functions recognized by the TW850 can be selected.

Remark If two or more functions of the same name exist, "function name (file name)" is displayed.

If two or more functions of the same function name or file name exist, "function name (absolute path name)" is displayed.

- [Hide Execution Time 0 Functions], [Hide Execution Count 0 Functions]

Specifies whether a function with execution time of 0 or an execution count of 0 is displayed when an internal window (such as the [Callgraph Window](#), [Function Information Table Window](#), and [Execution Module Information Window](#)) is displayed.

If this item is selected, the functions with an execution time of 0 or an execution count of 0 are not displayed.

- [Function Transition]

This item consists of the following items.

 - [Sort]

Selects a condition to sort the function boxes and interrupt boxes in the function transition information display area of the [Function Transition Information Window](#).

This item consists of the following items.

 - [Name]

Sorts the function boxes and interrupt boxes in name order in the function transition information display area of the [Function Transition Information Window](#).

However, the order that the function boxes, interrupt boxes, and other boxes are displayed is not changed.
 - [Address]

Sorts the function boxes and interrupt boxes in address order in the function transition information display area of the [Function Transition Information Window](#).

However, the order that the function boxes, interrupt boxes, and other boxes are displayed is not changed.
 - [Eile]

Sorts the function boxes and interrupt boxes in file order in the function transition information display area of the [Function Transition Information Window](#).

However, the order that the function boxes, interrupt boxes, and other boxes are displayed is not changed.
 - [Select Display Function...]

Opens a dialog box ([Select Display Function Dialog Box](#)) to select the function box and interrupt box displayed in the function transition information display area of the [Function Transition Information Window](#).
 - [Hide Transition Between Hide Function]

Specifies whether the function transition line of the hidden function is displayed in the function transition information display area of the [Function Transition Information Window](#).

If this item is checked, the function transition line of the hidden function is displayed.

Remark This item is invalid if the following condition is satisfied.

- If the [Function Transition Information Window](#) is not selected

(d) [Tool] menu

This menu consists of the following items.

- [Tuning Wizard...]

Starts wizard processing (performance analysis or tuning) of the TW850.

By selecting this item, a panel ([Initial Screen Panel](#)) to report that the TW850 has started wizard processing (performance analysis or tuning) is opened.
- [Option...]

Starts option setting processing.

By selecting this item, a dialog box ([Option Dialog Box](#)) to specify how to print the [Callgraph Window](#) and [Function Transition Information Window](#), or whether the intermediate files (temporary files) the TW850 output while it was executing performance analysis or tuning processing are to be deleted.

(e) [Window] menu

This menu consists of the following items.

- [Cascade]

Overlaps and displays the internal window (cascade display).
- [Tile]

Arranges and displays the internal window (tile display).

- [Split]
Specifies whether the function transition information display area of the [Function Transition Information Window](#) is split into two right and left.
If this item is checked, the function transition information display area of the [Function Transition Information Window](#) is split into two right and left.

Remark This item is invalid if the following condition is satisfied.

- If the [Function Transition Information Window](#) is not selected

(f) [Help] menu

This menu consists of the following items.

- [TW Help]
Starts the help related to the TW850.
- [Main Window]
Starts the help related to this window.
- [Current Window F1]
Starts the help related to the window, panel, or dialog box in the active status.
- [About...]
Opens a dialog box ([Version Information Dialog Box](#)) to display the version information (such as version and copyright) of the TW850.

(2) Tool bar

The tool bar consists of the following buttons and combo box.

(a) Tuning Wizard button



Starts wizard processing (performance analysis or tuning) of the TW850.

By selecting this button, a panel ([Initial Screen Panel](#)) to report that the TW850 has started wizard processing (performance analysis or tuning) is opened.

Remark When the mouse cursor is placed on this button, "Start Tuning Wizard to analyze and tune an execution module." is displayed in a pop-up menu.

(b) Add To History button



By clicking this button after selecting an item (such as [Original Execution Module Information] or [Tuned Execution Module Information]) displayed on [Newest Information] in the history display area, the selected item is added and registered to [History Information].

Up to 20 items can be added to [History Information].

Remark When the mouse cursor is placed on this button, "Add selected item to history information." is displayed in a pop-up menu.

(c) Restore From History button



By clicking this button after selecting an item (such as [Original Execution Module Information] or [Tuned Execution Module Information]) displayed on [History Information] in the history display area, the setting of the selected item is reflected on the wizard.

Remark When the mouse cursor is placed on this button, "Restore selected history information to this system." is displayed in a pop-up menu.

(d) Print button



By selecting this button after selecting an internal window (the [Callgraph Window](#), [Function Information Table Window](#), or [Function Transition Information Window](#)), the selected window is printed.

Remark When the mouse cursor is placed on this button, "Print selected window." is displayed in a pop-up menu.

(e) Root function change combo box



By selecting this combo box after selecting the [Callgraph Window](#), the root function displayed in the callgraph display area is changed.

All the functions recognized by the TW850 can be selected.

- Remarks 1. If two or more functions of the same name exist, "function name (file name)" is displayed.
If two or more functions of the same function name or file name exist, "function name (absolute path name)" is displayed.
2. When the mouse cursor is placed on this combo box, "Change root function." is displayed in a pop-up menu.

(f) Function Detail button



By clicking this button after selecting a function name displayed in the [Callgraph Window](#), [Function Information Table Window](#), or [Function Transition Information Window](#), a window ([Function Detail Information Window](#)) to display the detailed function information of the execution module (performance analysis information, parent function information, child function information) is opened.

Remark When the mouse cursor is placed on this button, "Show detail about selected function." is displayed in a pop-up menu.

(3) History display area

This area displays the result history of wizard processing (performance analysis or tuning) in tree form. Whether this display area is to be displayed or not can be specified by the [\[View\]](#) menu -> [\[History\]](#). The items displayed in this display area are as follows.

(a) [\[Newest Information\]](#) item

This item displays the latest result of wizard processing (performance analysis or tuning) executed by the TW850.

(b) [\[History Information\]](#) item

This item displays the results (up to 20 results in the past) of wizard processing (performance analysis or tuning) executed by the TW850.

(c) [\[Original Execution Module Information\]](#) item

This is a sub-item of the [\[Newest Information\]](#) and [\[History Information\]](#) item.
It displays the result of wizard processing (performance analysis) the TW850 has executed on the original execution module in light blue.

(d) [\[Tuned Execution Module Information\]](#) item

This is a sub-item of the [\[Newest Information\]](#) and [\[History Information\]](#) item.
It displays the result of wizard processing (performance analysis) the TW850 has executed on the tuned execution module in pink.

(e) [\[Execution Module Information\]](#) item

This is a sub-item of the [\[Original Execution Module Information\]](#) and [\[Tuned Execution Module Information\]](#) item.
By selecting the right mouse button menu -> [\[Open\]](#) or the [\[View\]](#) menu -> [\[Open\]](#) after selecting this item, a window ([Execution Module Information Window](#)) to display the execution module information (execution result and setting contents) of the corresponding execution module is opened.

(f) [\[Callgraph Information\]](#) item

This is a sub-item of the [\[Original Execution Module Information\]](#) and [\[Tuned Execution Module Information\]](#) item.
By selecting the right mouse button menu -> [\[Open\]](#) or the [\[View\]](#) menu -> [\[Open\]](#) after selecting this item, a window ([Callgraph Window](#)) to display the performance analysis information (such as callpair information and call count information) of the corresponding execution module is opened.

- (g) [Function Information Table] item
 This is a sub-item of the [Original Execution Module Information] and [Tuned Execution Module Information] item.
 By selecting the right mouse button menu -> [O]pen or the [V]iew menu -> [O]pen after selecting this item, a window ([Function Information Table Window](#)) to display the detailed function information of the corresponding execution module is opened.
- (h) [Function Transition Information] item
 This is a sub-item of the [Original Execution Module Information] and [Tuned Execution Module Information] item.
 By selecting the right mouse button menu -> [O]pen or the [V]iew menu -> [O]pen after selecting this item, a window ([Function Transition Information Window](#)) to display the function transition information of the corresponding execution module is opened.
- (4) Internal window display area
 This area displays an internal window (such as the [Callgraph Window](#), [Function Information Table Window](#), [Function Transition Information Window](#), [Execution Module Information Window](#), and [Function Detail Information Window](#)).
 The display format of this display area can be selected from the item (overlapped or arranged) that is displayed by selecting the [W]indow] menu.
- (5) Message display area
 This area displays the information to be reported to the user detected by the TW850.
 Whether this area is to be displayed or not can be specified by [V]iew] menu -> [M]essage].
- (6) Status bar
 This area displays the progress of the processing by the TW850.
 If the TW850 is executing processing that takes time, a progress bar is displayed.

Context menu

A context menu is displayed when a display item in the history display area is selected and the right mouse button is clicked.

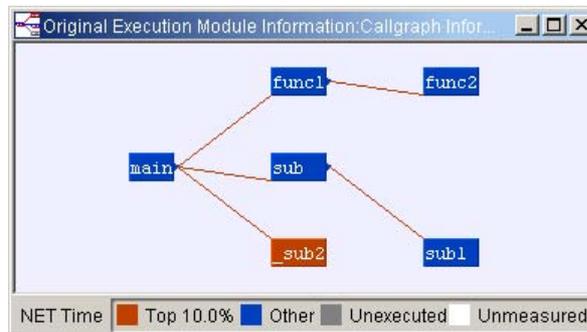
The context menu consists of the following items.

- (1) [O]pen
 Opens the window ([Callgraph Window](#), [Function Information Table Window](#), or [Execution Module Information Window](#)) corresponding to the selected item.
- (2) [A]dd To [H]istory
 Adds and registers the selected item to [History Information].
- (3) [R]estore From History...]
 Reflects the setting (such as performance analysis information and tuning information) of the selected item to the TW850.
- (4) [D]elete]
 Deletes the selected item.
- (5) [R]ename]
 Opens a dialog box to change the name of the selected item.

Callgraph Window

This window is used to display performance analysis information (such as callpair information and call count information) on an execution module in graph form.

Figure 6-2 Callgraph Window



This section describes the following items.

- [Opening](#)
- [Explanation of each area](#)
- [Context menu](#)

Opening

This window can be opened as follows.

- After selecting [Callgraph Information] in the history display area of the [Main Window](#),
 - Select the right mouse button menu -> [O]pen].
 - Select the [V]iew] menu -> [O]pen].

Explanation of each area

This window consists of the following items.

- [Callgraph display area](#)
- [Coloring reference display area](#)

(1) Callgraph display area

This area displays the information on the execution results of performance analysis (such as callpair information and call count information) in graph form.

The background color is light blue for the original execution module information, and pink for the tuned execution module information.

This display area consists of the following box and relational line.

(a) Function box

This box displays the function name of the target function in color.

If a function name of more than 20 characters is specified, the 18th and subsequent characters are displayed as "...".

If the target function calls a function (calls a child function), a black circle is shown on the right of the function box.

White: Cannot be measured

Gray: Function subject to performance analysis but not executed

Red: Function subject to performance analysis and in the higher xxx% category or above

Blue: Function subject to performance analysis and in the higher xxx% category or below

Remarks 1. Color boundary value xxx% can be changed in the [Input Dialog Box](#) that is opened by selecting [View] menu -> [Callgraph] -> [Coloring Threshoud...] on the [Main Window](#).

2. If a function box is clicked, the function box is selected, and the parent function box and child function box of the target function, and the calling relational line connecting the root function box and target function box is emphasized and displayed.

If another Callgraph Window or [Function Information Table Window](#) is displaying the target function, the TW850 executes the selection operation for each target function.

3. If a function box is double-clicked, the [Function Detail Information Window](#) to display the detailed information of the target function is opened.

If the function box is dragged, the target function box can be moved.

4. If an area other than a function box and a calling relational line is clicked, all the function boxes are not selected.

5. "Higher" means the "higher" sort result of the color display format (such as coloring by NET time or GROSS time) specified using the [Callgraph] on the [Main Window](#).

(b) Calling relational line

This line displays the calling relationship between functions (parent function and child function) in color.

Gray: The target child function is called 0 times.

Red: The target child function is called the higher xxx% or more of the total number of times of calling in the range subject to performance analysis.

Blue: The target child function is called the higher xxx% or less of the total number of times of calling in the range subject to performance analysis.

Remarks 1. Color boundary value xxx% can be changed in the [Input Dialog Box](#) that is opened by selecting [View] menu -> [Callgraph] -> [Coloring Threshoud...] on the [Main Window](#).

2. The function box in which a calling relational line is drawn from the right center indicates a parent function, and the function box in which the calling relational line is drawn from the left corner indicates a child function.

A recursive function is indicated by a curve drawn from the right center to the left corner in the TW850.

3. If a calling relational line is clicked, all the function boxes are not selected.

4. "Higher" means the "higher" sort result of the color display format (such as coloring by NET time or coloring by GROSS time) specified using the [Callgraph] on the [Main Window](#).

(2) Coloring reference display area

This area displays the color reference (such as for coloring by NET time, coloring by GROSS time, coloring by the number of times of execution, and coloring by the number of instruction cache misses) in the function box.

The color reference in the function box can be changed on the menu (such as for coloring by NET time, coloring by GROSS time, coloring by the number of times of execution, and coloring by the number of instruction cache misses) that is displayed by selecting [View] menu -> [Callgraph] on the [Main Window](#).

Context menu

A context menu is displayed if the right mouse button is clicked in the following areas.

(1) Function box

The context menu displayed when the right mouse button is clicked in the function box consists of the following items.

(a) [Set To Root Function]

Redraws the function at the selected position in the callgraph display area as a root function.

(b) [Open Function Detail]

Opens a window ([Function Detail Information Window](#)) to display the detailed information (performance analysis information, parent function information, and child function information) on the function at the selected position.

(c) [Expand], [Shrink]

Expands or folds the function box at the selected position.

(2) Areas other than function box

The context menu displayed when the right mouse button is clicked in an area other than the function box consists of the following items.

(a) [NET Time]

Specifies whether the NET time is used as a reference for coloring the function box.

(b) [GROSS Time]

Specifies whether the GROSS time is used as a reference for coloring the function box.

(c) [Execute Count]

Specifies whether the number of times of execution is used as a reference for coloring the function box.

(d) [I-Cache Miss]

Specifies whether the number of instruction cache misses is used as a reference for coloring the function box.

(e) [Coloring Threshoud...]

Opens a dialog box ([Input Dialog Box](#)) to specify the coloring boundary value of the function box.

(f) [Zoom...]

Opens a dialog box ([Input Dialog Box](#)) to change the display magnification of the callgraph display area.

(g) [Change Root Function...]

Opens a dialog box ([Input Dialog Box](#)) to change the root function displayed in the callgraph display area.

(h) [Hide Execution Time 0 Functions], [Hide Execution Count 0 Functions]

Specifies whether functions with an execution time of 0 or an execution count of 0 are displayed.

Function Information Table Window

This window is used to display a list of function information (such as the function name, number of times of execution, and NET time) of an execution module in table form.

Figure 6-3 Function Information Table Window

Function Name	Execution Count	NET Time[ns]	Average NET Time[ns]	GROSS Time[ns]	Average GROSS Time[ns]	Unknown Function Time[ns]	I-Cache Miss Count
func2	5	10,200 (10.71%)	2,040 (2.14%)	10,200 (10.71%)	2,040 (2.14%)	0 (0.00%)	0
_sub2	1	48,520 (50.98%)	48,520 (50.98%)	53,600 (56.32%)	53,600 (56.32%)	5,080 (5.33%)	0
func1	1	2,320 (2.43%)	2,320 (2.43%)	12,520 (13.15%)	12,520 (13.15%)	0 (0.00%)	0
main	1	1,240 (1.30%)	1,240 (1.30%)	95,160 (100.00%)	95,160 (100.00%)	5,080 (5.33%)	0
sub	1	24,800 (26.06%)	24,800 (26.06%)	27,800 (29.21%)	27,800 (29.21%)	0 (0.00%)	0
sub1	1	3,000 (3.15%)	3,000 (3.15%)	3,000 (3.15%)	3,000 (3.15%)	0 (0.00%)	0
Unknown Function		5,080 (5.33%)					0

This section describes the following items.

- [Opening](#)
- [Explanation of each area](#)
- [Context menu](#)

Opening

This window can be opened as follows.

- After selecting [Function Information Table] in the history display area of the [Main Window](#),
 - Select the right mouse button menu -> [Open].
 - Select the [View] menu -> [Open].

Explanation of each area

This window consists of the following items.

- [Function information display area](#)

(1) Function information display area

This area displays a list of the information on the execution results of performance analysis (function information: such as function name, number of times of execution, and NET time) in table form.

The background color is light blue for the original execution module information, and pink for the tuned execution module information.

This display area consists of the following display areas.

(a) Function name display area

This area displays the name of the target function.

Remark If two or more functions of the same name exist, "function name (file name)" is displayed.
If two or more functions of the same function name or file name exist, "function name (absolute path name)" is displayed.

(b) Number of times of execution display area

This area displays the number of times the target function has been executed.

If an overflow occurs, "(overflow)" is displayed following the number of times of execution.

Remark This display area is hidden if the following condition is satisfied.

- If the probe code insertion method is used for performance analysis processing and in the case of tuned execution module information

(c) NET time (not including execution time of child function) display area

This area displays the NET time (in ns or in clock) of the target function and the proportion (in %) of the range subject to performance analysis to the total execution time.

If the target function is a tuned execution module, the comparison of the performance with the execution module for original analysis is displayed in color.

If an overflow occurs, "(overflow)" is displayed following the execution time.

Blue: The NET time is shorter than the execution module for original analysis.
Black: The NET time is the same as the execution module for original analysis.
Red: The NET time is longer than the execution module for original analysis.

Remark This display area is hidden if the following condition is satisfied.

- If the probe code insertion method is used for performance analysis processing and if the execution time of each target function is not measured by using a time measurement file

(d) Average NET time display area

This area displays the average NET time (in ns or in clock) of the target function and the proportion (in %) of the range subject to performance analysis to the total execution time.

If the target function is a tuned execution module, the comparison of the performance with the execution module for original analysis is displayed in color.

If an overflow occurs, "(overflow)" is displayed following the execution time.

Blue: The average NET time is shorter than the execution module for original analysis.
Black: The average NET time is the same as the execution module for original analysis.
Red: The average NET time is longer than the execution module for original analysis.

Remark This display area is hidden if the following condition is satisfied.

- If the NET time display area is hidden

(e) GROSS time (including execution time of child function) display area

This area displays the GROSS time (in ns or in clock) of the target function and the proportion (in %) of the range subject to performance analysis to the total execution time.

If the target function is a tuned execution module, the comparison of the performance with the execution module for original analysis is displayed in color.

If an overflow occurs, "(overflow)" is displayed following the execution time.

Blue: The GROSS time is shorter than the execution module for original analysis.
Black: The GROSS time is the same as the execution module for original analysis.
Red: The GROSS time is longer than the execution module for original analysis.

Remark This display area is hidden if the following condition is satisfied.

- If the probe code insertion method is used for performance analysis processing

(f) Average GROSS time display area

This area displays the average GROSS time (in ns or in clock) of the target function and the proportion (in %) of the range subject to performance analysis to the total execution time.

If the target function is a tuned execution module, the comparison of the performance with the execution module for original analysis is displayed in color.

If an overflow occurs, "(overflow)" is displayed following the execution time.

Blue: The average GROSS time is shorter than the execution module for original analysis.

Black: The average GROSS time is the same as the execution module for original analysis.

Red: The average GROSS time is longer than the execution module for original analysis.

Remark This display area is hidden if the following condition is satisfied.

- If the GROSS time display area is hidden

(g) Unknown function (function not subject to performance analysis) execution time display area

This area displays the GROSS time (in ns or in clock) of the unknown function called by the target function and the proportion of the range subject to performance analysis to the total execution time.

If the target function is a tuned execution module, the comparison of the performance with the execution module for original analysis is displayed in color.

If an overflow occurs, "(overflow)" is displayed following the execution time.

Blue: The unknown function execution time is shorter than the execution module for original analysis.

Black: The unknown function execution time is the same as the execution module for original analysis.

Red: The unknown function execution time is longer than the execution module for original analysis.

Remark This display area is invalid if the following condition is satisfied.

- If the probe code insertion method is used for performance analysis processing

(h) Instruction cache miss display area

This area displays the number and occurrence rate of instruction cache misses in the target function.

The occurrence rate is calculated by the following expression from the number of instruction cache hits and misses in the target function.

$$\text{Number of instruction cache misses} / (\text{Number of instruction cache hits} + \text{Number of instruction cache misses})$$

Remark This display area is invalid if either of the following conditions is satisfied.

- If the probe code insertion method is used for performance analysis processing
- If the trace data analysis method is not used for performance analysis processing and if an execution engine that can acquire cache miss information from trace data is not specified

Caution If settings to acquire the information of each item are not made and if an execution engine cannot be acquired the information, the item is not displayed.

Remarks 1. By clicking the header of each item, the function information can be sorted and displayed on the basis of the specified item.

However, unknown functions and the system blocks of RTOS are not sorted.

If the header of each item is dragged, the display order of each item can be changed.

2. If the information of each function is double-clicked, the [Function Detail Information Window](#) to display the detailed information of the target function is opened.

If the function box is dragged, the target function box can be moved.

Context menu

A context menu is displayed when an applicable function is selected and the right mouse button is clicked in the function information table area.

The context menu consists of the following item.

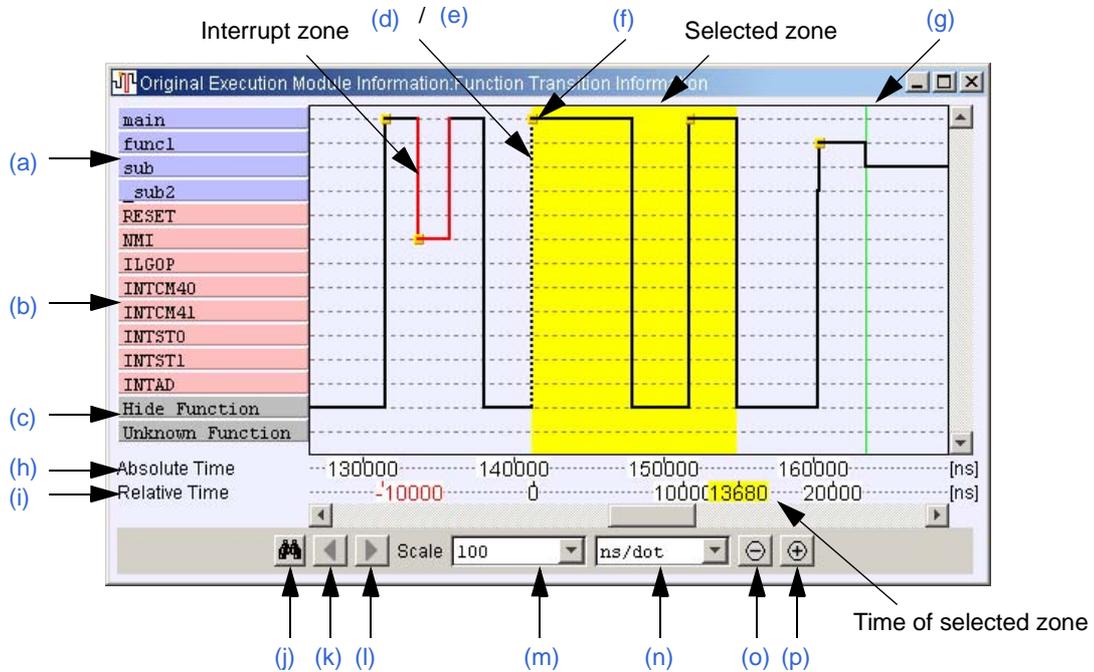
(1) [Open Function Detail]

Opens a window ([Function Detail Information Window](#)) to display the detailed information of the applicable function (performance analysis information, parent function information, and child function information).

Function Transition Information Window

This window is used to display function transition information on an execution module in graph form.

Figure 6-4 Function Transition Information Window



This section describes the following items.

- Opening
- Explanation of each area
- Context menu

Opening

This window can be opened as follows.

- After selecting [Function Transition Information] in the history display area of the [Main Window](#),
 - Select the right mouse button menu -> [Open].
 - Select the [View] menu -> [Open].

Explanation of each area

This window consists of the following items.

- [Function transition information display area](#)

(1) Function transition information display area

This area displays the information on the execution results of performance analysis (function transition information) in graph form.

The background color is light blue for the original execution module information, and pink for the tuned execution module information.

This display area consists of the following areas.

(a) Function box

This box displays the name (function symbol) of the target function.

If a function name of more than 20 characters is specified, the 18th and subsequent characters are displayed as "...".

Remarks 1. When the function box is clicked, the function box is selected. If the [Callgraph Window](#), [Function Information Table Window](#), or other Function Transition Information window is displayed, the same function on that window is also selected.

2. If a function box is double-clicked, the [Function Detail Information Window](#) to display the detailed information of the target function is opened.

If the function box is dragged, the target function box can be moved between function boxes.

(b) Interrupt Box

This box displays the interrupt name of the target interrupt.

Remarks 1. When the interrupt box is clicked, the interrupt box is selected. If other Function Transition Information window is displayed, the same interrupt on that window is also selected.

2. If the interrupt box is dragged, the target interrupt box can be moved between interrupt boxes.

(c) Other Box

This boxes display "Hide Function", the RTOS system name, "Unknown Function".

The RTOS system name differs depending on the recognized RTOS. If the recognized RTOS is the RX850, "RX850 System" is displayed. If the recognized RTOS is the RX850 Pro, "RX850 Pro System" is displayed.

Remark When other box is clicked, the box is selected. If other Function Transition Information window is displayed, the same box on that window is also selected.

(d) Function execution line

This line displays the execution of a function.

The interrupt zone is displayed in red, and the other zones are displayed in black.

Remarks 1. When the function execution line is clicked, the function execution line and function box are selected. If the [Callgraph Window](#), [Function Information Table Window](#), or other Function Transition Information window is displayed, the same function on that window is also selected.

2. If a function execution line is double-clicked, the [Function Detail Information Window](#) to display the detailed information of the target function is opened.

3. If a function execution line is selected and then dragged, a zone is selected from the beginning of the selected function execution line in function transition units. If the mouse cursor is moved to the selected zone, the time of the selected zone (unit: nanoseconds) is displayed in a pop-up menu.

4. When the mouse cursor is placed on the function execution line, the function name (interrupt name), start time, execution time are displayed in a pop-up menu.

(e) Function transition line

This line displays the transition of a function.

The interrupt zone is displayed in red, and the other zones are displayed in black.

Remarks 1. If a function transition line is clicked, the function transition line is selected.

2. If a function transition line is selected and then dragged, a zone is selected from the beginning of the selected function transition line in function transition units. If the mouse cursor is moved to the selected zone, the time of the selected zone (unit: nanoseconds) is displayed in a pop-up menu.

3. When the mouse cursor is placed on the function transition line, the transition source function name (interrupt name), transition destination function name (interrupt name), transition time are displayed in a pop-up menu.

(f) Function head marker

This marker is displayed at the top of the function that shifts and at the top of an interrupt.

Caution The function head marker is not displayed for a hide function, the RTOS system, and an unknown function.

(g) Trace data delimiting line

This line displays the time position of trace data breaks in a zone subject to performance analysis.

(h) Absolute time scale

This scale displays the time scale from the beginning of a zone subject to performance analysis.

The time display interval is fixed every 100 dots. The display unit is the time unit specified in the unit combo box.

(i) Relative time scale

This scale displays the time scale from the beginning of the selected function execution line or function transition line.

The time display interval is fixed every 100 dots. The display unit is the time unit specified in the unit combo box. Negative relative time is displayed in red.

Remark If the selected zone exists, the time of the selected zone is displayed on the scale.

(j) Search button

Opens a dialog box ([Search Function Transition Information Dialog Box](#)) to search the transition of a function.

Remark When the mouse cursor is placed on this button, "Search for function or function transition." is displayed in a pop-up menu.

(k) Search Before button

Searches the function execution line or function transition line of the condition (function or interrupt) specified in the [Search Function Transition Information Dialog Box](#) in the left direction, starting from the selected function execution line or function transition line.

Remark When the mouse cursor is placed on this button, "Search backward." is displayed in a pop-up menu.

(l) Search Next button

Searches the function execution line or function transition line of the condition (function or interrupt) specified in the [Search Function Transition Information Dialog Box](#) in the right direction, starting from the selected function execution line or function transition line.

Remark When the mouse cursor is placed on this button, "Search forward." is displayed in a pop-up menu.

(m) Scale combo box

Specifies the clock or time indicated by one dot of the function transition information display area.

The selectable items are 1, 2, 5, 10, 25, 50, 100, 250, 500, and 1000.

Text can also be input. The range of values that can be input is from 1 to 1000.

1 is specified in the default condition.

Remark When the mouse cursor is placed on this combo box, "Clocks or time per a dot." is displayed in a pop-up menu.

(n) Unit combo box

Specifies the unit of the clock or time indicated by one dot of the function transition information display area. The selectable items are Mclock/dot, ms/dot Kclock/dot, us/dot clock/dot, and ns/dot. In the default condition, ns/dot is specified if the execution time can be acquired in nanoseconds, and clock/dot is specified if only the clock can be acquired.

Remarks 1. The unit size relationship is as follows.

Mclock/dot, ms/dot > Kclock/dot, us/dot > clock/dot, ns/dot

When the item specified in the unit combo box is changed, the item specified in the scale combo box is changed. If the unit is decreased, it is changed to 1000. If the unit is increased, it is changed to 1.

2. When the mouse cursor is placed on this combo box, "Unit of clocks or time per a dot." is displayed in a pop-up menu.

(o) Reduce button

Changes the items specified in the scale combo box of the function transition information display area in the following sequence and reduces the display range.

1, 2, 5, 10, 25, 50, 100, 250, 500, 1000

Remarks 1. If a value other than the above is specified in the scale combo box, it is changed to the value closest to the above. If 1000 is specified in the scale combo box, the item specified in the unit combo box is increased by one, and the item specified in the scale combo box is changed to 2.

2. This button is invalid if the following condition is satisfied.

- If 1000 in the scale combo box is selected and Mclock/dot or ms/dot in the unit combo box is selected

3. When the mouse cursor is placed on this button, "Zoom out" is displayed in a pop-up menu.

(p) Expand button

Changes the items specified in the scale combo box of the function transition information display area in the following sequence and expands the display range.

1000, 500, 250, 100, 50, 25, 10, 5, 2, 1

Remarks 1. If a value other than the above is specified in the scale combo box, it is changed to the value closest to the above. If 1 is specified in the scale combo box, the item specified in the unit combo box is decreased by one, and the item specified in the scale combo box is changed to 500.

2. This button is invalid if the following condition is satisfied.

- If 1 in the scale combo box is selected and clock/dot or ns/dot in the unit combo box is selected

3. When the mouse cursor is placed on this button, "Zoom in" is displayed in a pop-up menu.

Context menu

A context menu is displayed if the right mouse button is clicked in the following areas.

(1) Function box

The context menu displayed when the right mouse button is clicked in the function box consists of the following items.

(a) [H]ide This Function]

Hides the corresponding function and redraws the function transition information display area.

(b) [O]pen Function D]etail]

Opens a window ([Function Detail Information Window](#)) to display the detailed information (performance analysis information, parent function information, and child function information) on the function at the selected position.

(2) Interrupt Box

The context menu displayed when the right mouse button is clicked in the interrupt box consists of the following menu.

(a) [H]ide This Interrupt]

Hides the corresponding interrupt and redraws the function transition information display area.

(3) Area other than function box and interrupt box

The context menu displayed when the right mouse button is clicked in an area other than the function boxes and interrupt boxes consists of the following items.

(a) [S]ort]

Selects a condition to sort the function boxes and interrupt boxes.

This menu consists of the following menus.

- [N]ame]

Sorts the function boxes and interrupt boxes in name order.

However, the order that the function boxes, interrupt boxes, and other boxes are displayed is not changed.

- [A]ddress]

Sorts the function boxes and interrupt boxes in address order.

However, the order that the function boxes, interrupt boxes, and other boxes are displayed is not changed.

- [E]ile]

Sorts the function boxes and interrupt boxes in file order.

However, the order that the function boxes, interrupt boxes, and other boxes are displayed is not changed.

(b) [S]elect D]isplay Function...]

Opens a dialog box ([Select Display Function Dialog Box](#)) to select the function box and interrupt box.

(c) [H]ide Transition Between Hide Function]

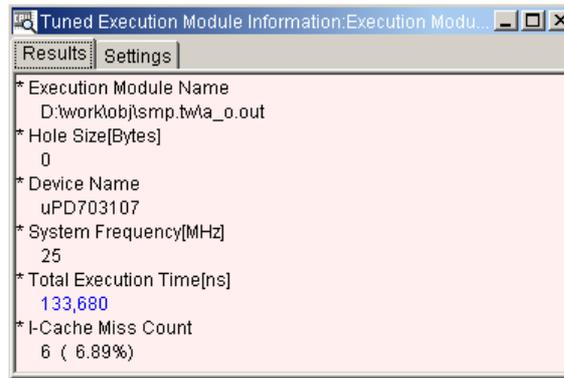
Specifies whether the function transition line of the hidden function is displayed.

If this item is checked, the function transition line of the hidden function is displayed.

Execution Module Information Window

This window is used to display execution module information (execution results and setting contents) of an execution module.

Figure 6-5 Execution Module Information Window



This section describes the following items.

- [Opening](#)
- [Explanation of each area](#)

Opening

This window can be opened as follows.

- After selecting [Execution Module Information] in the history display area of the [Main Window](#),
 - Select the right mouse button menu -> [Open].
 - Select the [View] menu -> [Open].

Explanation of each area

This window consists of the following items.

- [Results] tab
- [Settings] tab

(1) [Results] tab

This tab displays the information on the execution results of performance analysis (execution result information: such as execution module name, device name, and total execution time).

The background color is light blue for the original execution module information, and pink for the tuned execution module information.

This tab consists of the following display areas.

(a) Execution module name display area

This area displays the file name (execution module name) of the execution module.

(b) Hole size display area

This area displays the size of the hole created by tuning processing.

Remark This display area is hidden if the following condition is satisfied.

- For the original execution module information

(c) Device name display area

This area displays the target device name of the execution module.

(d) Operating frequency display area

This area displays the operating frequency (in MHz) of the execution module.

(e) Total execution time display area

This area displays the total execution time (in ns and in clock) in the range subject to performance analysis.

If the applicable execution module is a tuned execution module, the comparison of the performance with the execution module for original analysis is displayed in color.

Blue: The total execution time is shorter than the execution module for original analysis.

Black: The total execution time is the same as the execution module for original analysis.

Red: The total execution time is longer than the execution module for original analysis.

(f) Instruction cache miss information display area

This area displays the number and occurrence rate of instruction cache misses in the range subject to performance analysis.

The occurrence rate is calculated by the following expression from the number of instruction cache hits and instruction cache misses in the range subject to performance analysis.

$$\text{Number of instruction cache misses} / (\text{Number of instruction cache hits} + \text{Number of instruction cache misses})$$

Remark This display area is invalid if either of the following conditions is satisfied.

- If the probe code insertion method is used for performance analysis processing
- If the trace data analysis method is not used for performance analysis processing, and if an execution engine that can acquire cache miss information and cache hit information from trace data is not specified

(2) [Settings] tab

This tab displays the information related to the settings of performance analysis (setting contents information: such as execution contents, execution start position, execution analysis range).

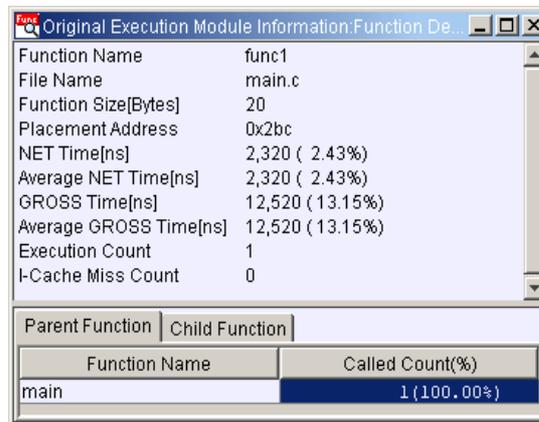
The contents set on each panel of the wizard are displayed.

The background color is light blue for the original execution module information, and pink for the tuned execution module information.

Function Detail Information Window

This window is used to display detailed function information (performance analysis information, parent function information, and child function information) of an execution module.

Figure 6-6 Function Detail Information Window



This section describes the following items.

- [Opening](#)
- [Explanation of each area](#)

Opening

This window can be opened as follows.

- On the [Main Window](#) after selecting a function name displayed in the callgraph display area of the [Callgraph Window](#),
 - Click the Function Detail button.
 - Select the [V]iew menu -> [F]unction Detail.
- On the [Callgraph Window](#),
 - Select (double-click) a function name displayed in the callgraph display area.
- On the [Main Window](#) after selecting a function name displayed in the function name display area of the [Function Information Table Window](#),
 - Click the Function Detail button.
 - Select the [V]iew menu -> [F]unction Detail.
- On the [Function Information Table Window](#),
 - Select (double-click) a function name displayed in the function name display area.
- On the [Main Window](#) after selecting a function execution line displayed in the function transition information display area of the [Function Transition Information Window](#),
 - Click the Function Detail button.
 - Select the [V]iew menu -> [F]unction Detail.

- On the [Function Transition Information Window](#),
 - Select (double-click) a function box displayed in the function transition information display area.
 - Select (double-click) a function execution line displayed in the function transition information display area.
- On the [Main Window](#) after selecting a function name displayed on the [Parent Function] tab or [Child Function] tab of this window,
 - Click the Function Detail button.
 - Select the [View] menu -> [Function Detail].
- On this window,
 - Select (double-click) a function name displayed on the [Parent Function] tab or [Child Function] tab.

Explanation of each area

This window consists of the following items.

- [Performance analysis information display area](#)
- [\[Parent Function\] tab](#)
- [\[Child Function\] tab](#)

(1) Performance analysis information display area

This area displays the information on the execution results of performance analysis (performance analysis information: such as function name, file name, and function size).

The background color is light blue for the original execution module information, and pink for the tuned execution module information.

This display area consists of the following display areas.

(a) Function name display area

This area displays the name of the target function.

(b) File name display area

This area displays the name of the file in which the target function is described.

Remark If the function is defined in a library, "file name (library name)" is displayed.

(c) Function size display area

This area displays the size (in bytes) of the text area occupied by the target function.

(d) Function allocation address display area

This area displays the address (start address) to which the text area of the target function is allocated.

(e) NET time (not including execution time of child function) display area

This area displays the NET time (in ns or in clock) of the target function and the proportion (in %) of the range subject to performance analysis to the total execution time.

If the target function is a tuned execution module, the comparison of the performance with the execution module for original analysis is displayed in color.

If an overflow occurs, "(overflow)" is displayed following the execution time.

Blue: The NET time is shorter than the execution module for original analysis.

Black: The NET time is the same as the execution module for original analysis.

Red: The NET time is longer than the execution module for original analysis.

Remark This display area is hidden if the following condition is satisfied.

- If the probe code insertion method is used for performance analysis processing and if the execution time of each target function is not measured by using a time measurement file

(f) Average NET time display area

This area displays the average NET time (in ns or in clock) of the target function and the proportion (in %) of the range subject to performance analysis to the total execution time.

If the target function is a tuned execution module, the comparison of the performance with the execution module for original analysis is displayed in color.

If an overflow occurs, "(overflow)" is displayed following the execution time.

Blue: The average NET time is shorter than the execution module for original analysis.

Black: The average NET time is the same as the execution module for original analysis.

Red: The average NET time is longer than the execution module for original analysis.

Remark This display area is hidden if the following condition is satisfied.

- If the NET time display area is hidden

(g) GROSS time (including execution time of child function) display area

This area displays the GROSS time (in ns or in clock) of the target function and the proportion (in %) of the range subject to performance analysis to the total execution time.

If the target function is a tuned execution module, the comparison of the performance with the execution module for original analysis is displayed in color.

If an overflow occurs, "(overflow)" is displayed following the execution time.

Blue: The GROSS time is shorter than the execution module for original analysis.

Black: The GROSS time is the same as the execution module for original analysis.

Red: The GROSS time is longer than the execution module for original analysis.

Remark This display area is hidden if the following condition is satisfied.

- If the probe code insertion method is used for performance analysis processing

(h) Average GROSS time display area

This area displays the average GROSS time (in ns or in clock) of the target function and the proportion (in %) of the range subject to performance analysis to the total execution time.

If the target function is a tuned execution module, the comparison of the performance with the execution module for original analysis is displayed in color.

If an overflow occurs, "(overflow)" is displayed following the execution time.

Blue: The average GROSS time is shorter than the execution module for original analysis.

Black: The average GROSS time is the same as the execution module for original analysis.

Red: The average GROSS time is longer than the execution module for original analysis.

Remark This display area is hidden if the following condition is satisfied.

- If the GROSS time display area is hidden

(i) Number of times of execution display area

This area displays the number of times the target function has been executed.

If an overflow occurs, "(overflow)" is displayed following the number of times of execution.

Remark This display area is hidden if the following condition is satisfied.

- If the probe code insertion method is used for performance analysis processing and in the case of tuned execution module information

(j) Instruction cache miss display area

This area displays the number and occurrence rate of instruction cache misses in the target function.

The occurrence rate is calculated by the following expression from the number of instruction cache hits and instruction cache misses in the range subject to performance analysis.

Number of instruction cache misses / (Number of instruction cache hits + Number of instruction cache misses)

Remark This display area is invalid if any of the following conditions is satisfied.

- If the size of the instruction cache is 0
- If the probe code insertion method is used for performance analysis processing
- If the trace data analysis method is not used for performance analysis processing and if an execution engine that can acquire cache miss information from trace data is not specified

(2) [Parent Function] tab

This tab displays a list of information on the execution results of performance analysis (parent function information: such as parent function name, number of times function is called) in table form.

This tab consists of the following display areas.

(a) Function name display area

This area displays the parent function of the target function.

Remark If two or more functions of the same name exist, "function name (file name)" is displayed.
If two or more functions of the same function name or file name exist, "function name (absolute path name)" is displayed.

(b) Number of times called display area

This area displays the number of times the target function called by the parent function, and the proportion (in %) to the total number of times the target function is called.

If an overflow occurs, "(overflow)" is displayed following the number of times of execution.

Remark This display area is hidden if the following condition is satisfied.

- If the probe code insertion method is used for performance analysis processing and in the case of tuned execution module information

Remarks 1. By clicking the header of each item, the function information can be sorted and displayed on the basis of the specified item.

2. If the information of each function is double-clicked, the [Function Detail Information Window](#) to display the detailed information of the target function is opened.

(3) [Child Function] tab

This tab displays a list of information on the execution results of performance analysis (child function information: such as child function name, number of times function calls) in table form.

This tab consists of the following display areas.

(a) Child function name display area

This area displays the child function of the target function.

Remark If two or more functions of the same name exist, "function name (file name)" is displayed.
If two or more functions of the same function name or file name exist, "function name (absolute path name)" is displayed.

(b) Number of times of calling display area

This area displays the number of times the target function called the child function, and the proportion (in %) to the total number of times the target function calls.

If an overflow occurs, "(overflow)" is displayed following the number of times of execution.

Remark This display area is hidden if the following condition is satisfied.

- If the probe code insertion method is used for performance analysis processing and in the case of tuned execution module information

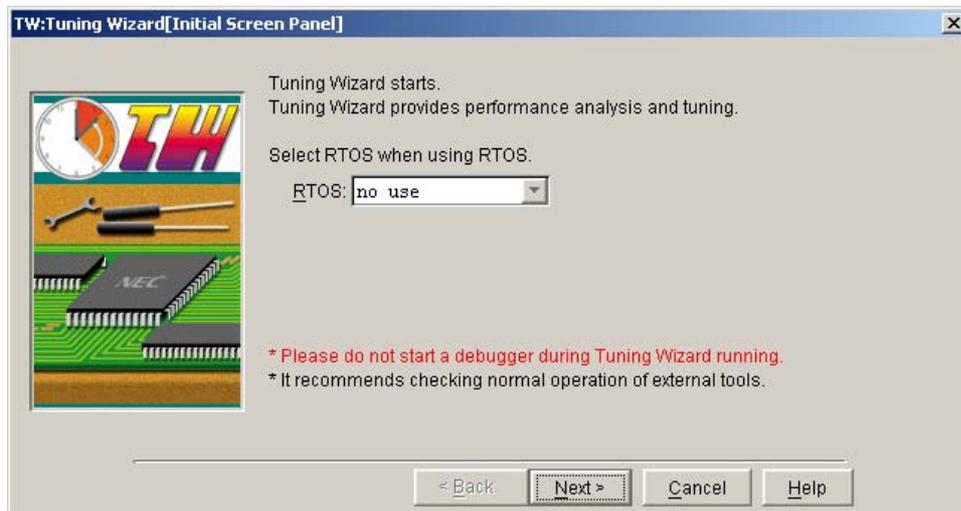
Remarks 1. By clicking the header of each item, the function information can be sorted and displayed on the basis of the specified item.

2. If the information of each function is double-clicked, the [Function Detail Information Window](#) to display the detailed information of the target function is opened.

Initial Screen Panel

This panel is used to report that the TW850 has started wizard processing (performance analysis or tuning) and specify the RTOS used for the program that executes performance analysis.

Figure 6-7 Initial Screen Panel



This section describes the following items.

- [Opening](#)
- [Explanation of each area](#)

Opening

This panel can be opened as follows.

- On the [Main Window](#),
 - Click the Tuning Wizard button.
 - Select the [Tool] menu -> [Tuning Wizard...].

Explanation of each area

This panel consists of the following items.

- [Message display area](#)
- [\[RTOS\] combo box](#)

(1) Message display area

This area displays a message reporting that the TW850 has started wizard processing (performance analysis or tuning).

(2) [RTOS] combo box

This combo box is used to specify the type of RTOS to be used or "no use".

"no use" is specified in the default condition.

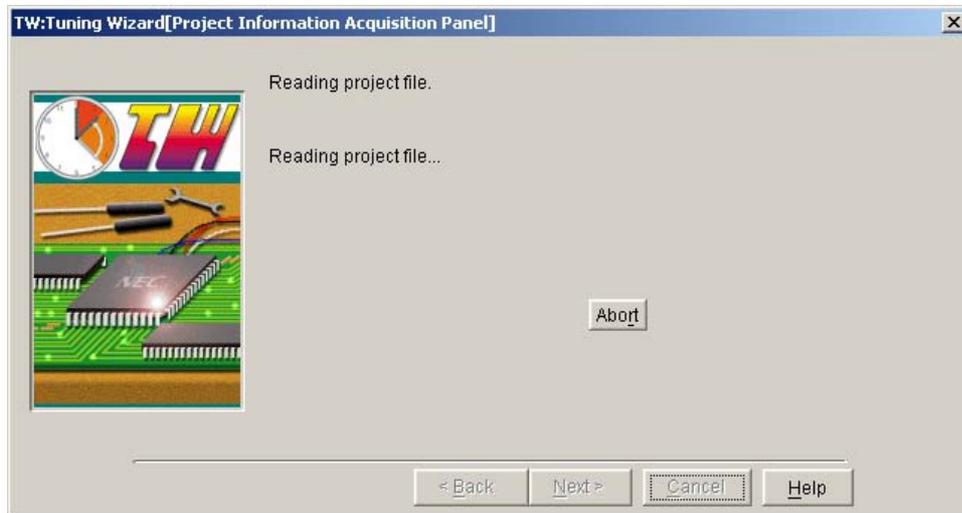
If an RTOS type is specified, the TW850 executes performance analysis on the assumption that an RTOS is used.

Caution If an RTOS is used, always specify the RTOS type.

Project Information Acquisition Panel

This panel is used to report that the TW850 is executing wizard processing (project information acquiring).

Figure 6-8 Project Information Acquisition Panel



This section describes the following items.

- [Explanation of each area](#)

Explanation of each area

This panel consists of the following items.

- [Message display area](#)

(1) Message display area

This area displays a message reporting that the TW850 is executing wizard processing (project information acquiring).

(a) <Abort> button

This button is used to stop wizard processing (project information acquiring).

Remark It may take time for wizard processing to actually stop after this button is clicked.

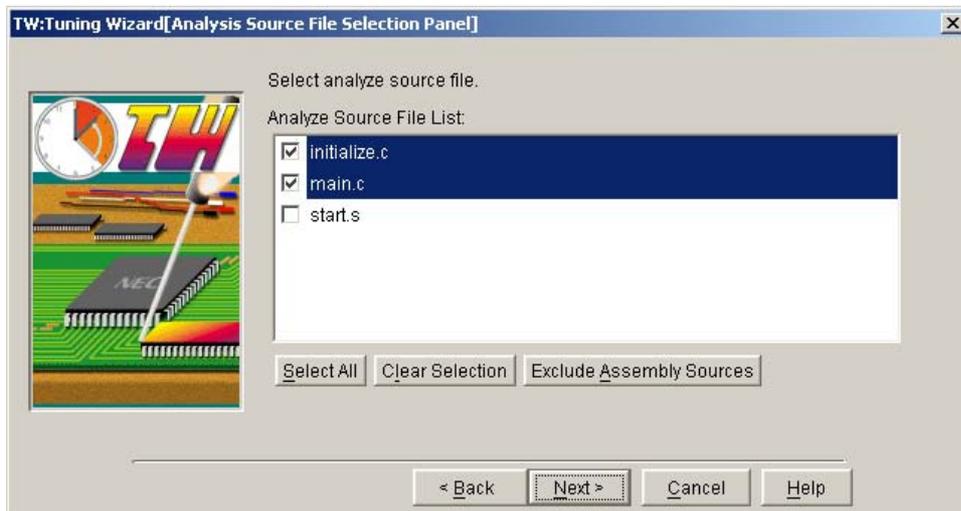
(b) <Retry> button

This button is used to re-execute wizard processing (project information acquiring).

Analysis Source File Selection Panel

This panel is used to specify source files subject to performance analysis or tuning during wizard processing (performance analysis or tuning) of the TW850.

Figure 6-9 Analysis Source File Selection Panel



This section describes the following items.

- [Explanation of each area](#)

Explanation of each area

This panel consists of the following items.

- [Message display area](#)
- [Analysis source file specification area](#)

(1) Message display area

This area displays an explanation messages related to this panel.

(2) Analysis source file specification area

This area is used to specify source files subject to performance analysis or tuning.

By limiting the source files subject to performance analysis or tuning, the time required for wizard processing (performance analysis or tuning) and the memory area on the target necessary for the probe code insertion method can be reduced. This is especially effective when there are many source files and functions.

Caution If there is an assembly source file that the TW850 cannot recognize, the TW850 cannot correctly perform wizard processing (performance analysis or tuning). Deselect the corresponding assembly source file to exclude it from subject to performance analysis or tuning.

Refer to the "[4.6.3 Description format of assembly source file](#)" for details of descriptions on an assembly source file.

This specification area consists of the following display area and buttons.

(a) Analysis source file list display area

This area is used to specify source files subject to performance analysis or tuning. Files other than assembly source files are selected in the default condition.

(b) <Select All> button

This button is used to select all source files displayed.

(c) <Clear Selection> button

This button is used to unselect all source files displayed.

(d) <Exclude Assembly Sources> button

This button is used to deselect only assembly source files in the source files displayed.

Execution Module Information Acquisition Panel

This panel is used to report that the TW850 is executing wizard processing (generation processing of an execution module for original analysis, and analysis processing of functions and an original execution module).

The execution contents of the TW850 are as follows.

- Output assembly source files for analysis
- Analyze function information
- Build and analyze an execution module for original analysis

Figure 6-10 Execution Module Information Acquisition Panel



This section describes the following items.

- [Explanation of each area](#)

Explanation of each area

This panel consists of the following items.

- [Message display area](#)

(1) Message display area

This area displays a message reporting that the TW850 is executing wizard processing (performance analysis processing: generating an execution module for original analysis and performance analysis).

(a) <Abort> button

This button is used to stop wizard processing (performance analysis processing: generating an execution module for original analysis and performance analysis).

Remark It may take time for wizard processing to actually stop after this button is clicked.

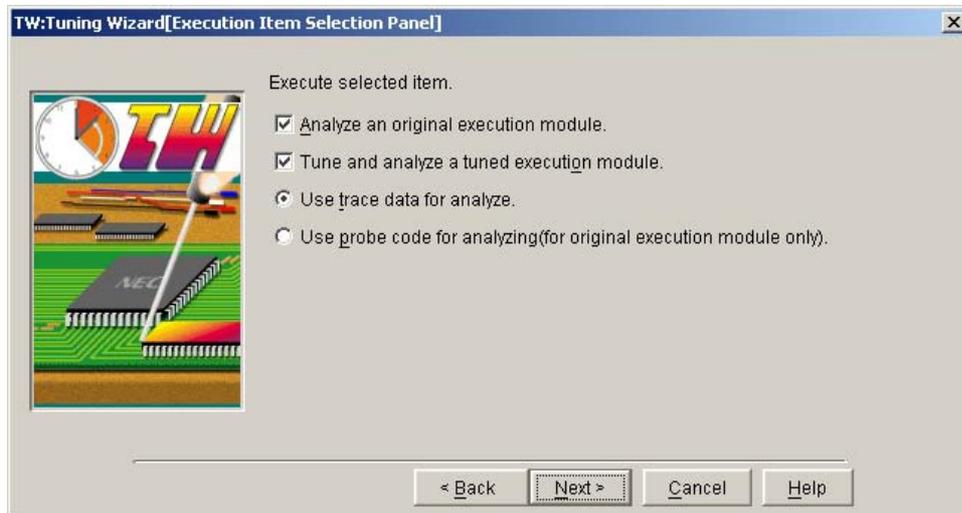
(b) <Retry> button

This button is used to re-execute wizard processing (performance analysis processing: generating an execution module for original analysis and performance analysis).

Execution Item Selection Panel

This panel is used to specify execution items for wizard processing (performance analysis or tuning) of the TW850.

Figure 6-11 Execution Item Selection Panel



This section describes the following items.

- [Explanation of each area](#)

Explanation of each area

This panel consists of the following items.

- [Message display area](#)
- [Execution item specification area](#)
- [Performance analysis information acquisition method specification area](#)

(1) Message display area

This area displays an explanation messages related to this panel.

(2) Execution item specification area

This area is used to make settings related to the execution items of wizard processing (performance analysis or tuning).

This specification area consists of the following check boxes.

(a) [Analyze an original execution module.] check box

This check box is used to specify whether performance analysis processing is executed on an original execution module as wizard processing of the TW850.

This check box is checked in the default condition.

If this check box is checked, performance analysis is executed on an original execution module.

(b) [Tune and analyze a tuned execution module.] check box

This check box is used to specify whether tuning processing on an original execution module (generation a tuned execution module) is performed and whether performance analysis processing on a tuned execution module is performed, as wizard processing of the TW850.

This check box is checked in the default condition.

If this check box is checked, tuning processing on an original execution module (generation a tuned execution module) and performance analysis processing on a tuned execution module is performed.

(3) Performance analysis information acquisition method specification area

This area is used to make settings related to the method of acquiring performance analysis information in wizard processing (performance analysis or tuning processing).

This specification area consists of the following radio buttons.

(a) [Use trace data for analyze.] radio button

This radio button is used to specify whether the trace data analysis method is used to acquire performance analysis information in wizard processing.

Refer to "[1\) Trace data analysis method](#)" for details of the trace data analysis method.

This check box is checked in the default condition.

If this radio button is checked, the trace data analysis method is used as the method to acquire performance analysis information. If the [Tune and analyze a tuned execution module.] check box is checked, the trace data analysis method is used not only for the original execution module but also the tuned execution module.

Remark This radio button is invalid if the following condition is satisfied.

- If the [Analyze an original execution module.] check box is not checked

(b) [Use probe code for analyzing (for original execution module only).] radio button

This radio button is used to specify whether the probe code insertion method is used to acquire performance analysis information in wizard processing.

Refer to "[2\) Probe code insertion method](#)" for details of the probe code insertion method.

If the [Use trace data for analyze.] radio button is not checked, this check box is checked in the default condition.

If this radio button is checked, the probe code insertion method is used as the method to acquire performance analysis information.

Remark This radio button is invalid if the following condition is satisfied.

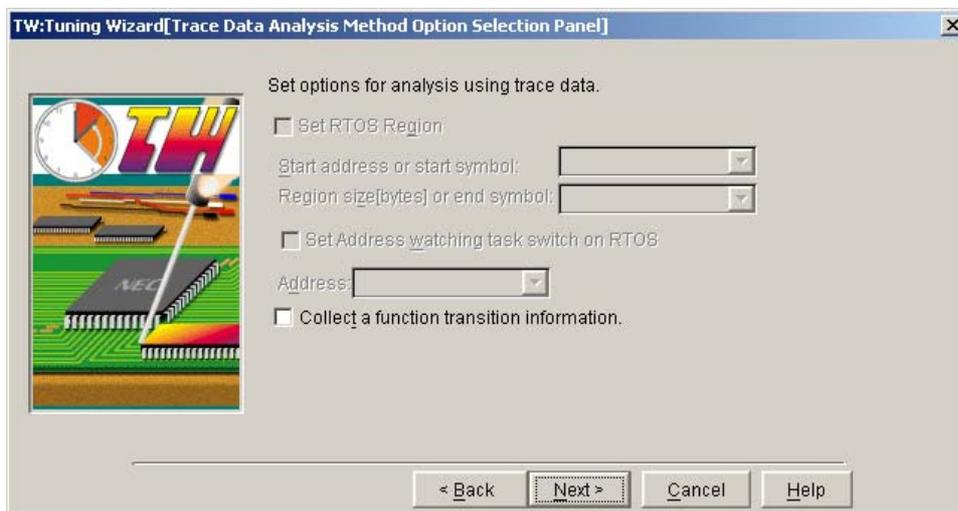
- If the [Analyze an original execution module.] check box is not checked

Caution A probe code is not inserted in a tuned execution module. If this radio button is checked, therefore, information, such as the number of times of execution and execution time of each function, cannot be acquired after tuning.

Trace Data Analysis Method Option Selection Panel

This panel is used to specify an option using the trace data analysis method to acquire performance analysis information for wizard processing (performance analysis) of the TW850.

Figure 6-12 Trace Data Analysis Method Option Selection Panel



This section describes the following items.

- [Explanation of each area](#)

Explanation of each area

This panel consists of the following items.

- [Message display area](#)
- [RTOS region specification area](#)
- [Address specification area for monitoring RTOS task transition](#)
- [\[Collect a function transition information.\] check box](#)

(1) Message display area

This area displays an explanation messages related to this panel.

(2) RTOS region specification area

(a) [Set RTOS Region] check box

This check box is used to specify whether an area that the RTOS system uses is specified.

This check box is checked in the default condition.

If this check box is checked, an area that the RTOS system uses is specified.

Remark This check box is invalid if the following condition is satisfied.

- If the RTOS of the V850 series or "no use" in the [RTOS] combo box on the [Initial Screen Panel](#) is selected

(b) [Start address or start symbol] combo box

This combo box is used to specify the start address or the start symbol of an area that RTOS system uses.

Remark This combo box is invalid if the following condition is satisfied.

- If an RTOS of the V850 series or "no use" in the [RTOS] combo box on the [Initial Screen Panel](#) is selected
- [If the [Set RTOS Region] check box is not checked

(c) [Region size[bytes] or end symbol] combo box

This combo box is used to specify the size or the end symbol of a RTOS region.

Remark This combo box is invalid if the following condition is satisfied.

- If an RTOS of the V850 series or "no use" in the [RTOS] combo box on the [Initial Screen Panel](#) is selected
- [If the [Set RTOS Region] check box is not checked

(3) Address specification area for monitoring RTOS task transition

(a) [Set Address watching task switch on RTOS] check box

This check box is used to specify whether an address is specified to monitor the task transition of the RTOS. This check box is checked in the default condition.

If this check box is checked, an address is specified to monitor the task transition of the RTOS.

Remark This check box is invalid if the following conditions are satisfied.

- If an RTOS of the V850 series or "no use" in the [RTOS] combo box on the [Initial Screen Panel](#) is selected
- If BUS trace data cannot be acquired and only N-Wire trace data can be acquired

(b) [Address] combo box

This combo box is used to specify the start address or the start symbol of a RTOS region.

Remark This combo box is invalid if the following condition is satisfied.

- If an RTOS of the V850 series or "no use" in the [RTOS] combo box on the [Initial Screen Panel](#) is selected
- If the [Set Address watching task switch on RTOS] check box is not checked
- If a cacheable memory area and an area used for the user program are specified

Caution The user must monitor task transition and describe a processing code that writes transition source task ID information and transition destination task ID information to a specified address, in the program. Describe the transition source task ID in a 4-byte area starting from the specified address, and the processing code that writes the transition destination task ID to a 4-byte area that starts from the specified address + 4 bytes.

(4) [Collect a function transition information.] check box

This check box is used to specify whether the function transition information is acquired when the performance of the trace data is analyzed.

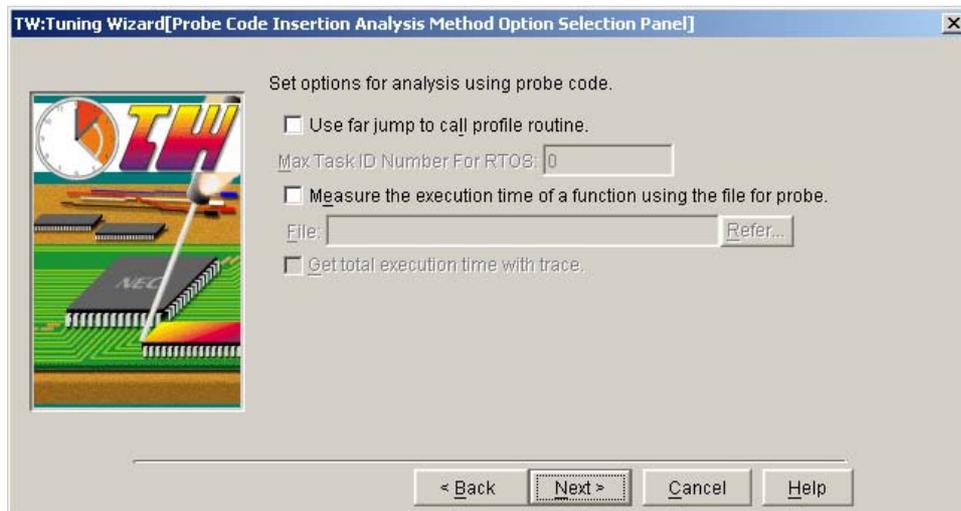
This check box is not checked in the default condition.

If this check box is checked, the function transition information is acquired.

Probe Code Insertion Analysis Method Option Selection Panel

This panel is used to specify an option using the probe code insertion method to acquire performance analysis information for wizard processing (performance analysis) of the TW850.

Figure 6-13 Probe Code Insertion Analysis Method Option Selection Panel



This section describes the following items.

- [Explanation of each area](#)

Explanation of each area

This panel consists of the following items.

- [Message display area](#)
- [\[Use far jump to call profile routine.\] check box](#)
- [\[Max Task ID Number For RTOS\] specification area](#)
- [Performance analysis function execution time measurement processing specification area](#)
- [\[Get total execution time with trace.\] check box](#)

(1) [Message display area](#)

This area displays an explanation messages related to this panel.

(2) [\[Use far jump to call profile routine.\] check box](#)

This check box is used to specify whether the far jump instruction is used to call a profile routine output as an intermediate file in wizard processing (tuning) of the TW850.

This check box is not checked in the default condition.

If this check box is checked, the far jump instruction is used to call a profile routine.

The probe code to be inserted in the probe code insertion method may call a profile routine. If probe code and profile routine are located far away from each other, a compilation error may occur. If the calling that can specify the entire address space (far jump) is used to call the profile routine in this case, a profile routine located far away can be called.

Remarks 1. This check box is invalid if the following condition is satisfied.

- If the [\[Use probe code for analyzing \(for original execution module only\).\]](#) radio button on the [Execution Item Selection Panel](#) is not checked
- 2. If the far jump instruction is used to call a profile routine, the size of the probe code inserted increases.
- 3. If the profile routine may not be allocated properly, allocate it in the profile routine specification area on the [Allocation Destination Selection Panel](#) so that it is close to the location where the program is allocated.

(3) [\[Max Task ID Number For RTOS\] specification area](#)

This area is used to specify the maximum task ID number for profile required by the TW850 to execute wizard processing (performance analysis processing; acquiring RTOS information).

The maximum task ID number for profile means the maximum value of the task ID number for which processing to acquire RTOS information must be executed. The transition of the task is monitored to the specified task ID number.

The range of values that can be specified is from 1 to 127 if the RTOS used is the RX850, and from 1 to 32767 if the RTOS used is the RX850 Pro.

In the default condition, the maximum task ID is specified if RX850 or RX850 Pro is used and the maximum task ID can be acquired from an execution module. 127 is specified if the maximum task ID cannot be acquired.

Remark This specification area is invalid if the following condition is satisfied.

- If the [\[Use probe code for analyzing \(for original execution module only\).\]](#) radio button on the [Execution Item Selection Panel](#) is not checked
- If RTOS is not used, or if "no use" in the [\[RTOS\]](#) combo box on the [Initial Screen Panel](#) is not specified

Caution If a task whose task ID is larger than that of the specified task changes, the change is not monitored. Therefore, the performance of the changed task cannot be analyzed correctly. Therefore, the performance of the task cannot be correctly analyzed. However, a large numeric value beyond necessity is specified, the memory area required on the target increase.

(4) Performance analysis function execution time measurement processing specification area

This area is used to make settings related to execution time measurement processing of functions subject to performance analysis.

(a) [Measure the execution time of a function using the file for probe.] check box

This check box is used to specify whether the execution time for each function subject to performance analysis is measured as wizard processing (performance analysis processing: execution time measurement of functions subject to performance analysis) of the TW850.

To measure the execution time of a performance analysis function, a file for measuring time (timer control code block) must be prepared. Refer to "[4.4 Timer Control Code Block](#)" for details of a file for measuring time (timer control code block).

This check box is not checked in the default condition.

If this check box is checked, execution time of functions subject to performance analysis is measured.

(b) [File] specification area

This area is used to specify the name of a file for measuring time (timer control code block), required by the TW850 to execute wizard processing (performance analysis processing: execution time measurement of functions subject to performance analysis).

Nothing will be specified in the default condition.

Refer to "[4.4 Timer Control Code Block](#)" for details of a file for measuring time (timer control code block).

Remark This specification area is invalid if the following condition is satisfied.

- If the [Measure the execution time of a function using the file for probe.] check box is not checked

(c) <Refer...> button

This button is used to open the [File Selection Dialog Box](#).

Remark This button is invalid if the following condition is satisfied.

- If the [Measure the execution time of a function using the file for probe.] check box is not checked

(5) [Get total execution time with trace.] check box

This check box is used to specify whether the total execution time is acquired by setting trace to on if the total execution time can be acquired only from trace data.

If this check box is checked, the total execution time is acquired by setting trace to on for acquiring the total execution time.

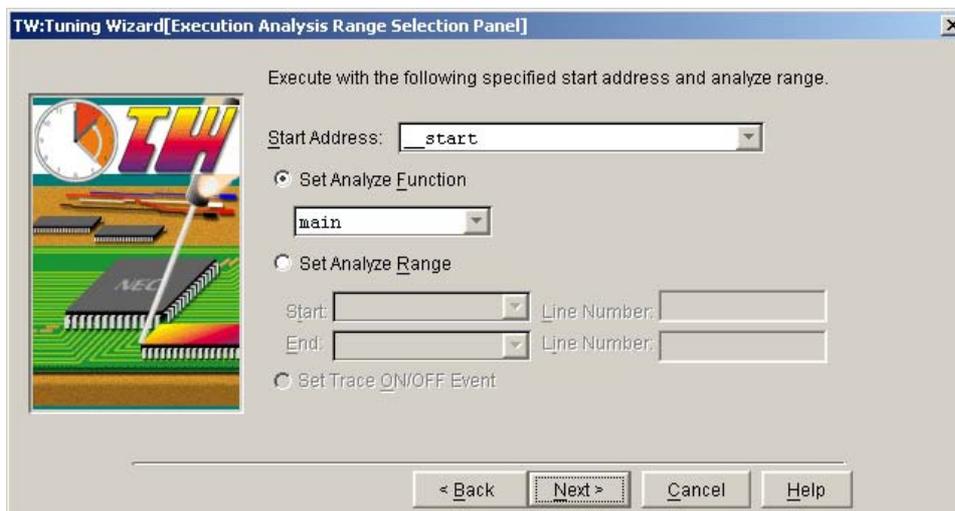
Remark This check box is invalid if the following condition is satisfied.

- If the NEC Electronics tool chain is used

Execution Analysis Range Selection Panel

This panel is used to specify an execution start position of a user program or an execution module and range of performance analysis for wizard processing (performance analysis).

Figure 6-14 Execution Analysis Range Selection Panel



This section describes the following items.

- [Explanation of each area](#)

Explanation of each area

This panel consists of the following items.

- [Message display area](#)
- [\[Start Address\] combo box](#)
- [Performance analysis target function specification area](#)
- [Performance analysis target range specification area](#)
- [\[Set Trace ON/OFF Event\] radio button](#)

(1) Message display area

This area displays an explanation messages related to this panel.

(2) [Start Address] combo box

This combo box is used to specify the execution start position of a user program or an execution module. The address of an entry point, and all the symbols acquired by analyzing the original execution module information can be selected.

Text can also be input. The range of values that can be input is from 0 to maximum address of the chip specified by the project.

The address of an entry point is specified in the default condition. If a symbol of the same value as the address of the entry point exists, the symbol is specified.

Execution of the user program and execution module is started from the position specified in this combo box, without depending on the range of performance analysis. Usually, therefore, specify a reset address.

Remark This combo box is invalid if the following condition is satisfied.

- If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked

(3) Performance analysis target function specification area

This area is used to specify functions subject to performance analysis for wizard processing (performance analysis).

There are the following features depend on the method to acquire performance analysis information.

[With trace data analysis method]

The program is stopped at the specified execution start position and end position, and within the range of performance analysis each time the trace buffer becomes full.

[With probe code insertion method]

The program is stopped at the specified execution start position and end position.

If the execution time is measured on the target using the ID850 and ID850NW, however, the execution time of a function that includes the specified execution start position and end position cannot be measured.

This specification area consists of the following radio button and check box.

(a) [Set Analyze Function] radio button

This radio button is used to specify whether the range subject to performance analysis for wizard processing (performance analysis) is specified by a function name.

This check box is checked in the default condition.

If this radio button is checked, the range subject to performance analysis is specified by a function name.

Remark This radio button is invalid if the following condition is satisfied.

- If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked

(b) Performance analysis target function name specification combo box

This combo box is used to select the range (function name) subject to performance analysis for wizard processing (performance analysis).

The function names acquired by analyzing the original execution module information can be selected.

Nothing will be selected in the default condition.

- Remarks
1. If two or more functions of the same name exist, "function name (file name)" is displayed.
If two or more functions of the same function name or file name exist, "function name (absolute path name)" is displayed.
 2. If the task name is selected, performance analysis is performed from starting the task to ending the task.
 3. This combo box is invalid if the following condition is satisfied.
 - If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked
 - If the [Set Analyze Function] radio button is not checked

Cautions

1. The TW850 performs processing assuming that the return address of a function is set to register LP. Therefore, if a function whose return address is not set to register LP is specified in this combo box, the operation is not guaranteed.

2. Specify the function that the processing is to be terminated. If a function that the processing is not terminated is specified, the performance analysis processing will not end.

(4) Performance analysis target range specification area

This area is used to specify the range subject to performance analysis for wizard processing (performance analysis).

There are the following features depend on the method to acquire performance analysis information.

[With trace data analysis method]

The program is stopped at the specified execution start position and end position, and within the range of performance analysis each time the trace buffer becomes full.

[With probe code insertion method]

The program is stopped at the specified execution start position and end position.

If the execution time is measured on the target using the ID850 and ID850NW, however, the execution time of a function that includes the specified execution start position and end position cannot be measured.

This specification area consists of the following radio button, combo boxes, and specification areas.

(a) [Set Analyze Range] radio button

This radio button is used to specify whether the range subject to performance analysis for wizard processing (performance analysis) is specified by a file name, symbol name, or line number.

This check box is not checked in the default condition.

If this radio button is checked, the range subject to performance analysis is specified by a file name, symbol name, or line number.

Remark This radio button is invalid if the following condition is satisfied.

- If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked

(b) [Start] combo box

This combo box is used to select the performance analysis start position (file name or symbol name) for wizard processing (performance analysis).

The name of C source files and all the symbols in the text-attribute area acquired by analyzing the original execution module information can be selected.

Nothing will be selected in the default condition.

Remarks 1. This combo box is invalid if either of the following conditions is satisfied.

- If the [Set Analyze Range] radio button is not checked
 - If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked
2. If two or more functions of the same name exist, "function name (file name)" is displayed.
If two or more functions of the same function name or file name exist, "function name (absolute path name)" is displayed.

Cautions 1. To display file names as selection items in this combo box, an original execution module must be generated by using a debug option (-g with the NEC Electronics compiler).

2. Specify the name of a file or a symbol that is always executed in this combo box.

(c) [Line Number] specification area

This area is used to specify the performance analysis start position (line number) for wizard processing (performance analysis).

The range of values that can be input is from 1 to 4,294,967,295.

Nothing will be specified in the default condition.

Remarks 1. This specification area is invalid if any of the following conditions is satisfied.

- If nothing is input to the [Start] combo box
 - If a file other than a source file is specified in the [Start] combo box
 - If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked
2. If there is a possibility that the specified line number indicates a position that is not expected by the user by optimization of the compiler, the line number is displayed in red.

Caution Specify the number of a line that a breakpoint can be set and executed on the debugger.

(d) [End] combo box

This combo box is used to select the performance analysis end position (file name or symbol name) for wizard processing (performance analysis).

The name of C source files and all the symbols in the text-attribute area acquired by analyzing the original execution module information can be selected.

Nothing will be selected in the default condition.

Remarks 1. This combo box is invalid if either of the following conditions is satisfied.

- If the [Set Analyze Range] radio button is not checked
 - If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked
2. If two or more functions of the same name exist, "function name (file name)" is displayed.
If two or more functions of the same function name or file name exist, "function name (absolute path name)" is displayed.

Cautions 1. To display file names as selection items in this combo box, an original execution module must be generated by using a debug option (-g with the NEC Electronics compiler).

2. Specify the name of a file or a symbol that is always executed in this combo box.

(e) [Line Number] specification area

This area is used to specify the performance analysis end position (line number) for wizard processing (performance analysis).

The range of values that can be input is from 1 to 4,294,967,295.

Nothing will be specified in the default condition.

Remarks 1. This specification area is invalid if any of the following conditions is satisfied.

- If nothing is input to the [End] combo box
 - If a file other than a source file is specified in the [End] combo box
 - If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked
2. If there is a possibility that the specified line number indicates a position that is not expected by the user by optimization of the compiler, the line number is displayed in red.

Caution Specify the number of a line that a breakpoint can be set and executed on the debugger.

(5) [Set Trace ON/OFF Event] radio button

This radio button is used to specify whether writing data in the program is used as an event (trace event) to set tracing on or off (to start or end performance analysis) when the trace data analysis method is used.

This check box is not checked in the default condition.

If this radio button is checked, writing data in the program is used as a trace event when the trace data analysis method is used.

This radio button has the following features, compared with the [Set Analyze Function] and [Set Analyze Range] radio buttons.

- The program stops at the position where a trace event has occurred (it also stops at the performance analysis end position if a performance analysis end position is specified), but does not stop even when the trace buffer has become full.
- When the trace buffer has become full, only trace data of the trace buffer capacity is acquired.
- Data that controls the trace event must be written in the program.
- The performance analysis by occurring a trace event can be performed from the start position to the end position more than once and can be totaled.
- The trace data the user intended to acquire can be acquired because starting or ending performance analysis can be controlled in the user program.

Remark This radio button is invalid if the following condition is satisfied.

- If the ID850NW is not used as the debugger or if an access break event cannot be set for the debugger
- If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked

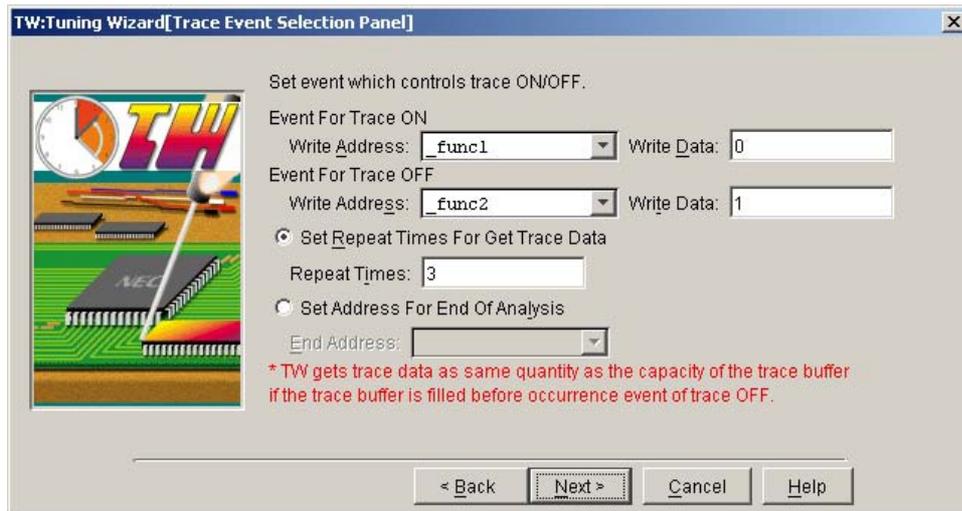
Trace Event Selection Panel

This panel is used to select a trace event in range of performance analysis for wizard processing (performance analysis) of the TW850.

Caution This panel is not opened if the following condition is satisfied.

- If the ID850NW is not used as the debugger

Figure 6-15 Trace Event Selection Panel



This section describes the following items.

- [Explanation of each area](#)

Explanation of each area

This panel consists of the following items.

- [Message display area](#)
- [Trace on event specification area](#)
- [Trace off event specification area](#)
- [Number of times of trace data acquisition specified area](#)
- [Analysis end position specification area](#)

(1) Message display area

This area displays an explanation messages related to this panel.

(2) Trace on event specification area

This area is used to specify an event to set trace to on (to start performance analysis). This specification area consists of the following combo box and specification area.

(a) [Write Address] combo box

This combo box is used to specify the write destination symbol name or address that is used as an event that sets trace to on.

The names of all the global symbols acquired by analyzing the original execution module information can be selected.

Text can also be input. The range of values that can be input is from 0x0 to maximum address of the chip specified by the project.

Nothing will be specified in the default condition.

Remarks 1. This radio button is invalid if the following condition is satisfied.

- If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked
2. If there is a possibility that the write address is changed by tuning processing, specify a symbol name.
 3. With this combo box, the same position as the [Write Address] combo box in the trace off event specification area can be specified.

(b) [Write Data] specification area

This area is used to specify the 4-byte write data that is used as an event that sets trace to on.

A 4-byte non-negative integer can be input in a range of 0 to 4294967295 (0xffffffff).

Remarks 1. This radio button is invalid if the following condition is satisfied.

- If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked
2. With this combo box, the same position as the [Write Data] specification area in the trace off event specification area can be specified.

Cautions 1. The TW850 stops execution (CPU) at the location that an event has occurred and resumes execution after setting trace to on, if it detects a trace on event with trace off (if it detects as an access break event that specified write data is written to a specified write position) while an execution module is being executed. This trace on event must be described in the user program. The TW850 does nothing if it detects a trace on event with trace set to on.

2. If a target that can acquire N-Wire trace data is used, only the following address ranges can be specified.

64MB mode: 0x0 to 0x3ffffff

256MB mode: 0x0 to 0xffffffff

3. Depending on the target, detection of an access break event may lag behind the actual timing of writing data. If a trace on event is set with such a target, either the delay should be recognized as an error or a countermeasure must be taken. For whether the timing of occurrence of an access break event is delayed and for the countermeasure, refer to the user's manual of the target.

(3) Trace off event specification area

This area is used to specify an event to set trace to off (to end performance analysis).

This specification area consists of the following combo box and specification area.

(a) [Write Address] combo box

This combo box is used to specify the write destination symbol name or address that is used as an event that sets trace to off.

The names of all the global symbols acquired by analyzing the original execution module information can be selected.

Text can also be input. The range of values that can be input is from 0x0 to maximum address of the chip specified by the project.

Nothing will be specified in the default condition.

Remarks 1. This radio button is invalid if the following condition is satisfied.

- If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked
2. If there is a possibility that the write address is changed by tuning processing, specify a symbol name.
 3. With this combo box, the same position as the [Write Address] combo box in the trace on event specification area can be specified.

(b) [Write Data] specification area

This area is used to specify the 4-byte write data that is used as an event that sets trace to on.

A 4-byte non-negative integer can be input in a range of 0 to 4294967295 (0xffffffff).

Remarks 1. This radio button is invalid if the following condition is satisfied.

- If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked
2. With this combo box, the same position as the [Write Data] specification area in the trace on event specification area can be specified.
 3. The TW850 stops execution (CPU) at the location that an event has occurred and resumes execution after setting trace to off, if it detects a trace off event with trace on (if it detects as an access break event that specified write data is written to a specified write position) while an execution module is being executed. This trace off event must be described in the user program. The TW850 does nothing if it detects a trace off event with trace set to off.
 4. If a target that can acquire N-Wire trace data is used, only the following address ranges can be specified.
 - 64MB mode: 0x0 to 0x3ffffff
 - 256MB mode: 0x0 to 0xffffffff
 5. Depending on the target, detection of an access break event may lag behind the actual timing of writing data. If a trace on event is set with such a target, either the delay should be recognized as an error or a countermeasure must be taken. For whether the timing of occurrence of an access break event is delayed and for the countermeasure, refer to the user's manual of the target.

(4) Number of times of trace data acquisition specified area

This area is used to make settings related to the number of times of trace data is to be acquired in wizard processing (performance analysis).

Performance analysis (execution of the program) ends when the trace data has been acquired the specified number of times.

This specification area consists of the following radio button and specification area.

(a) [Set Repeat Times For Get Trace Data] radio button

This button is used to specify whether analysis is executed with the number of times the trace data is to be acquired specified.

This check box is checked in the default condition.

If this radio button is checked, analysis is executed with the number of times the trace data is to be acquired specified.

Remark This radio button is invalid if the following condition is satisfied.

- If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked

(b) [Repeat Times] specification area

This area is used to specify the number of times trace data is to be acquired.
The range of values that can be input is from 0 to 2,147,483,647.

Caution If a number of times greater than the number of times the trace data can be acquired (the number of times a trace off event occurs after occurrence of a trace on event) is specified, the performance analysis processing will not end. Do not specify a number of times exceeding the number of times a trace off event occurs.

Remark This specification area is invalid if the following condition is satisfied.

- If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked
- If the [Set Repeat Times For Get Trace Data] radio button is not checked

(5) Analysis end position specification area

This area is used to make settings related to the analysis end position in wizard processing (performance analysis). This specification area consists of the following radio button and combo box.

(a) [Set Address For End Of Analysis] radio button

This button is used to specify whether analysis is executed with the end position specified.
This check box is not checked in the default condition.
If this radio button is checked, the analysis end position is specified.

Remark This radio button is invalid if the following condition is satisfied.

- If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked

(b) [End Address] combo box

This combo box is used to specify the symbol name or address at the analysis end position.
The names of all the symbols acquired by analyzing the original execution module information can be selected.
Text can also be input. The range of values that can be input is from 0x0 to maximum address of the chip specified by the project.
Nothing will be specified in the default condition.

Remarks 1. If there is a possibility that the write address is changed by tuning processing, specify a symbol name.

2. This radio button is invalid if the following condition is satisfied.

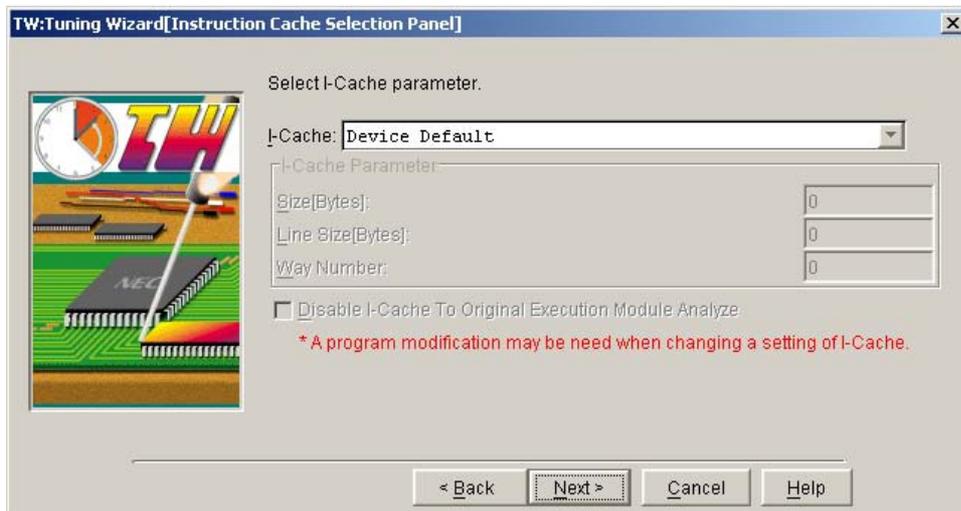
- If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked
- If the [Set Address For End Of Analysis] radio button is not checked

Caution Specify the name of a file or a symbol that is always executed in this combo box.

Instruction Cache Selection Panel

This panel is used to specify a type and parameters of an instruction cache used for wizard processing (performance analysis or tuning) of the TW850.

Figure 6-16 Instruction Cache Selection Panel



This section describes the following items.

- [Explanation of each area](#)

Explanation of each area

This panel consists of the following items.

- [Message display area](#)
- [\[I-Cache\] combo box](#)
- [Instruction cache parameter specification area](#)
- [\[Disable I-Cache To Original Execution Module Analyze\] check box](#)

(1) Message display area

This area displays an explanation messages related to this panel.

(2) [I-Cache] combo box

This combo box is used to select the type of the instruction cache used for wizard processing (performance analysis or tuning).

The selectable items are "Device Default", "None"(if an execution engine that can specify an instruction cache size from combinations of cache macros is used), "NB85E212(4 Kbyte, 2 Way Set Associative)"(V850 Series), "NB85E213(8 Kbyte, 2 Way Set Associative)"(V850 Series), "Custom"(if an execution engine that can specify an instruction cache size from other than combinations of cache macros is used).

"Device Default" is specified in the default condition.

Caution If the instruction cache is usable, the TW850 set the instruction cache selected in this combo box to the execution engine, but it does not execute CPU initialization processing to use the instruction cache (setting of the instruction cache as described in the User's Manual "Instruction Cache, Data Cache NB85E, NB85ET"). To use the instruction cache, therefore, processing to set the instruction cache must be added to the startup routine by the CPU.

The instruction cache does not have to be set to the startup routine for an NEC Electronics simulator.

(3) Instruction cache parameter specification area

This area is used to specify the parameters of the instruction cache used for wizard processing (performance analysis or tuning).

This specification area consists of the following specification areas.

(a) [Size[Bytes]] specification area

This area is used to specify the size (in bytes) of the instruction cache used for wizard processing (performance analysis or tuning).

The value that can be specified in this area is limited to an integer multiple of the result of calculating "Value specified in [Line Size] specification area" * "Value specified in [Way Number] specification area".

The size of the instruction cache of the chip specified by the project is specified in the default condition.

Remark This specification area is invalid if the following condition is satisfied.

- If an item other than "Custom" is selected in the [I-Cache] combo box

Caution If the size of the instruction cache is set to 0, the value of the ICC register is not 0x0. As a result, completion of initialization of a tag may not be detected. In this case, wait until the lower 2 bits of the ICC register are cleared to 0 to detect completion of initialization of a tag.

This description is irrelevant for an NEC Electronics simulator.

(b) [Line Size[Bytes]] specification area

This area is used to specify the line size (in bytes) of the instruction cache used for wizard processing (performance analysis or tuning).

The range of values that can be input differs depending on the execution engine.

The line size of the instruction cache of the chip specified by the project is specified in the default condition.

Remark This specification area is invalid if the following condition is satisfied.

- If an item other than "Custom" is selected in the [I-Cache] combo box

(c) [Way Number] specification area

This area is used to specify the number of ways of the instruction cache used for wizard processing (performance analysis or tuning).

The range of values that can be input differs depending on the execution engine.

The number of ways of the chip specified by the project is specified in the default condition.

Remark This specification area is invalid if the following condition is satisfied.

- If an item other than "Custom" is selected in the [I-Cache] combo box

Caution If the number of ways of the instruction cache is set to 1 or less, the value of the ICC register is not 0x0. As a result, completion of initialization of a tag may not be detected. In this case, wait until the lower 2 bits of the ICC register are cleared to 0 to detect completion of initialization of a tag.
This description is irrelevant for an NEC Electronics simulator.

(4) [Disable I-Cache To Original Execution Module Analyze] check box

This check box is used to specify whether the instruction cache is made invalid when the performance of the original execution module is analyzed.

This check box is not checked in the default condition.

If this check box is checked, the instruction cache is made invalid when the performance of the original execution module is analyzed.

Remark This check box is invalid if any of the following conditions is satisfied.

- If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked
- If the size of the instruction cache is 0
- If a device that the cache cannot be invalidated is specified

Fast Access Memory Area Selection Panel

This panel is used to specify a type and parameters of a fast access memory area used for wizard processing (tuning) of the TW850.

Figure 6-17 Fast Access Memory Area Selection Panel



This section describes the following items.

- [Explanation of each area](#)

Explanation of each area

This panel consists of the following items.

- Message display area
- [Use Fast Accessible Memory] check box
- [Fast Accessible Memory] combo box
- Fast access memory area parameter specification area

(1) Message display area

This area displays an explanation messages related to this panel.

(2) [Use East Accessible Memory] check box

This check box is used to specify whether a function that causes the execution performance of the execution module to drop is given priority and allocated to the fast access memory area (such as internal ROM) in wizard processing (tuning).

This check box is checked in default condition if the chip specified by the project has an internal ROM.

If this check box is checked, a function that causes the execution performance to drop is given priority and allocated to the fast access memory area (such as internal ROM).

(3) [Fast Accessible Memory] combo box

This combo box is used to select the type of the fast access memory area used for wizard processing (tuning).

The selectable items are "Internal ROM" (in the case of the V850 Series) and "Custom".

In default condition, "Internal ROM" is specified if the chip has an internal ROM. "Custom" is specified if the chip does not have an internal ROM.

Remark This combo box is invalid if the following condition is satisfied.

- If the [Use East Accessible Memory] check box is not checked

(4) Fast access memory area parameter specification area

This area is used to specify the parameters of the fast access memory area used for wizard processing (tuning).

This specification area consists of the following specification areas.

(a) [Start Address] specification area

This area is used to specify the start address of the fast access memory area used for wizard processing (tuning).

The range of values that can be input is from 0x0 to maximum address of the chip specified by the project.

In the default condition, the start address of the chip specified by the project is specified if the chip has an internal ROM. 0x0 is specified if the chip does not have an internal ROM.

Remark This specification area is invalid if the following condition is satisfied.

- If the [Use East Accessible Memory] check box is not checked

(b) [Size[Bytes]] specification area

This area is used to specify the size (in Kbytes) of the fast access memory area used for wizard processing (tuning).

The range of values that can be input depends on the execution engine if "Internal ROM" in the [Fast Accessible Memory] combo box is specified, and the range is from 1 to the maximum value of the address space specified by the project if "Custom" is specified.

In the default condition, the size of the chip specified by the project is specified if the chip has an internal ROM. 0 is specified if the chip does not have an internal ROM.

Remark This specification area is invalid if the following condition is satisfied.

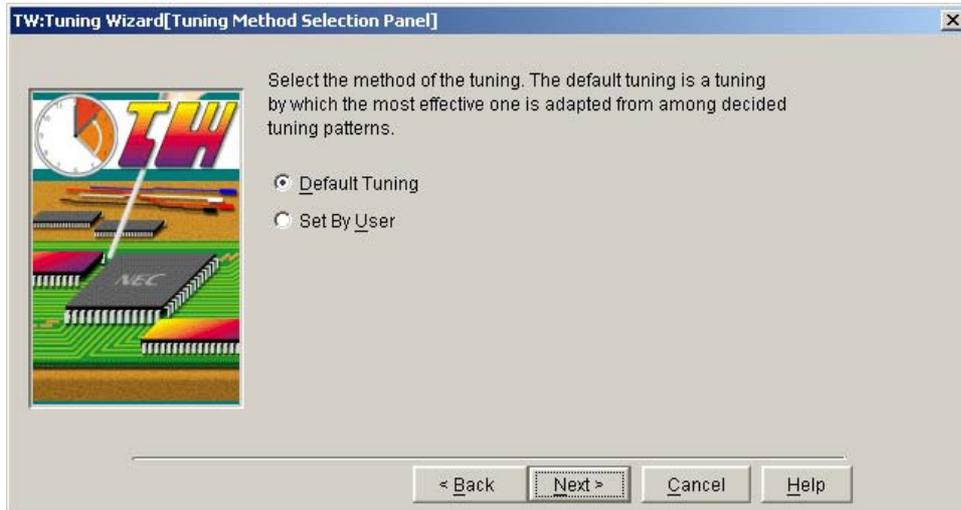
- If the [Use East Accessible Memory] check box is not checked
- If the ID850 or ID850NW that is specified internal ROM is used

Tuning Method Selection Panel

This panel is used to specify a method of tuning a user program or an execution module (standard pattern or user-specified pattern).

Refer to "4.2 [Tuning Functions](#)" for details of tuning functions.

Figure 6-18 Tuning Method Selection Panel



This section describes the following items.

- [Explanation of each area](#)

Explanation of each area

This panel consists of the following items.

- [Message display area](#)
- [Tuning method specification area](#)

(1) Message display area

This area displays an explanation messages related to this panel.

(2) Tuning method specification area

This area is used to make settings related to the tuning method of a user program and an execution module. This specification area consists of the following radio buttons.

(a) [Default Tuning] radio button

This radio button is used to specify whether a user program or an execution module is tuned by a standard pattern (tuning pattern recommended by the TW850).

This check box is checked in the default condition.

If this radio button is checked, tuning is performed by a standard pattern (tuning pattern recommended by the TW850).

In the standard pattern, tuning that two or more tuning options are combined is executed more than once, and the result of tuning that has been executed with the combination of tuning options that the speed could be maximized is reported. Two or more types of tuning can be automatically executed without the user having to understand the tuning in detail.

Remark Because tuning and program execution are executed more than once, tuning may take longer.

The combination of tuning options that the TW850 tries are as follows.

[If the instruction cache exists]

If the instruction cache exists, the following patterns 1 to 3 are tried.

Tuning Options	Pattern 1	Pattern 2	Pattern 3
Executing tuning of fast access memory area when use of fast access memory area is specified	Yes	Yes	Yes
Giving priority to a function that takes time to be executed if trace data analysis method is used in tuning of fast access memory area Giving priority to a function that often calls a callpair if the probe code insertion method is used	Yes	Yes	Yes
Tuning of instruction cache	Yes	Yes	Yes
Using coloring algorithm for tuning of instruction cache and giving priority to number of times callpair is called as allocation priority correction option	Yes	No	No
Using coloring algorithm for tuning of instruction cache and giving priority to depth of callpair as allocation priority correction option	No	Yes	No
Using grouping algorithm for tuning of instruction cache and giving priority to number of times callpair is called as allocation priority correction option Giving consideration to number of ways if number of ways is 2 or more	No	No	Yes
No callpair correction option specified	Yes	No	Yes
Giving priority to function corresponding to hot path as callpair correction option if the trace data analysis method is used	No	Yes	No

[If the instruction cache does not exist]

If the instruction cache does not exist, the following patterns 1 to 3 are tried.

Tuning Options	Pattern 1	Pattern 2	Pattern 3
Tuning of fast access memory area	Yes	Yes	Yes
Giving priority to function that often calls callpair in tuning of fast access memory area	Yes	Yes	No
Giving priority to a function that takes time to be executed in tuning of fast access memory area	No	No	Yes
No callpair correction option specified	Yes	No	Yes
Giving priority to function of float runtime library as callpair correction option (if system library for the TW850 is used and float runtime library function is included in user program) and giving priority to callpair in loop processing	No	Yes	No

Remark The pattern 3 is tried only if measurement of execution time of function is specified when trace data analysis method or probe code analysis method is used

[If the total execution time cannot be acquired]

If the total execution time cannot be acquired, the following patterns 1 to 3 are tried.

- Executing tuning of fast access memory area when use of fast access memory area is specified
- Giving priority to a function that takes time to be executed if probe code analysis method is used in tuning of fast access memory area
Giving priority to a function that often calls a callpair if the probe code insertion method is used
- Tuning of instruction cache if instruction cache exists
- Using coloring algorithm for tuning of instruction cache and giving priority to number of times callpair is called as allocation priority correction option
- No callpair correction option specified

(b) [Set By User] radio button

This radio button is used to specify whether a user program or an execution module is tuned by a user-specified pattern.

This check box is not checked in the default condition.

If this radio button is checked, tuning is performed by a user-specified pattern.

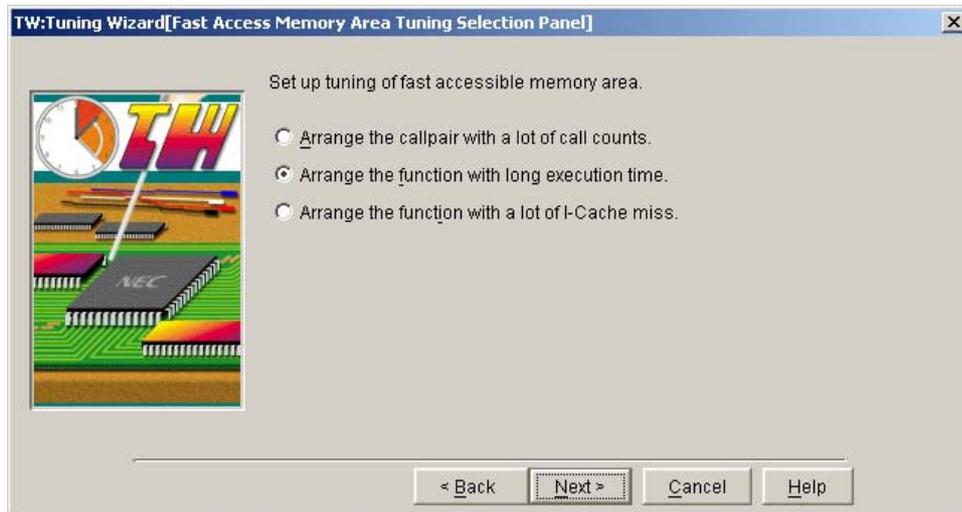
If the optimum tuning option is determined for the standard pattern, the user-specified pattern can execute tuning again, or with various tuning options that are not available for the standard pattern set.

Remark "Settings of each tuning item" that is required when this radio button is checked is performed during wizard processing (such as the [Fast Access Memory Area Tuning Selection Panel](#), [Instruction Cache Tuning Selection Panel](#), [Callpair Correction Option Selection Panel](#)), which is continued by clicking the <Next> button.

Fast Access Memory Area Tuning Selection Panel

This panel is used to specify a method of allocating functions in a fast access memory area (such as internal ROM), based on callpair information that was acquired when a performance of a user program or an execution module was analyzed.

Figure 6-19 Fast Access Memory Area Tuning Selection Panel



This section describes the following items.

- [Explanation of each area](#)

Explanation of each area

This panel consists of the following items.

- [Message display area](#)
- [Function allocation method specification area](#)

(1) Message display area

This area displays an explanation messages related to this panel.

(2) Function allocation method specification area

This area is used to make settings related to the method of allocating functions in the fast access memory area (such as internal ROM), based on the information (such as callpair information and execution time information) that was acquired when the performance of a user program or an execution module was analyzed.

This specification area consists of the following radio buttons.

(a) [Arrange the callpair with a lot of call counts.] radio button

This button is used to specify whether, among functions that have been called many times (calling functions and called functions), a function with small code size is given priority and allocated to the fast access memory area, based on the callpair information (calling relationship between functions) that was acquired when the performance of a user program or an execution module was analyzed.

If the [Arrange the Function with long execution time.] radio button is not checked, this check box is checked in the default condition.

If this radio button is checked, among functions that have been called many times (calling functions and called functions), a function with small code size is given priority and allocated to the fast access memory area.

This radio button is especially effective for tuning if there are many functions that are often called and have small code size, or if an instruction cache exists and the fast access memory is uncacheable.

Remark This radio button is invalid if the following condition is satisfied.

- If the [Use East Accessible Memory] check box on the [Fast Access Memory Area Selection Panel](#) is not checked

(b) [Arrange the function with long execution time.] radio button

This button is used to specify whether a function that takes time to execute one instruction is given priority and allocated to the fast access memory area, based on the execution time information of each function that was acquired when the performance of a user program or an execution module was analyzed.

If this radio button is valid, this radio button is checked in the default condition.

If this button is checked, a function that takes time to execute one instruction is given priority and allocated to the fast access memory area.

This radio button is especially effective for tuning functions with a great load, such as when complicated processing is performed in a function or when part of a function is repeatedly executed in loop processing.

Remarks 1. This radio button is invalid if all the following condition is satisfied.

- If the [Use probe code for analyzing (for original execution module only).] radio button on the [Execution Item Selection Panel](#) is checked
- If the [Measure the execution time of a function using the file for probe.] check box on the [Probe Code Insertion Analysis Method Option Selection Panel](#) is not checked

2. This radio button is invalid if the following condition is satisfied.

- If the [Use East Accessible Memory] check box on the [Fast Access Memory Area Selection Panel](#) is not checked

(c) [Arrange the function with a lot of I-Cache miss.] radio button

This radio button is used to specify whether a function with many instruction cache misses is given priority and allocated to the fast access memory area, based on the instruction cache information (the number of instruction cache hits and misses) of each function that was acquired when the performance of the user program or execution module was analyzed.

This check box is not checked in the default condition.

If this radio button is checked, a function with many instruction cache misses is given priority and allocated to the fast access memory area.

This radio button brings a tuning effect of eliminating the instruction cache misses caused by a function by not only increasing the speed of a function whose execution speed has dropped because of an instruction cache miss but also by allocating a function causing many instruction cache misses in the fast access memory if the fast access memory is an uncacheable area.

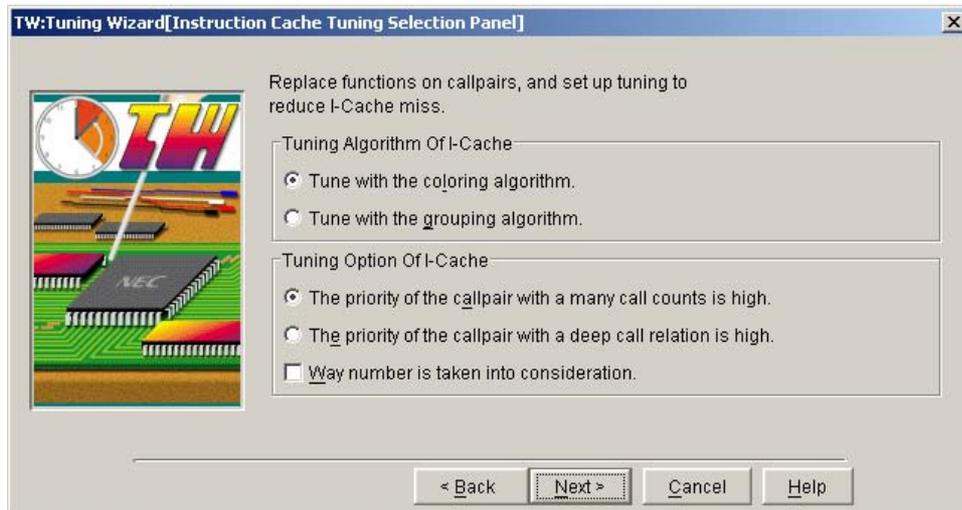
Remark This radio button is invalid if any of the following conditions is satisfied.

- If the [Use trace data for analyze.] radio button on the [Execution Item Selection Panel](#) is not checked
- If the size of the instruction cache is 0
- If the [Disable I-Cache To Original Execution Module Analyze] check box on the [Instruction Cache Selection Panel](#) is checked
- If the [Use East Accessible Memory] check box on the [Fast Access Memory Area Selection Panel](#) is not checked

Instruction Cache Tuning Selection Panel

This panel is used to specify a method of tuning an instruction cache (tuning algorithm and allocation priority correction option), based on callpair information that was acquired when a performance of a user program or an execution module was analyzed.

Figure 6-20 Instruction Cache Tuning Selection Panel



This section describes the following items.

- [Explanation of each area](#)

Explanation of each area

This panel consists of the following items.

- [Message display area](#)
- [Tuning algorithm specification area](#)
- [Allocation priority correction option specification area](#)

(1) Message display area

This area displays an explanation messages related to this panel.

(2) Tuning algorithm specification area

This area is used to make settings related to the tuning algorithm of the instruction cache, based on the callpair information that was acquired when the performance of a user program or an execution module was analyzed. This specification area consists of the following radio buttons.

(a) [Tune with the coloring algorithm.] radio button

This button is used to specify whether functions are allocated to the instruction cache by a coloring algorithm. This check box is checked in the default condition.

If this radio button is checked, functions are allocated to the instruction cache by a coloring algorithm.

Remarks 1. The coloring algorithm allocates functions, based on the callpair information (calling relationship between functions) of each function, so that instruction cache misses among all callpairs can be reduced in as a wide range as possible. This algorithm therefore can reduce instruction cache misses in a wider range but may take time to execute tuning processing. In addition, a hole is inserted between functions and this may create a gap. As a result, the hole size may increase.

2. This radio button is invalid if the following condition is satisfied.

- If the size of the instruction cache is 0

(b) [Tune with the grouping algorithm.] radio button

This button is used to specify whether functions are allocated to the instruction cache by a grouping algorithm. This check box is not checked in the default condition.

If this radio button is checked, functions are allocated to the instruction cache by a grouping algorithm.

Because this algorithm is simpler than the coloring algorithm, the time required for tuning processing can be shortened. This algorithm is effective especially for a program that locally calls many small functions over and over again.

Remarks 1. The grouping algorithm groups functions in a close calling relationship and allocates them so that instruction cache misses are decreased within the group, with the size of the instruction cache as the upper-limit value of the group size, based on the callpair information (calling relationship between functions) of each function. It also allocates functions so that instruction cache misses of functions allocated to different groups are reduced, by changing the sequence of functions in a group.

2. This radio button is invalid if the following condition is satisfied.

- If the size of the instruction cache is 0

(3) Allocation priority correction option specification area

This area is used to make settings related to the allocation priority correction options for the instruction cache, based on the callpair information that was acquired when the performance of a user program or an execution module was analyzed.

This specification area consists of the following radio buttons and check box.

(a) [The priority of the callpair with a many call counts is high.] radio button

This button is used to specify whether a callpair called many times is given priority and allocated to the instruction cache.

This check box is checked in the default condition.

If this radio button is checked, a callpair called many times is given priority and allocated to the instruction cache.

Remarks 1. The items that are set on the [Callpair Correction Option Selection Panel](#) and [Priority Allocation Function Selection Panel](#) affect the number of times of calling callpairs. If settings are made on these panels, check this radio button to maximize the tuning effect.

2. This radio button is invalid if the following condition is satisfied.

- If the size of the instruction cache is 0

(b) [The priority of the callpair with a deep call relation is high.] radio button

This button is used to specify whether a callpair close to the end of a calling relationship is given priority and allocated to the instruction cache.

This check box is not checked in the default condition.

If this radio button is checked, a callpair close to the end of a calling relationship is given priority and allocated to the instruction cache.

This radio button is especially effective for tuning a program that consumes a long execution time at the terminal part of a calling relationship.

Remark This radio button is invalid if the following condition is satisfied.

- If the size of the instruction cache is 0

(c) [Way number is taken into consideration.] check box

This check box is used to specify whether the cache mechanism (the number of ways) is taken into consideration when a function is allocated to the instruction cache.

This check box is not checked in the default condition.

If this check box is checked, the cache mechanism (the number of ways) is taken into consideration.

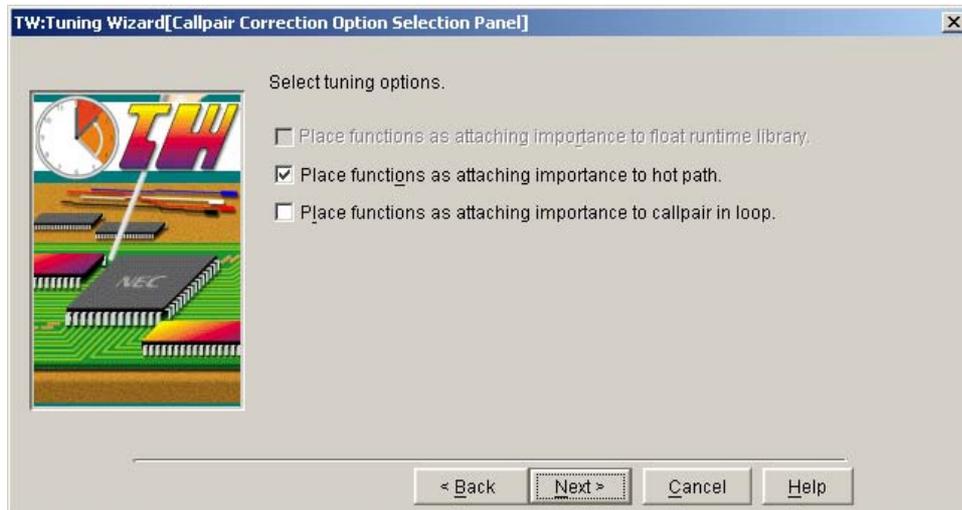
Remark This check box is invalid if either of the following conditions is satisfied.

- If the size of the instruction cache is 0
- If the cache mechanism (the number of ways) of the instruction cache is less than 2

Callpair Correction Option Selection Panel

This panel is used to specify a method of tuning functions subject to be relocated (the callpair priority correction method in tuning of a fast access memory and an instruction cache), based on callpair information that was acquired when a performance of a user program or an execution module was analyzed.

Figure 6-21 Callpair Correction Option Selection Panel



This section describes the following items.

- [Explanation of each area](#)

Explanation of each area

This panel consists of the following items.

- [Message display area](#)
- [Callpair correction option specification area](#)

(1) Message display area

This area displays an explanation messages related to this panel.

(2) Callpair correction option specification area

This panel is used to make settings related to the method of tuning functions subject to be relocated (the callpair priority correction method in tuning of the fast access memory and instruction cache), based on the callpair information that was acquired when the performance of a user program or an execution module was analyzed. This specification area consists of the following check boxes.

(a) [Place functions as attaching importance to float runtime library.] check box

This check box is used to specify whether the float runtime library functions (library functions used by the compiler when a floating-point operation is performed) are given priority and allocated to the instruction cache or fast access memory area.

This check box is not checked in the default condition.

If this check box is checked, functions included in a float runtime library (such as expf and cosf) are given priority and allocated to the instruction cache or fast access memory area.

To speed up a float runtime library function, the function can be allocated and executed at high speed by correcting the number of times a callpair related to the float runtime library function is called.

Remarks 1. This check box is invalid if all the following conditions are satisfied.

- If a radio button other than [Arrange the callpair with a lot of call counts.] is selected on the [Fast Access Memory Area Tuning Selection Panel](#)
- If the size of the instruction cache is 0

2. This check box is invalid if the following condition is satisfied.

- If functions included in a float runtime library (such as expf and cosf) are not included in a user program or an execution module

(b) [Place functions as attaching importance to hot path.] check box

This check box is used to specify whether the functions corresponding to a hot path (function transition path that takes the longest time in the program) are given priority and allocated to the instruction cache or fast access memory area.

This check box is checked in the default condition.

If this check box is checked, a function corresponding to a hot path is given priority and allocated to the instruction cache or fast access memory area.

To speed up a function corresponding to a hot path, the function can be allocated and executed at high speed by correcting the number of times a callpair related to the function, based on information of execution time including the child function of the function (GROSS time).

Remarks 1. This check box is invalid if all the following conditions are satisfied.

- If a radio button other than [Arrange the callpair with a lot of call counts.] is selected on the [Fast Access Memory Area Tuning Selection Panel](#)
- If the size of the instruction cache is 0

2. This check box is invalid if the following condition is satisfied.

- If the [Use probe code for analyzing (for original execution module only).] radio button on the [Execution Item Selection Panel](#) is selected

(c) [Place functions as attaching importance to callpair in loop.] check box

This check box is used to specify whether a callpair function related to the same loop processing is given priority and allocated to the instruction cache or fast access memory area.

This check box is not checked in the default condition.

If this check box is checked, a callpair function related to the same loop processing is given priority and allocated to the instruction cache or fast access memory area.

If functions are called at many places in the loop and the places should be sped up, the function can be allocated and executed the functions in the loop at high speed by correcting the number of times a callpair related to the same loop processing is called.

Remarks 1. If functions are called at many places in the loop, tuning may take time.

2. This check box is invalid if all the following conditions are satisfied.

- If a radio button other than [Arrange the callpair with a lot of call counts.] is selected on the [Fast Access Memory Area Tuning Selection Panel](#)
- If the size of the instruction cache is 0

Remark If two or more check boxes are specified at the same time, the TW850 corrects callpair information in the following order.

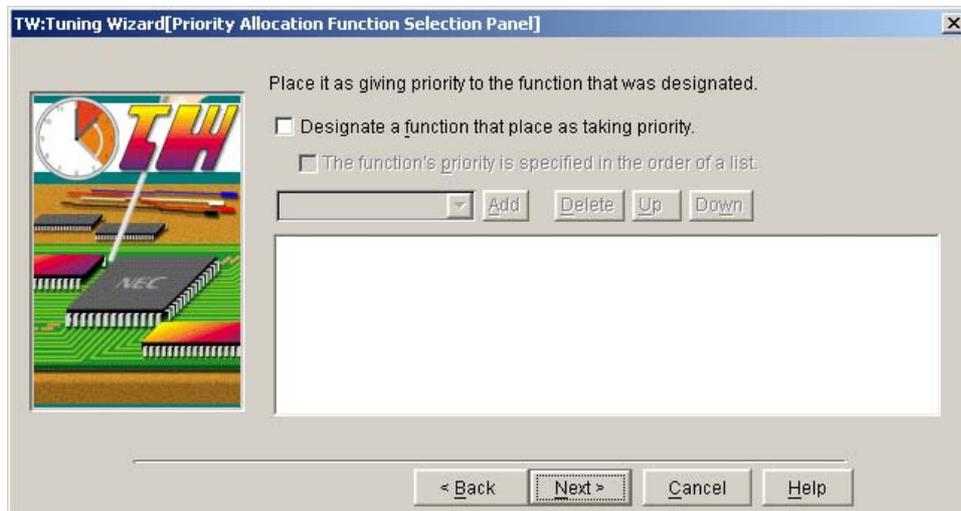
Float runtime library > Function corresponding to a hot path > Function in loop processing

However, specifying all the callpair correction functions does not always bring a good result. If the priority is corrected, the priority of the corrected callpair increases, and the callpair is preferentially allocated. However, the priority of the callpairs that have not been corrected drops.

Priority Allocation Function Selection Panel

This panel is used to specify a function that is given priority and relocated to correct a callpair priority in tuning of a fast access memory and an instruction cache during wizard processing (tuning) of the TW850.

Figure 6-22 Priority Allocation Function Selection Panel



This section describes the following items.

- [Explanation of each area](#)

Explanation of each area

This panel consists of the following items.

- [Message display area](#)
- [\[Designate a function that place as taking priority.\] check box](#)
- [\[The function's priority is specified in the order of a list.\] check box](#)
- [Relocation target function specification area](#)
- [Relocation target function display area](#)

(1) Message display area

This area displays an explanation messages related to this panel.

(2) [Designate a function that place as taking priority.] check box

This panel is used to specify whether a function that is given priority and relocated to correct the callpair priority in tuning of the fast access memory and instruction cache during wizard processing (tuning) is specified.

This check box is not checked in the default condition.

If this check box is checked, a function that is given priority and allocated is specified.

If there is a function that priority should be given to speed, the function can be allocated so that the specified function is executed at high speed.

Remark The specified function is given the top priority and allocated (the function takes precedence over all the callpair correction options) while fast access memory tuning is executed. If the function is not allocated, it is given the top priority and allocated while instruction cache tuning is executed.

Caution Any function can be specified as a function that is given priority and relocated. If many functions are specified, however, the effect of tuning may decrease.

(3) [The function's priority is specified in the order of a list.] check box

This check box is used to specify whether function relocation processing is executed in the display order of the relocation target function display area on this panel (top: highest priority, bottom: lowest priority) during wizard processing (tuning).

This check box is not checked in the default condition.

If this check box is checked, function relocation processing is executed in the display order of the relocation target function display area on this panel (top: highest priority, bottom: lowest priority) during wizard processing (tuning).

If the sequence of functions that should be sped up is specified, the function can be allocated according to the priority of the specified function list.

If this check box is not checked, the priority is determined for fast access memory tuning based on the items specified on the [Fast Access Memory Area Selection Panel](#). For instruction cache tuning, the same priority is given to all the functions.

Remark This check box is invalid if the following condition is satisfied.

- If the [Designate a function that place as taking priority.] check box is not checked

(4) Relocation target function specification area

This area is used to specify a function that is given priority and relocated during wizard processing (tuning).

This specification area consists of the following combo box and buttons.

(a) Relocation target function name specification combo box

This combo box is used to select the function name that is given priority and relocated during wizard processing (tuning).

The function names acquired by analyzing the original execution module information can be selected.

Nothing will be specified in the default condition.

Remarks 1. If two or more functions of the same name exist, "function name (file name)" is displayed.
If two or more functions of the same function name or file name exist, "function name (absolute path name)" is displayed.

2. This combo box is invalid if the following condition is satisfied.

- If the [Designate a function that place as taking priority.] check box is not checked

(b) <Add> button

This button is used to register the function selected in the relocation target function name specification combo box as a function that is given priority and relocated.

By clicking this button, the function name specified in the relocation target function name specification combo box is added to the lowest position in the relocation target function display area.

Remark This button is invalid if the following condition is satisfied.

- If the [Designate a function that place as taking priority.] check box is not checked

(c) <Delete> button

This button is used to cancel registration of the function selected in the relocation target function name specification combo box as a function that is given priority and relocated.

By clicking this button, the function name specified in the relocation target function name specification combo box is deleted from the relocation target function display area.

Remark This button is invalid if the following condition is satisfied.

- If the [Designate a function that place as taking priority.] check box is not checked

(d) <Up> button

This button is used to increase by one level the priority of the function selected in the relocation target function name specification combo box.

By clicking this button, the position of the function name specified in the relocation target function name specification combo box is moved one level up in the relocation target function display area.

Remark This button is invalid if the following condition is satisfied.

- If the [Designate a function that place as taking priority.] check box is not checked

(e) <Down> button

This button is used to decrease by one level the priority of the function selected in the relocation target function name specification combo box.

By clicking this button, the position the function name specified in the relocation target function name specification combo box is moved one level down in the relocation target function display area.

Remark This button is invalid if the following condition is satisfied.

- If the [Designate a function that place as taking priority.] check box is not checked

(5) Relocation target function display area

This area is used to display a list of the function names that is given priority and relocated during wizard processing (tuning).

Nothing will be displayed in the default condition.

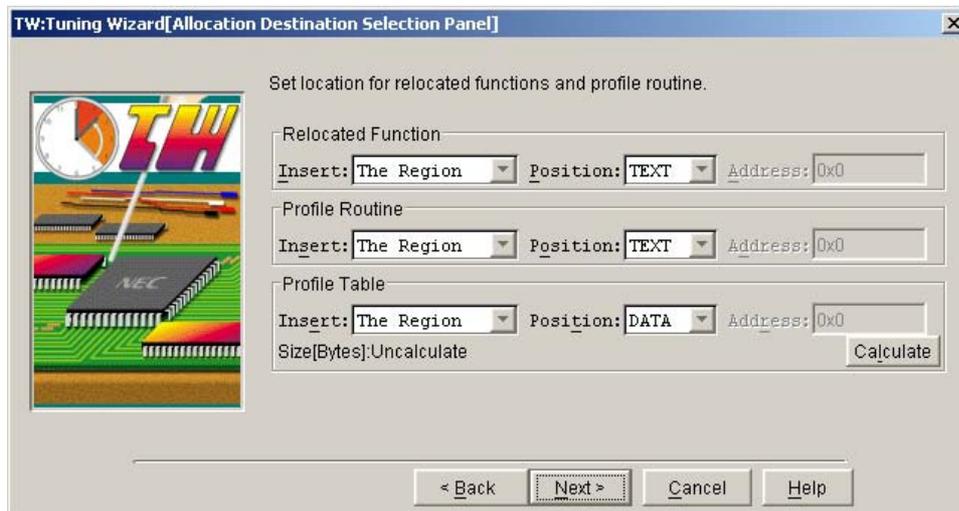
Remark This display area is invalid if the following condition is satisfied.

- If the [Designate a function that place as taking priority.] check box is not checked

Allocation Destination Selection Panel

This panel is used to specify a memory area in which to relocate a function that is subject to relocation during tuning, and a memory area where a profile routine or a profile table output as intermediate files during tuning are to be allocated.

Figure 6-23 Allocation Destination Selection Panel



This section describes the following items.

- [Explanation of each area](#)

Explanation of each area

This panel consists of the following items.

- [Message display area](#)
- [Relocation target function specification area](#)
- [Profile routine specification area](#)
- [Profile table specification area](#)

(1) Message display area

This area displays an explanation messages related to this panel.

(2) Relocation target function specification area

This area is used to make settings related to the memory area for relocating the function to be relocated during tuning.

The function allocated to the fast access memory area is exclude it from relocating. This specification area consists of the following combo boxes and specification area.

(a) [Insert] combo box

This combo box is used to select the insertion position of the memory area for relocating the function subject to relocation during tuning.

The selectable items are "The Region", "In Post-region", "In Pre-region", and "Set Address". "The Region" is selected in the default condition.

Remarks 1. This combo box is invalid if the following condition is satisfied.

- If the [Tune and analyze a tuned execution module.] check box on the [Execution Item Selection Panel](#) is not checked

2. The items other than the "The Region" in this combo box are invalid if the following condition is satisfied.

- If the TWTEXT segment is defined for the target link directive file

(b) [Position] combo box

This combo box is used to select the insertion target area (segment name) of the memory area for relocating the function subject to relocation during tuning.

All text attribute segments specified in the link directive file and all sections specified in the link directive file can be selected.

In the default condition, "TWTEXT" is selected if "TWTEXT" exists, and "TEXT" is selected next if "TEXT" exists.

Usually specify "TEXT segment region" or "behind of .text section".

Remark This combo box is invalid if the following condition is satisfied.

- If the [Tune and analyze a tuned execution module.] check box on the [Execution Item Selection Panel](#) is not checked
- If the TWTEXT segment is defined for the target link directive file

(c) [Address] specification area

This area is used to specify the insertion target area (address) of the memory area for relocating the function subject to relocation during tuning.

The range of values that can be input is from 0x0 to maximum address of the chip specified by the project.

Nothing will be specified in the default condition.

Caution Only multiples of 4 can be specified in this specification area.

Remark This specification area is invalid if the following condition is satisfied.

- If the [Tune and analyze a tuned execution module.] check box on the [Execution Item Selection Panel](#) is not checked
- If an item other than "Set Address" is selected in the [Insert] combo box

(3) Profile routine specification area

This area is used to make settings related to the memory area for relocating the profile routine output as an intermediate file during performance analysis or tuning.

Caution If the profile routine is allocated to a memory whose performance is different from that of the program to be analyzed, the error will increase.

This specification area consists of the following combo boxes and specification area.

(a) [Insert] combo box

This combo box is used to select the insertion position of the memory area for allocating the profile routine output as an intermediate file during performance analysis or tuning.

The selectable items are "The Region", "In Post-region", "In Pre-region", and "Set Address".

"The Region" is selected in the default condition.

Remarks 1. This combo box is invalid if the following condition is satisfied.

- If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked
- If the [Use probe code for analyzing (for original execution module only).] radio button on the [Execution Item Selection Panel](#) is not checked

2. The items other than the "The Region" in this combo box are invalid if the following condition is satisfied.

- If the TWROUTINE segment is defined for the target link directive file

(b) [Position] combo box

This combo box is used to select the insertion target area (segment name) of the memory area for allocating the profile routine output as an intermediate file during performance analysis or tuning.

All text attribute segments specified in the link directive file and all sections specified in the link directive file can be selected.

In the default condition, "TWROUTINE" is selected if "TWROUTINE" exists, and "TEXT" is selected next if "TEXT" exists.

Usually specify "TEXT segment region" or "behind of .text section".

Remark This combo box is invalid if the following condition is satisfied.

- If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked
- If the [Use probe code for analyzing (for original execution module only).] radio button on the [Execution Item Selection Panel](#) is not checked
- If the TWROUTINE segment is defined for the target link directive file

(c) [Address] specification area

This area is used to specify the insertion target area (address) of the memory area for allocating the profile routine output as an intermediate file during performance analysis or tuning.

The range of values that can be input is from 0x0 to maximum address of the chip specified by the project.

Nothing will be specified in the default condition.

Caution Only multiples of 4 can be specified in this specification area.

Remark This specification area is invalid if any of the following conditions is satisfied.

- If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked
- If the [Use probe code for analyzing (for original execution module only).] radio button on the [Execution Item Selection Panel](#) is not checked
- If an item other than "Set Address" is selected in the [Insert] combo box

(4) Profile table specification area

This area is used to make settings related to the memory area for allocating the profile table output as an intermediate file during performance analysis or tuning.

Specify an area that can be written and that has a sufficient capacity. Specify the allocation destination in the RAM area.

This specification area consists of the following combo boxes, specification area, display area, and buttons.

(a) [Insert] combo box

This combo box is used to select the insertion position of the memory area for allocating the profile table output as an intermediate file during performance analysis or tuning.

The selectable items are "The Region", "In Post-region", "In Pre-region", and "Set Address".

"The Region" is selected in the default condition.

Remark The menus other than the "The Region" of this combo box are invalid if the following condition is satisfied.

- If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked
- If the [Use probe code for analyzing (for original execution module only).] radio button on the [Execution Item Selection Panel](#) is not checked
- If the TWTABLE segment is defined for the target link directive file

(b) [Position] combo box

This combo box is used to select the insertion target area (segment name) of the memory area for allocating the profile table output as an intermediate file during performance analysis or tuning.

All data attribute segments specified in the link directive file and all sections specified in the link directive file can be selected.

In the default condition, "TWTABLE" is selected if "TWTABLE" exists, and "DATA" is selected next if "DATA" exists.

Usually specify "DATA segment region".

Remark This combo box is invalid if the following condition is satisfied.

- If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked
- If the [Use probe code for analyzing (for original execution module only).] radio button on the [Execution Item Selection Panel](#) is not checked
- If the TWTABLE segment is defined for the target link directive file

(c) [Address] specification area

This combo box is used to specify the insertion target area (address) of the memory area for allocating the profile table output as an intermediate file during performance analysis or tuning.

The range of values that can be input is from 0x0 to maximum address of the chip specified by the project.

Nothing will be specified in the default condition.

Caution Only multiples of 4 can be specified in this specification area.

Remark This specification area is invalid if the following condition is satisfied.

- If the [Analyze an original execution module.] check box on the [Execution Item Selection Panel](#) is not checked
- If the [Use probe code for analyzing (for original execution module only).] radio button on the [Execution Item Selection Panel](#) is not checked
- If an item other than "Set Address" is selected in the [Insert] combo box

(d) Size display area

This area is used to display the size (in bytes) of the profile table that can be acquired by clicking the <Calculate> button.

If the [Use probe code for analyzing (for original execution module only).] radio button on the [Execution Item Selection Panel](#) is not checked, "None" is displayed.

"Uncalculate" is displayed in the default condition.

Remark Because the hole size is not included in the calculated size, the memory capacity actually occupied increases by the hole size.

(e) <Calculate> button

This button is used to calculate the size (in bytes) of the profile table and display the result of calculation in the size display area.

However, calculation may take time.

Remark This button is invalid if the following condition is satisfied.

- If the size of the profile table has already been calculated

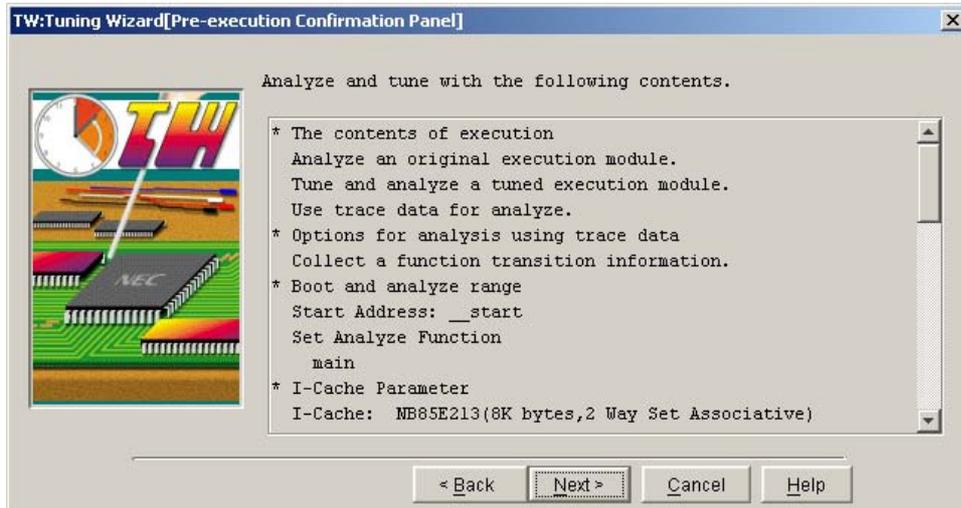
(f) <Cancel> button

This button is used to stop calculation of the size of the profile table.

Pre-execution Confirmation Panel

This panel is used to report that settings of performance analysis or tuning items required for the TW850 to execute wizard processing (performance analysis or tuning) has been completed.

Figure 6-24 Pre-execution Confirmation Panel



This section describes the following items.

- [Explanation of each area](#)

Explanation of each area

This panel consists of the following items.

- [Message display area](#)

(1) Message display area

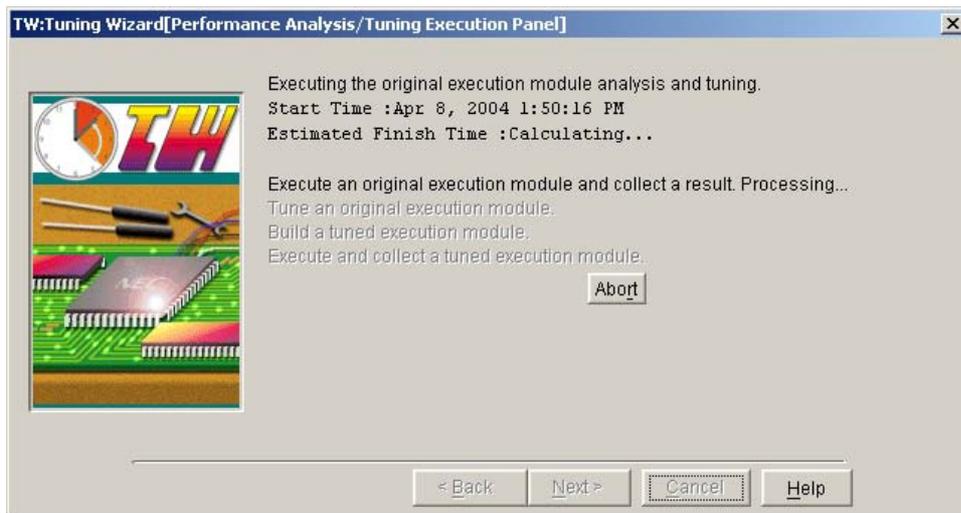
This area displays a message reporting that settings of the performance analysis or tuning items required for the TW850 to execute wizard processing (performance analysis or tuning) has been completed.

Performance Analysis/Tuning Execution Panel

This panel is used to report that the TW850 is executing wizard processing (performance analysis or tuning). The execution contents of the TW850 are as follows.

- Executing an execution module for original analysis and acquiring the results of performance analysis when performance analysis processing on an execution module for original analysis is performed
- Building and executing an execution module for profile, and acquiring the results of performance analysis when performance analysis processing is performed using a probe code insertion method
- Tuning, building, executing a tuned execution module, and acquiring the results of performance analysis and tuning when tuning and performance analysis processing on a tuned execution module is performed

Figure 6-25 Performance Analysis/Tuning Execution Panel



This section describes the following items.

- [Explanation of each area](#)

Explanation of each area

This panel consists of the following items.

- [Message display area](#)

(1) Message display area

This area displays a message reporting that the TW850 is executing wizard processing (performance analysis or tuning).

The message contents include the execution status, start time, and expected end time of the processing in progress. If an execution module for profile is generated, the size of the profile table is also displayed.

Remark Because the hole size is not included in the calculated size of the profile table, the memory capacity actually occupied increases by the hole size.

(a) <Abort> button

This button is used to stop wizard processing (performance analysis or tuning).

Remark It may take time for wizard processing to actually stop after this button is clicked.

(b) <Retry> button

This button is used to re-execute wizard processing (performance analysis or tuning).

Execution Completion Confirmation Panel

This panel is used to report that the TW850 has completed wizard processing (performance analysis or tuning).

Figure 6-26 Execution Completion Confirmation Panel



This section describes the following items.

- [Explanation of each area](#)

Explanation of each area

This panel consists of the following items.

- [Message display area](#)

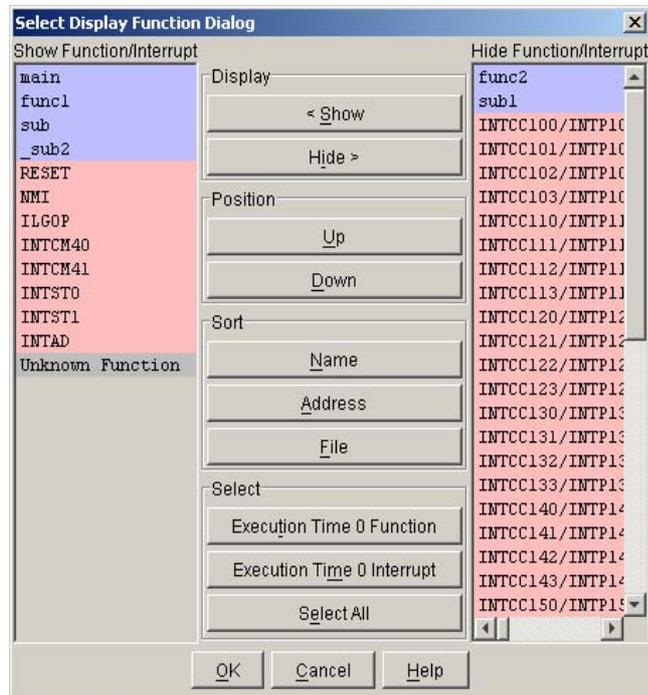
(1) Message display area

This area displays a message (such as the folder name storing an output file and the output file name) reporting that the TW850 has correctly completed wizard processing (performance analysis or tuning).

Select Display Function Dialog Box

This dialog box is used to select functions or interrupts displayed on the [Function Transition Information Window](#).

Figure 6-27 Select Display Function Dialog Box



This section describes the following items.

- [Opening](#)
- [Explanation of each area](#)

Opening

This dialog box can be opened as follows.

- After the [Function Transition Information Window](#) has become active in the [Main Window](#),
 - Select the right mouse button menu -> [Select Display Function...].
 - Select the [View] menu -> [Select Display Function ...].

Explanation of each area

This dialog box consists of the following items.

- [\[Show Function/Interrupt\] area](#)
- [Function display specification area](#)
- [Function move specification area](#)
- [Function sort specification area](#)
- [Function select specification area](#)
- [\[Hide Function/Interrupt\] area](#)
- [Function buttons](#)

(1) [Show Function/Interrupt] area

This area is used to select functions and interrupts displayed on the [Function Transition Information Window](#). The functions and interrupts displayed on the [Function Transition Information Window](#) at that time are displayed in the same sequence in the default condition.

- Remarks
1. If a function or an interrupt is clicked, the item is selected. Two or more functions and interrupts can be specified by clicking them while holding down the Ctrl key.
 2. Functions and interrupts can be moved in this area by dragging them. However, functions and interrupts cannot be moved if they are specified mixed together.

(2) Function display specification area

This area is used to specify whether functions and interrupts are displayed on the [Function Transition Information Window](#).

This specification area consists of the following buttons.

(a) <<Show> button

Adds the function or interrupt selected in the [Hide Function/Interrupt] area to the lowest position of the [Show Function/Interrupt] area.

Remark This button is invalid if the following condition is satisfied.

- If a function or an interrupt is not selected in the [Hide Function/Interrupt]

(b) <Hide>> button

Adds functions or interrupt selected in the [Show Function/Interrupt] area to the lowest position of the [Hide Function/Interrupt] area.

Remark This button is invalid if the following condition is satisfied.

- If a function or an interrupt is not selected in the [Show Function/Interrupt] area

(3) Function move specification area

This area is used to move the function or interrupt selected in the [Show Function/Interrupt] area or the [Hide Function/Interrupt] area to a position one higher or lower.

This specification area consists of the following buttons.

(a) <Up> button

Moves the function or interrupt selected in the [Show Function/Interrupt] area or the [Hide Function/Interrupt] area to a position one higher.

Remark This button is invalid if the following condition is satisfied.

- If the function or interrupt is not selected in the [Show Function/Interrupt] area or the [Hide Function/Interrupt] area

Caution An interrupt cannot be moved to a position higher than a function.

(b) <Down> button

Moves the function or interrupt selected in the [Show Function/Interrupt] area or the [Hide Function/Interrupt] area to a position one lower.

Remark This button is invalid if the following condition is satisfied.

- If the function or interrupt is not selected in the [Show Function/Interrupt] area or the [Hide Function/Interrupt] area

Caution A function cannot be moved to a position lower than an interrupt.

(4) Function sort specification area

This area is used to sort the functions or interrupts in the [Show Function/Interrupt] area and the [Hide Function/Interrupt] area.

This specification area consists of the following buttons.

(a) <Name> button

This button is used to sort the functions or interrupts in the selected [Show Function/Interrupt] area or [Hide Function/Interrupt] area in the name order.

Remark This button is invalid if the following condition is satisfied.

- If the [Show Function/Interrupt] area or [Hide Function/Interrupt] area is not selected

Caution A function is always displayed at a position higher than an interrupt.

(b) <Address> button

This button is used to sort the functions or interrupts in the selected [Show Function/Interrupt] area or [Hide Function/Interrupt] area in the address order.

Remark This button is invalid if the following condition is satisfied.

- If the [Show Function/Interrupt] area or [Hide Function/Interrupt] area is not selected

Caution A function is always displayed at a position higher than an interrupt.

(c) <File> button

This button is used to sort the functions in the selected [Show Function/Interrupt] area or [Hide Function/Interrupt] area in the order of files that functions belong.

Remarks 1. This button is invalid if the following condition is satisfied.

- [Show Function/Interrupt] area or [Hide Function/Interrupt] area
2. If two or more functions belong to the same file, they are displayed in the order before sorting.

(5) Function select specification area

This area is used to select the functions and interrupts whose execution time is 0, or all the functions and interrupts in the [Show Function/Interrupt] area and the [Hide Function/Interrupt] area.

This specification area consists of the following buttons.

(a) <Execution Time 0 Function> button

This button is used to select the functions whose execution time is 0 in the selected [Show Function/Interrupt] area or [Hide Function/Interrupt] area.

Remark This button is invalid if the following condition is satisfied.

- If the [Show Function/Interrupt] area or [Hide Function/Interrupt] area is not selected

(b) <Execution Time 0 Interrupt> button

This button is used to select the interrupts whose execution time is 0 in the selected [Show Function/Interrupt] area or [Hide Function/Interrupt] area.

Remark This button is invalid if the following condition is satisfied.

- If the [Show Function/Interrupt] area or [Hide Function/Interrupt] area is not selected

(c) <Select All> button

This button is used to select all the functions or interrupts in the selected [Show Function/Interrupt] area or [Hide Function/Interrupt] area.

Remark This button is invalid if the following condition is satisfied.

- If the [Show Function/Interrupt] area or [Hide Function/Interrupt] area is not selected

(6) [Hide Function/Interrupt] area

This area is used to select the functions and interrupts not displayed on the [Function Transition Information Window](#).

The functions and interrupts not displayed on the [Function Transition Information Window](#) at that time are displayed in the same sequence in the default condition.

Remarks 1. If a function or an interrupt is clicked, the item is selected. Two or more functions and interrupts can be selected.

2. Functions and interrupts can be moved in this area by dragging them.

However, functions and interrupts cannot be moved if they are specified mixed together.

(7) Function buttons

The function buttons consist of the following buttons.

(a) <OK> button

Sets the [Function Transition Information Window](#) so that functions and interrupts on the display function list are displayed, and then closes this dialog box.

(b) <Cancel> button

Closes this dialog box.

(c) <Help> button

Starts the help.

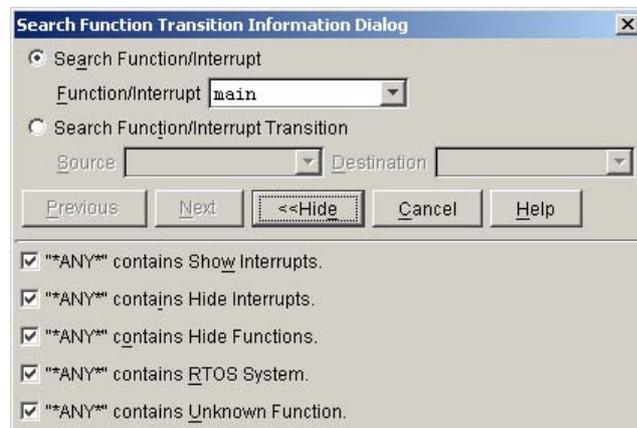
Search Function Transition Information Dialog Box

This dialog box is used to search a function execution or a function transition line on the [Function Transition Information Window](#).

Figure 6-28 Search Function Transition Information Dialog Box



Figure 6-29 Search Function Transition Information Dialog Box (When Expansion Search Condition Is Displayed)



This section describes the following items.

- [Opening](#)
- [Explanation of each area](#)

Opening

This dialog box can be opened as follows.

- After selecting [Function Transition Information Window](#) of the [Main Window](#),
 - Select the [Edit] menu -> [Search...].
- On the [Function Transition Information Window](#),
 - Click the Search button.

Explanation of each area

This dialog box consists of the following items.

- [Function search condition specification area](#)
- [Function transition search condition specification area](#)
- [Expansion search condition specification area](#)
- [Function buttons](#)

(1) Function search condition specification area

This area is used to specify search conditions related to functions in the [Function Transition Information Window](#). This specification area consists of the following radio button and combo box.

(a) [Search Function/Interrupt] radio button

This radio button is used to specify whether a function or an interrupt is searched.

If nothing is selected on the [Function Transition Information Window](#), or if any of function box, interrupt box, or function execution line is selected, this radio button is checked in the default condition.

If this radio button is checked, a function or an interrupt is searched.

(b) [Function/Interrupt] combo box

This combo box is used to specify the function name or interrupt name to be searched.

The names of functions acquired by executing performance analysis processing and the names of all interrupts of chip specified in the project can be selected. The item "**ANY**", which means any function and interrupt, is also displayed on the list.

If any of function box, interrupt box, or function execution line is selected on the [Function Transition Information Window](#), the selected function name or interrupt name is displayed in the default condition.

Remarks 1. This combo box is invalid if the following condition is satisfied.

- If the [Search Function/Interrupt] radio button is not checked
- 2. If two or more functions of the same name exist, "function name (file name)" is displayed.
If two or more functions of the same function name or file name exist, "function name (absolute path name)" is displayed.

(2) Function transition search condition specification area

This area is used to specify search conditions related to function transition in the [Function Transition Information Window](#).

This specification area consists of the following radio button and combo boxes.

(a) [Search Function/Interrupt Transition] radio button

This radio button is used to specify whether a function or a function transition line is searched.

If a function transition line is selected on the [Function Transition Information Window](#), this radio button is checked in the default condition.

If this radio button is checked, a function or a function transition line is searched.

(b) [Source] combo box

This combo box is used to specify the function name or interrupt name that the search condition changes.

The names of functions acquired by executing performance analysis processing and the names of all interrupts of chip specified in the project can be selected. The item "**ANY**", which means any function and interrupt, is also displayed on the list.

If the function execution line is selected in the [Function Transition Information Window](#), the function name or interrupt name that indicates the source of the function transition line selected is displayed in the default condition.

Remarks 1. This combo box is invalid if the following condition is satisfied.

- If the [Search Function/Interrupt Transition] radio button is not checked
- 2. If two or more functions of the same name exist, "function name (file name)" is displayed.
If two or more functions of the same function name or file name exist, "function name (absolute path name)" is displayed.

(c) [Destination] combo box

This combo box is used to specify the function name or interrupt name as the transition destination of the search condition.

The names of functions acquired by executing performance analysis processing and the names of all interrupts of chip specified in the project can be selected. The item **"*ANY*"**, which means any function and interrupt, is also displayed on the list.

If the function execution line is selected in the [Function Transition Information Window](#), the function name or interrupt name that indicates the destination of the function transition line selected is displayed in the default condition.

Remarks 1. This combo box is invalid if the following condition is satisfied.

- If the [Search Function/Interrupt Transition] radio button is not checked
- 2. If two or more functions of the same name exist, "function name (file name)" is displayed.
If two or more functions of the same function name or file name exist, "function name (absolute path name)" is displayed.

(3) Expansion search condition specification area

This area is used to specify expansion search conditions related to functions and function transition in the [Function Transition Information Window](#).

This specification area consists of the following check boxes.

(a) **"*ANY*" contains Show Interrupts.** check box

This check box is used to specify whether the interrupt displayed on the [Function Transition Information Window](#) is searched if **"*ANY*"** is specified in the [Function/Interrupt] combo box, [Source] combo box, or [Destination] combo box.

This check box is checked in the default condition.

If this check box is checked, the interrupt displayed on the [Function Transition Information Window](#) is searched.

(b) **"*ANY*" contains Hide Interrupts.** check box

This check box is used to specify whether the interrupt not displayed on the [Function Transition Information Window](#) is searched if **"*ANY*"** is specified in the [Function/Interrupt] combo box, [Source] combo box, or [Destination] combo box.

This check box is checked in the default condition.

If this check box is checked, the interrupt not displayed on the [Function Transition Information Window](#) is searched.

(c) **"*ANY*" contains Hide Functions.** check box

This check box is used to specify whether the function not displayed on the [Function Transition Information Window](#) is searched if **"*ANY*"** is specified in the [Function/Interrupt] combo box, [Source] combo box, or [Destination] combo box.

This check box is checked in the default condition.

If this check box is checked, the function not displayed on the [Function Transition Information Window](#) is searched.

(d) **"*ANY*" contains RTOS System.** check box

This check box is used to specify whether the RTOS system is searched if **"*ANY*"** is specified in the [Function/Interrupt] combo box, [Source] combo box, or [Destination] combo box.

This check box is checked in the default condition.

If this check box is checked, the RTOS system is searched.

(e) **"*ANY*" contains Unknown Function.** check box

This check box is used to specify whether the unknown function is searched if **"*ANY*"** is specified in the [Function/Interrupt] combo box, [Source] combo box, or [Destination] combo box.

This check box is checked in the default condition.

If this check box is checked, the unknown function is searched.

Remark This specification area is invalid if the following condition is satisfied.

- If the <<<Hide> button is clicked

(4) Function buttons

The function buttons consist of the following buttons.

(a) <Prvious> button

Searches the function execution line or function transition line of the condition in the specified search condition specification area in the left direction, starting from the selected function execution line or function transition line.

Remark This button is invalid if any of the following conditions is satisfied.

- If the [Search Function/Interrupt] radio button is checked and data specified on the [Eunction/Interrupt] combo box is illegal
- If the [Search Function/Interrupt Transition] radio button is checked and data specified on the [Source] combo box or [Destination] combo box is illegal

(b) <Next>> button

Searches the function execution line or function transition line of the condition in the specified search condition specification area in the right direction, starting from the selected function execution line or function transition line.

Remark This button is invalid if either of the following conditions is satisfied.

- If the [Search Function/Interrupt] radio button is checked and data specified on the [Eunction/Interrupt] combo box is illegal
- If the [Search Function/Interrupt Transition] radio button is checked and data specified on the [Source] combo box or [Destination] combo box is illegal

(c) <Detail>>> button

Displays the Expansion search condition specification area.

(d) <<<Hide> button

Hides the Expansion search condition specification area.

(e) <Cancel> button

Closes this dialog box.

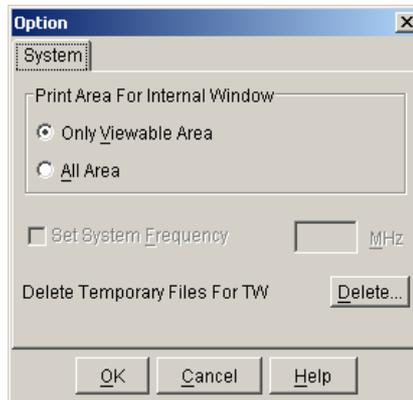
(f) <Help> button

Starts the help.

Option Dialog Box

This dialog box is used to specify how to print the [Callgraph Window](#) or [Function Transition Information Window](#) displayed in the internal window display area on the [Main Window](#), or whether intermediate files (temporary files) the TW850 output while it was executing performance analysis or tuning processing are to be deleted.

Figure 6-30 Option Dialog Box



This section describes the following items.

- [Opening](#)
- [Explanation of each area](#)

Opening

This dialog box can be opened as follows.

- On the [Main Window](#),
 - Select the [Tool] menu -> [Option...].

Explanation of each area

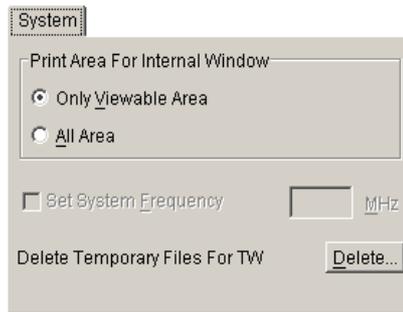
This dialog box consists of the following items.

- [System-related tab](#)
- [Function buttons](#)

(1) System-related tab

This tab is used to set options related to the overall processing of the TW850.

Figure 6-31 [System] Tab



This tab consists of the following specification areas, check box, and button.

(a) Printing method specification area

- [Only Viewable Area] radio button

This radio button is used to specify whether the [Callgraph Window](#) or [Function Transition Information Window](#) displayed in the internal window display area of the [Main Window](#) are printed as a displayed image. This radio button is checked in the default condition.

If this radio button is checked, the [Callgraph Window](#) or [Function Transition Information Window](#) is printed as a displayed image.

- [All Area] radio button

This radio button is used to specify whether the [Callgraph Window](#) or [Function Transition Information Window](#) displayed in the internal window display area of the [Main Window](#) as the entire [Callgraph Window](#) or [Function Transition Information Window](#).

This radio button is not checked in the default condition.

If this radio button is checked, the [Callgraph Window](#) or [Function Transition Information Window](#) is printed as the entire [Callgraph Window](#) or [Function Transition Information Window](#).

(b) [Set System Frequency] check box

This check box is used to specify whether the information for the operating frequency specification area (operating frequency of the execution engine) can be specified.

If this check box is checked, the information for the operating frequency specification area (operating frequency of the execution engine) can be specified.

This check box is not checked in the default condition.

Remark This check box is invalid if the following condition is satisfied.

- If the TW850 can automatically detect the operating frequency of the execution engine (debugger or simulator)

(c) Operating frequency specification area

This area is used to specify the operating frequency (in MHz) of the execution engine (such as a debugger or simulator).

Nothing will be specified in the default condition.

Remark This specification area is invalid if the following condition is satisfied.

- If the [Set System Frequency] check box is not checked

(d) <Delete...> button

This button is used to delete the intermediate files (temporary files) output by the TW850 while it was executing performance analysis or tuning processing.

Of the intermediate files output by the TW850, files other than those shown below are deleted.

- Project file for the TW850 "(original project file name).pdb"
- Tuned execution module "(original execution module name)_o.out"
- Hex format file of tuned execution module "(original execution module name)_o.hex" (If generation of a hex file is specified)
- Tcl file of tuned execution module "(original execution module name)_o.tcl"

(2) Function buttons

The function buttons consist of the following buttons.

(a) <OK> button

Executes processing corresponding to the information specified by the system-related tab, and then closes this dialog box.

(b) <Cancel> button

Closes this dialog box without executing processing corresponding to the information specified by the System-related tab.

(c) <Help> button

Starts the help.

Version Information Dialog Box

This dialog box is used to display version information (such as the version and copyright) of the TW850.

Figure 6-32 Version Information Dialog Box



This section describes the following items.

- [Opening](#)
- [Explanation of each area](#)

Opening

This dialog box can be opened as follows.

- On the [Main Window](#),
 - Select the [\[Help\]](#) menu -> [\[About...\]](#).

Explanation of each area

This dialog box consists of the following items.

- [Message display area](#)
- [Function buttons](#)

(1) Message display area

This area displays the version information (such as the version and copyright) of the TW850.

(2) Function buttons

The function buttons consist of the following button.

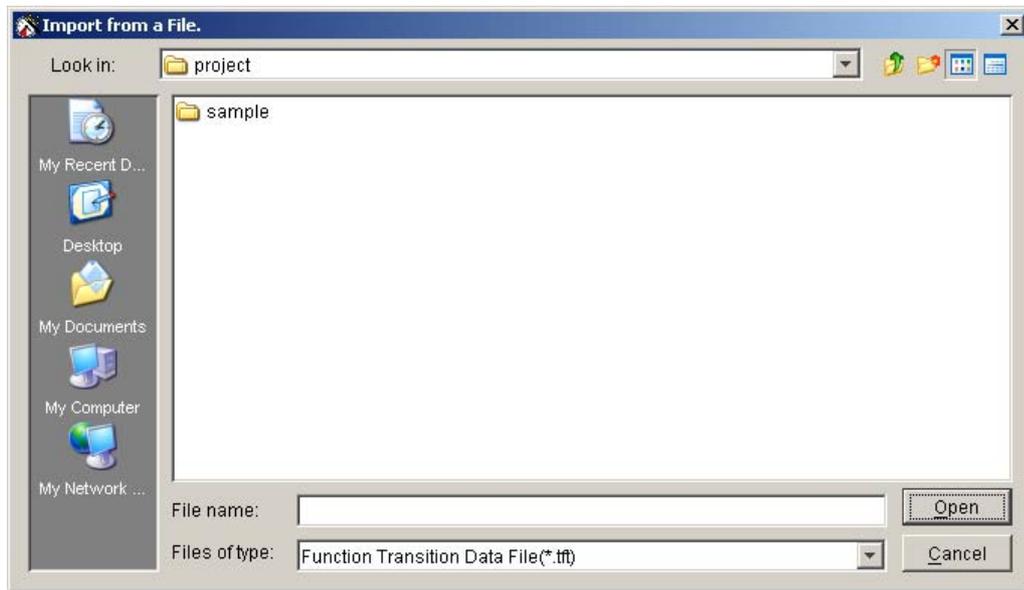
(a) <QK> button

Closes this dialog box.

File Selection Dialog Box

This dialog box is used to specify files required by the TW850 to execute performance analysis or tuning processing (such as timer control code block).

Figure 6-33 File Selection Dialog Box



This section describes the following items.

- [Opening](#)
- [Explanation of each area](#)

Opening

This dialog box can be opened as follows.

- On the [Probe Code Insertion Analysis Method Option Selection Panel](#),
 - Click the <Refer...> button.
- On the [Main Window](#),
 - Select [File] menu -> [Import...].
- On the [Main Window](#), after moving the focus to the [Function Information Table Window](#), [Function Transition Information Window](#), [Execution Module Information Window](#), or [Function Detail Information Window](#),
 - Select [File] menu -> [Export...].

Explanation of each area

This dialog box consists of the following items.

- Opening location specification combo box
- File name display area
- File name specification area
- File type specification combo box
- Function buttons

(1) Opening location specification combo box

This combo box is used to select the name of the folder stores the corresponding file (such as timer control code block).

If a folder name is too long to display, "..." is displayed at the end of the name.

(2) File name display area

This area is used to display a list of the folders and files stored in the corresponding folder, based on the information specified in the opening location specification combo box and file type specification combo box.

Remarks 1. If a file name is clicked, the file is selected and displayed in the file name specification area.

2. If a file name is double-clicked, the file is selected and becomes the same status as when the <Open> or <Write> button is clicked.

If a folder name is double-clicked, all the folders and files in the specified folder are listed in the file name display area.

(3) File name specification area

This area is used to specify the name of the file required by the TW850 to execute performance analysis or tuning (such as timer control code block).

(4) File type specification combo box

This combo box is used to select the type of files to be displayed in the file name display area.

(5) Function buttons

The function buttons consist of the following buttons.

(a) <Up One Level> button



Changes the folder to be displayed in the file name display area to a folder one level up. However, this button is invalid while the root folder is displayed in the file name display area.

(b) <Create New Folder> button



Creates a new folder in the folder currently displayed in the file name display area.

(c) <Select>, <Open>, <Write> button

Closes this dialog box after reporting to the TW850 that the file specified in the file name specification area is a file required by the TW850 to execute performance analysis or tuning processing, or after opening the file specified in the file name specification area, or after saving the execution status of the TW850 to the file specified in the file name specification area.

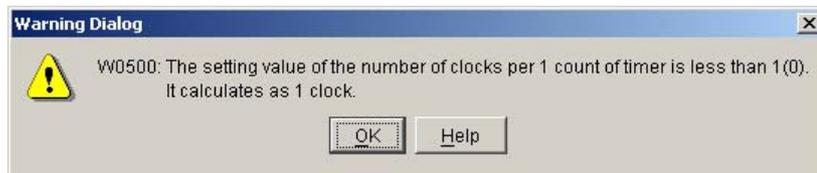
(d) <Cancel> button

Closes this dialog box without reporting to the TW850 that the file specified in the file name specification area is a file required by the TW850 to execute performance analysis or tuning processing, without opening the file specified in the file name specification area, or without saving the execution status of the TW850 to the file specified in the file name specification area.

Message Dialog Box

This dialog box is used to display information to be reported to a user (such as the progress of processing and occurrence of a warning) detected by the TW850 while it was executing performance analysis or tuning processing.

Figure 6-34 Message Dialog Box



This section describes the following items.

- [Opening](#)
- [Explanation of each area](#)

Opening

This dialog box is automatically opened when the TW850 detects the information to be reported to the user (such as the progress of processing and occurrence of a warning) while it is executing performance analysis or tuning processing.

Explanation of each area

This dialog box consists of the following items.

- [Message display area](#)
- [Function buttons](#)

(1) Message display area

This area displays the information to be reported to the user (such as the progress of processing and occurrence of a warning) detected by the TW850 while it was executing performance analysis or tuning processing.

Refer to "[A.2 Information Messages](#)" and "[A.3 Warning Messages](#)" for details of the information to be reported to the user (such as the progress of processing and occurrence of a warning).

(2) Function buttons

The function buttons consist of the following buttons.

- (a) <OK> button
Closes this dialog box.
- (b) <Help> button
Starts the help.

Error Dialog Box

This dialog box is used to display information to be reported to a user (such as the occurrence of a fatal error) detected by the TW850 when it was started or while it was executing performance analysis or tuning processing.

Figure 6-35 Error Dialog Box (When Starting TW850)

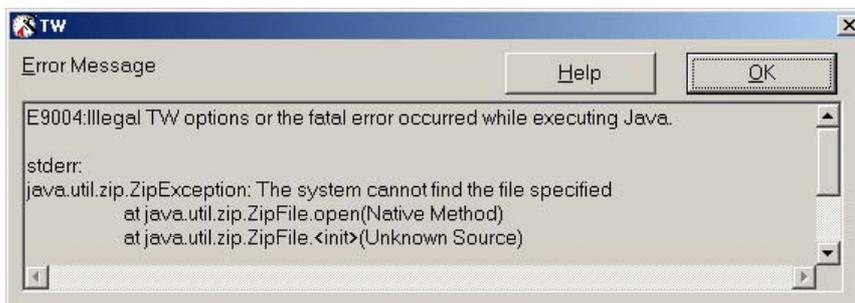


Figure 6-36 Error Dialog Box (When Executing TW850)



This section describes the following items.

- [Opening](#)
- [Explanation of each area](#)

Opening

This dialog box is automatically opened when the TW850 is started or detects the information to be reported to the user (such as the occurrence of a fatal error) while it is executing processing.

Explanation of each area

This dialog box consists of the following items.

- [Message display area](#)
- [Function buttons](#)

(1) Message display area

This area displays the information to be reported to the user (such as the occurrence of a fatal error) detected by the TW850 when it is started or while it was executing performance analysis or tuning processing.

Refer to "[A.4 Error Messages](#)" for details of the information to be reported to the user (such as the occurrence of a fatal error).

(2) Function buttons

The function buttons consist of the following buttons.

(a) <Detail>>> button

Displays details of the corresponding information in this dialog box.

Double-clicking the message in the details starts the help for that message.

(b) <OK> button

Closes this dialog box.

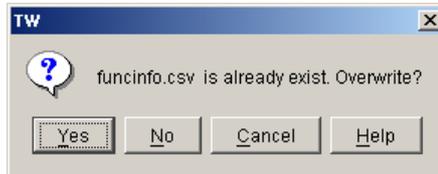
(c) <Help> button

Starts the help for the displayed message.

Verification Dialog Box

This dialog box is used to display inquiry information (such as the problem that prevents continuation of processing) detected by the TW850 while it was executing performance analysis or tuning processing.

Figure 6-37 Verification Dialog Box



This section describes the following items.

- [Opening](#)
- [Explanation of each area](#)

Opening

This dialog box can be opened as follows.

- On the [Main Window](#),
 - Select [File] menu -> [Exit...].
 - Select the [Edit] menu -> [Restore From History...].
 - Click the Restore From History button on the toolbar.
- On the tuning wizard (such as the [Initial Screen Panel](#), [Execution Module Information Acquisition Panel](#), or [Execution Item Selection Panel](#)),
 - Click the <Close> button.
 - Click the <Cancel> button.
- On the [File Selection Dialog Box](#),
 - Exist a specified file already and click the <Write> button.
- On the [System] tab in the [Option Dialog Box](#),
 - Click the <Delete...> button.

Explanation of each area

This dialog box consists of the following items.

- [Message display area](#)
- [Function buttons](#)

(1) Message display area

This area is used to display the inquiry information (such as a problem that prevents continuation of processing) detected by the TW850 while it was executing performance analysis or tuning processing.

(2) Function buttons

The function buttons consist of the following buttons.

(a) <Yes> button

Executes processing corresponding to the contents of the displayed message, and then closes this dialog box.

(b) <No> button

Closes this dialog box without executing processing corresponding to the contents of the displayed message.

(c) <Cancel> button

Closes this dialog box without executing processing to save data to the file specified in the [File Selection Dialog Box](#).

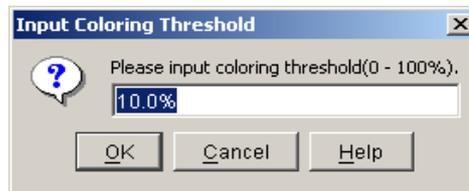
(d) <Help> button

Starts the help.

Input Dialog Box

This dialog box is used to specify the information (such as the color boundary, display magnification rate, and root function) requested by the TW850.

Figure 6-38 Input Dialog Box



This section describes the following items.

- [Opening](#)
- [Explanation of each area](#)

Opening

This dialog box can be opened as follows.

- On the [Main Window](#), after moving the focus to the [Callgraph Window](#),
 - Select [[View](#)] menu -> [[Callgraph](#)] -> [[Coloring Threshold...](#)].
 - Select [[View](#)] menu -> [[Callgraph](#)] -> [[Zoom...](#)].
 - Select [[View](#)] menu -> [[Callgraph](#)] -> [[Change Root Function...](#)].

Explanation of each area

This dialog box consists of the following items.

- [Message display area](#)
- [Request information specification area and request information specification combo box](#)
- [Function buttons](#)

(1) Message display area

This area displays an explanation messages related to this dialog box.

(2) Request information specification area and request information specification combo box

This combo box is used to specify the information (such as the color boundary, display magnification rate, and root function) requested by the TW850.

(3) Function buttons

The function buttons consist of the following buttons.

(a) <OK> button

Reports to the TW850 that the information requested by the TW850 is the information specified in the request information specification area or request information specification combo box, and then closes this dialog box.

(b) <Cancel> button

Closes this dialog box without reporting to the TW850 that the information requested by the TW850 is the information specified in the request information specification area or request information specification combo box.

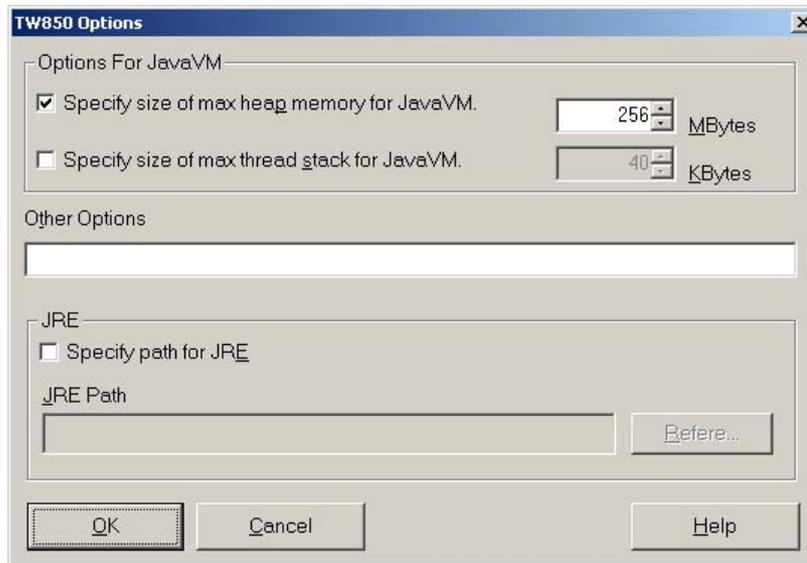
(c) <Help> button

Starts the help.

TW850 Options Dialog Box

This dialog box is used to specify start options of the TW850.

Figure 6-39 TW850 Options Dialog Box



This section describes the following items.

- [Opening](#)
- [Explanation of each area](#)

Opening

This dialog box can be opened as follows.

- On the [Main Window](#) of project manager the PM plus,
 - Select the [Tool] menu -> [TW850 Options...].
 - Click the <TW850 Options...> button.

Explanation of each area

This dialog box consists of the following items.

- [JavaVM option specification area](#)
- [\[Other Options\] specification area](#)
- [JRE specification area](#)
- [Function buttons](#)

(1) JavaVM option specification area

This area is used to set options related to JavaVM.

This specification area consists of the following check boxes and specification areas.

(a) [Specify size of max heap memory for JavaVM] check box

This check box is used to specify whether the maximum memory size that JavaVM can use as a heap memory area is specified.

This check box is checked in the default condition.

If this check box is checked, the maximum size usable as a heap memory area is specified.

If the memory necessary for analyzing the project is insufficient, the heap memory area used can be increased.

(b) Maximum heap memory size specification area

This area is used to specify the maximum memory size (in Mbytes) that JavaVM can use as a heap memory area.

256 is specified in the default condition.

Remarks 1. If the memory necessary for analyzing the project is insufficient, specify 65 Mbytes or more. It is recommended to specify 128 Mbytes or more if the number of source files is 500 and the number of functions is 5000, and 256 Mbytes or more if many memories are mounted.

2. This specification area is invalid if the following condition is satisfied.

- If the [Specify size of max heap memory for JavaVM] check box is not checked

(c) [Specify size of max thread stack for JavaVM] check box

This is used to specify whether the maximum memory size that JavaVM can use as a thread stack memory area is specified.

This radio button is not checked in the default condition.

If this check box is checked, the maximum size usable as a thread stack memory area is specified.

(d) Thread stack size specification area

This area is used to specify the maximum memory size (in Kbytes) that JavaVM can use as a thread stack memory area.

40 is specified in the default condition.

Remark This specification area is invalid if the following condition is satisfied.

- If the [Specify size of max thread stack for JavaVM] check box is not checked

(2) [Other Options] specification area

This area is used to set options related to wizard processing (performance analysis or tuning processing) of the TW850.

Nothing will be specified in the default condition.

Remark Refer to "[3.1.2 Start options](#)" for details of the options.

(3) JRE specification area

This area is used to specify the JRE (Java runtime environment) path.

This specification area consists of the following check box and specification area, and button.

(a) [Specify path for JRE] check box

This check box is used to specify whether the JRE path used in the TW850 is specified.

This radio button is not checked in the default condition.

If this check box is checked, the JRE path used in the TW850 is specified.

When the check box is not checked, the JRE path for a version supported by the TW850 is used.

(b) [JRE Path] specification area

This area is used to specify the JRE full path used in the TW850 (e.g. C:\Program Files\Java\j2re1.4.2_04\bin).

The path that the JRE supported by the TW850 is installed is specified in the default condition.

When the TW850 starts, the path specified in this process is added to the beginning of the environment variable PATH.

Remarks 1. The JRE path is acquired from "JavaHome" in the registry key below.

HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Java Runtime Environment*version*

2. This specification area is invalid if the following condition is satisfied.

- If the [Specify path for JRE] check box is not checked

(c) <Refer...> button

This button is used to open dialog box select the JRE path.

(4) Function buttons

The function buttons consist of the following buttons.

(a) <OK> button

Reports the information specified in the JavaVM option specification area and other option specification area to the PM plus, and then closes this dialog box.

(b) <Cancel> button

Closes this dialog box without reporting the information specified in the JavaVM option specification area and other option specification area to the PM plus.

(c) <Help> button

Starts the help.

APPENDIX A MESSAGES

This appendix explains the messages displayed in the message display area of the [Main Window](#), [Message Dialog Box](#), and [Error Dialog Box](#) of the TW850.

A.1 Display Format

A message is generated if the TW850 detects "the information to be reported to a user (such as progress of processing, occurrence of a warning, and occurrence of a fatal error)" while it is executing processing and displayed in the message display area of the [Main Window](#), [Message Dialog Box](#), or [Error Dialog Box](#).

Figure A-1 Message Display Area in Main Window

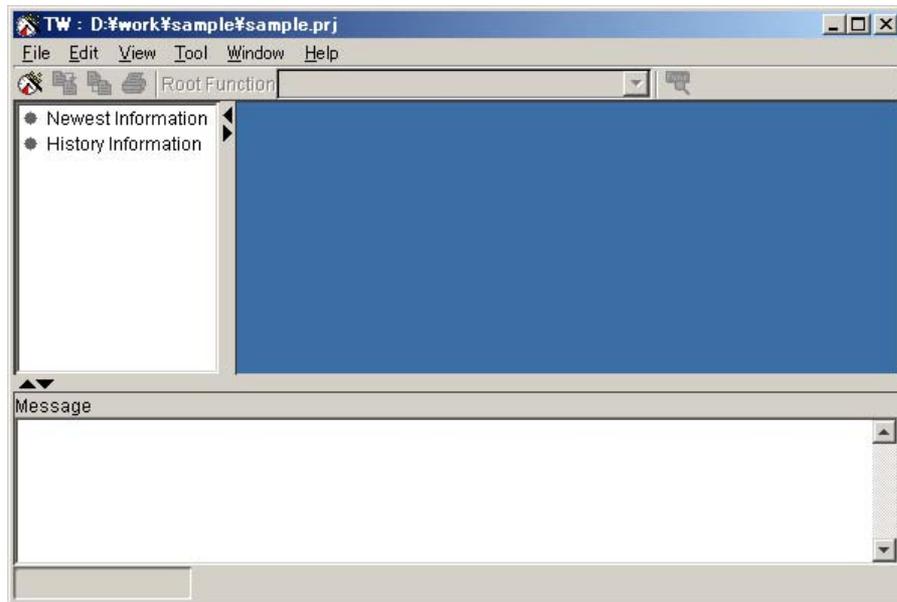


Figure A-2 Message Dialog Box



Figure A-3 Error Dialog Box



The TW850 classifies its messages into three types: information messages, warning messages, and error messages. When a message is output, the first character of the message indicates its type.

I: [Information Messages](#)

When the TW850 detects information (such as progress of processing), it outputs an information message to the message display area of the [Main Window](#) or [Message Dialog Box](#), and continues processing.

W: [Warning Messages](#)

When the TW850 detects information (such as occurrence of a warning), it outputs a warning message to the message display area of the [Main Window](#) or [Message Dialog Box](#), and continues processing.

E: [Error Messages](#)

When the TW850 detects information (such as occurrence of a fatal error), it outputs an error message to the message display area of the [Main Window](#) or [Error Dialog Box](#), and stops processing.

A.2 Information Messages

The information messages that are output when the TW850 detects information (such as progress of processing) while it is executing processing are listed below (in the order of message numbers).

%s in an information message is determined when information (such as progress of processing) has been detected.

I0200: Failed to search specified conditions.

- 1) Information satisfying the specified condition could not be found, even though it was searched to the end.

I0500: Failed to substitute the symbol(%s), but TW ignore it.(%s)

- 1) Processing to assign symbol %s has been terminated abnormally. As the abnormal termination of assignment processing does not hinder the TW850 from continuing processing at present, the TW850 ignores the abnormal termination of assignment processing and continues processing.

I0501: The specification of file symbol was illegal, but TW ignore it.(%s)

- 1) The assembly source file may have a description of an illegal file symbol declaration. As the abnormal detection of a file symbol declaration does not hinder the TW850 from continuing processing at present, the TW850 ignores the abnormal detection of a file symbol declaration and continues processing.

I0502: The definition of the assembler symbol by the option(%s=%s) was illegal, but TW ignore it.(%s)

- 1) Illegal assembler symbol declaration %s may have been specified by an assembler option. The TW850 continues processing, assuming that 0 is set as the setting value of assembler symbol declaration %s.

I0503: Analyzing functions in (%s) source file(%s).

- 1) The TW850 reports progress %s of performance analysis processing on the function subject to performance analysis in source file %s and continues processing.

I0504: Process of analyzing functions was done.

- 1) The TW850 reports completion of analysis processing on all functions subject to performance analysis, and continues processing.

I0505: Analyzing callpairs in (%s) source file(%s)...

- 1) The TW850 reports progress %s of callpair information (calling relationship between functions) analysis processing on the function subject to performance analysis in source file %s and continues processing.

I0506: Process of analyzing callpairs was done.

- 1) The TW850 reports completion of callpair information (calling relationship between functions) processing on all functions subject to performance analysis, and continues processing.

I0507: Preparing to analyze callpairs...

- 1) This is a message output in the preprocessing for callpair analysis.

I0508: Process of analyzing the execution module for original analysis...

- 1) This is a message indicating that an execution module for original analysis is being analyzed.

I0509: Process of analyzing the execution module for profiling...

- 1) This is a message indicating that an execution module for profile is being analyzed.

I0510: Process of analyzing the tuned execution module...

- 1) This is a message indicating that a tuned execution module is being analyzed.

I0511: Process of analyzing the execution module was done.

- 1) This is a message indicating completion of analysis of execution module information.

I0800: Read the TW project file of earlier version(%s).

- 1) This is a message displayed if a project file for the TW850 saved with the TW850 of a version older than the TW850 currently used is read.

I1401: Analyzing trace data...

- 1) This is a message indicating that trace data is being analyzed.

I1402: Processing of analyzing trace data was done.

- 1) This is a message indicating completion of analysis of trace data.

I1500: Top address of region %s moved from %s to %s, because which was overlapped to existing region.

- 1) The size of area %s (segment or section) specified by the link directive file may cause overlapping of the memory area during allocation of the tuned execution module to memory. The TW850 continues processing, assuming that %s is specified as first address %s specified by the link directive file.

I1600: %s

- 1) The TW850 reports the message output by another tool (such as the C compiler or assembler) and continues processing.

I1800: Could not get device name, but TW ignore it.

- 1) Device information (such as the device name) could not be detected. As the abnormal termination of device information detection processing does not hinder the TW850 from continuing processing at present, the TW850 ignores the abnormal termination of device information detection processing and continues processing.

A.3 Warning Messages

The warning messages that are output when the TW850 detects information (such as occurrence of a warning) while it is executing processing are listed below (in the order of message numbers).

%s in a warning message is determined when information (such as occurrence of a warning) has been detected.

W0200: Current locale is different from the saved project's locale. So, characters may not be displayed normally.

- 1) The current "Regional Setting" of Windows may be different from the region set when the project information was saved. The TW850 starts processing assuming the current set value.

W0201: Java runtime environment version is later than the version(%s) TW supported.

It is recommended that the Java runtime environment TW supported should be used.

- 1) This Java execution environment is newer than the version supported by the TW850. Because this Java execution environment is not supported, correct operation cannot be guaranteed. Use the Java execution environment (Java2 Runtime Environment Standard Edition V1.4.2_04) supported by the TW850.

W0400: Could not find the symbol of RTOS.

TW regards it as the RTOS(%s).

- 1) The wrong RTOS may be specified on the [Initial Screen Panel](#) of the tuning wizard. Go back to the [Initial Screen Panel](#) of the tuning wizard, specify the correct RTOS, and execute analysis again.

W0500: The setting value of the number of clocks per 1 count of timer is less than 1(%s).

It calculates as 1 clock.

- 1) A value %s less than 1 may have been set as "the number of CPU clocks per timer count tw_clock_count" in the timer control code block. The TW850 starts processing assuming that 1 is set as "the number of CPU clocks per timer count".

W0700: TW project file(%s) is ignored.

- 1) The contents of project file %s may have been updated or destroyed. The TW850 does not perform reading processing of "the past analysis information" but deletes "the past analysis information" and starts processing.

W0701: Failed to restore the project(%s).

- 1) The contents of project file %s may have been destroyed. The TW850 does not restore the environment of project file %s and starts processing in the default environment.
- 2) Project file %s could not be detected. The TW850 does not restore the environment of project file %s and starts processing in the default environment.

W0800: Read the TW project file of later version(%s).

Initializes the project.

- 1) A project file for the TW850 saved with the TW850 of a version newer than the TW850 currently used has been read. Reading from the project file for the TW850 will be aborted and the project will be initialized.

W1000: Unknown interrupt information is contained in this device file(%s).

- 1) Information the TW850 cannot recognize is included in the device file. The TW850 will ignore this information and continue operating. A device file of a version not supported by the TW850 may be being used. Refer to the User's Manual and check if the version of the device file is one supported by the TW850.

W1500: Exist mapping of the section(%s) that may cause illegal mapping of functions.(%s)

- 1) Check if there is an expression that includes a section name reserved by the TW850 (section name starting with "TW...") in the input section that specifies section in the link directive file allocation. If such an allocation specification exists, tuning may not be correctly executed. Describe the input section so that no section reserved by the TW850 is included.
- 2) Check that a specific object file that does not use a regular expression or an arbitrary object file that uses only a wildcard ("*") is specified as an object file in the link directive file. If an object file is specified by using a regular expression other than the above, the execution module may not be correctly generated. Specify an object file that does not use a regular expression or an object file that uses only a wildcard.

A.4 Error Messages

The error messages that are output when the TW850 detects information (such as occurrence of a fatal error) while it is executing processing are listed below (in the order of message numbers).

%s in an error message is determined when information (such as occurrence of a fatal error) has been detected.

E0000: Could not find the file(%s).

- 1) File %s could not be detected. Check that file %s exists.
- 2) Access to the attribute of file %s or the folder storing file %s may be limited. Check the attribute of file %s and the folder storing file %s.
- 3) Include file %s specified in the assembly source file could not be detected. Check that "search path to include file %s" is set in the project file.

E0001: Could not read/write the file.(%s)

- 1) Processing to read or write a file has been terminated abnormally. Check the contents of displayed message %s.
- 2) Access to the attribute of the file or the folder storing the file may be limited. Check the file or folder storing the file in accordance with the contents of displayed message %s.
- 3) Windows may be unstable and the file may be locked. Restart Windows in accordance with the contents of displayed message %s.

E0002: Could not open the file(%s).

- 1) Access to the attribute of file %s or the folder storing file %s may be limited. Check the attribute of file %s and the folder storing file %s.

E0003: Could not read the file(%s).

- 1) Access to the attribute of file %s or the folder storing file %s may be limited. Check the attribute of file %s and the folder storing file %s.

E0004: Could not write the file(%s).

- 1) Access to the attribute of file %s or the folder storing file %s may be limited. Check the attribute of file %s and the folder storing file %s.
- 2) The disk area necessary for writing data to file %s may be insufficient. Delete unnecessary files to secure a sufficient disk area.
- 3) Windows may be unstable and file %s may be locked. Restart Windows.

E0005: Could not delete the file(%s).

- 1) Access to the attribute of file %s or the folder storing file %s may be limited. Check the attribute of file %s and the folder storing file %s.
- 2) Windows may be unstable and file %s may be locked. Restart Windows.

E0006: The fatal error occurred.

- 1) Windows may be unstable and the file may be locked. Restart Windows.
- 2) The Java execution environment may not have been installed correctly. Install the Java execution environment (Java2 Runtime Environment Standard Edition V1.4.2_04) supported by the TW850.
- 3) The TW850 may have a bug. Contact NEC Electronics.

E0007: Could not read the project(%s).

- 1) Read processing of the PM plus project file %s has been terminated abnormally. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.

E0100: Failed to load DLL.

- 1) Windows may be unstable and the DLL file may be locked. Restart Windows.
- 2) The TW850 may not have been installed correctly. Uninstall and then reinstall the TW850.

E0103: TW is exited because failed to initialize the system.

- 1) Windows may be unstable. Restart Windows.
- 2) The Java execution environment may not have been installed correctly. Install the Java execution environment (Java2 Runtime Environment Standard Edition V1.4.2_04) supported by the TW850.
- 3) The TW850 may not have been installed correctly. Uninstall and then reinstall the TW850.

E0105: TW is exited compulsorily, because the fatal error occurred.

- 1) Windows may be unstable. Restart Windows.
- 2) The Java execution environment may not have been installed correctly. Install the Java execution environment (Java2 Runtime Environment Standard Edition V1.4.2_04) supported by the TW850.
- 3) The TW850 may not have been installed correctly. Uninstall and then reinstall the TW850.
- 4) The TW850 may have a bug. Contact NEC Electronics.

E0106: TW is exited, because the free memory is insufficient.

- 1) The heap memory area or thread stack memory area may be insufficient. Increase the size of the corresponding memory area in the [TW850 Options Dialog Box](#).
- 2) The memory area necessary for JavaVM to operate may be insufficient. Terminate unnecessary applications to secure a sufficient memory area.
- 3) Windows may be unstable. Restart Windows.

E0107: TW is exited, because the stack memory is insufficient.

- 1) The stack area necessary for the project may be insufficient. Increase the size of the corresponding stack area in the [TW850 Options Dialog Box](#).
- 2) The thread stack memory area necessary for JavaVM to operate may be insufficient. Terminate unnecessary applications to secure a sufficient memory area.
- 3) Windows may be unstable. Restart Windows.

E0200: Java runtime environment version differs from the version(%s) TW supported.

- 1) The version of the Java execution environment may not be supported by the TW850. Check that the Java execution environment version is version %s supported by the TW850, or whether that Java execution environment is used.

E0202: Could not save the TW project file(%s).

- 1) Access to the attribute of project file %s for the TW850 or the project data folder for the TW850 that stores project file %s for the TW850 may be limited. Check the attribute of project file %s for the TW850 and the project data folder for the TW850 that stores project file %s for the TW850.
- 2) The disk area necessary for saving project file %s for the TW850 may be insufficient. Delete unnecessary files to secure a sufficient disk area.

E0204: Could not delete the TW temporary file(%s).

- 1) Access to the attribute of temporary file for the TW850 %s or the folder storing temporary file for the TW850 %s may be limited. Check the attribute of temporary file for the TW850 %s and the folder storing temporary file for the TW850 %s.

E0206: Could not print.

- 1) The printer driver may not be operating correctly. Check that printing can be correctly started from Windows.

E0207: Could not start the help.

- 1) Windows may be unstable. Restart Windows.
- 2) The TW850 may not have been installed correctly. Uninstall and then reinstall the TW850.

E0208: File name is not input.

- 1) A file name may not have been input. Check that a file name has been input in the [File Selection Dialog Box](#) requiring input of a file name.

E0210: The free memory is insufficient.

- 1) The heap memory area or thread stack memory area may be insufficient. Increase the size of the corresponding memory area in the [TW850 Options Dialog Box](#).
- 2) The memory area necessary for JavaVM to operate may be insufficient. Terminate unnecessary applications to secure a sufficient memory area.
- 3) Windows may be unstable. Restart Windows.

E0211: The stack memory is insufficient.

- 1) The stack area necessary for the project may be insufficient. Increase the size of the corresponding stack area in the [TW850 Options Dialog Box](#).
- 2) The thread stack memory area necessary for JavaVM to operate may be insufficient. Terminate unnecessary applications to secure a sufficient memory area.
- 3) Windows may be unstable. Restart Windows.

E0213: The specified file is system reserved file name(%s).
Please input another file name.

- 1) File name %s may match a device name reserved for Windows. Change file name %s from a device name (such as AUX, COM1, COM2, COM3, COM4, CON, LPT1, LPT2, LPT3, LPT4, PRN, and NUL) reserved for Windows.

E0214: Failed to get the active project.

- 1) An active project may not have been set. Set an active project using the [Select Active Project...] menu of the PM plus.
- 2) Communication processing with the PM plus has been terminated abnormally. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.

E0215: Failed to restore PM plus to original state.

- 1) Check that the PM plus or ToolLinkManager may not have been started or operating correctly. Check that the PM plus or ToolLinkManager has been correctly started.
- 2) Windows may be unstable. Restart Windows.
- 3) Check that the PM plus or ToolLinkManager may not have been installed correctly. Uninstall and then reinstall the PM plus or ToolLinkManager.

E0216: Specified file (%s) is not a function transition file.

- 1) A file that is not a function transition information file is specified as a function transition information file. Specify a correct function transition information file.

E0217: Specified file (%s) is unsupported format. You must use TW version %s or later.

- 1) This file may have been saved with new-version the TW850. Open this file with the TW850 of a version that is the same as or later than the TW850 that the file was saved.

E0218: Specified file (%s) has broken.

- 1) The file may have been destroyed or edited. Specify a correct function transition information file.

E0300: Failed to calculate the profile table size.

- 1) Size calculation processing of the profile table has been terminated abnormally. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.

E0301: Failed to build or get information of the execution module for original analysis.

- 1) An illegal project may have been specified. Check that the project can correctly generate an execution module for original analysis.
- 2) A C compiler package of a version not supported by the TW850 may be being used. Check that the version is supported by the TW850.

E0303: There is no analyzable function with TW.

- 1) An illegal link directive file may have been specified. The TW850 cannot recognize a link directive file for which no function is assigned to the TEXT segment or .text section, as subject to performance analysis or tuning. Specify a source file that functions subject to performance analysis or tuning for a specific segment or section exist, so that the TW850 can perform analysis. Or, create a link directive that maps functions subject to performance analysis or tuning for a specific segment or section.
- 2) A C source file or assembly source file in an illegal format may have been specified. Change the contents of the C source file or assembly source file to a format that the C compiler package supported by the TW850 can support.

E0304: Failed to execute performance analysis.

- 1) An illegal project may have been specified. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.

E0305: Failed to execute tuning.

- 1) An illegal project may have been specified. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.

E0306: Failed to open the tuning wizard.

- 1) An illegal project may have been specified. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.

E0308: Failed to restore the tuned result.

- 1) Access to the attribute of project data folder for the TW850 that stores project file for the TW850 may be limited. Check the attribute of project data folder for the TW850 that stores project file for the TW850.
- 2) The disk area necessary for restoring the tuning result may be insufficient. Delete unnecessary files to secure a sufficient disk area.

E0309: Failed to save the temporary file to %s.

- 1) Access to the attribute of the temporary file or the folder storing the temporary file may be limited. Check the attribute of the temporary file and the folder storing the temporary file.
- 2) The disk area necessary for saving the temporary file may be insufficient. Delete unnecessary files to secure a sufficient disk area.

E0310: Failed to restore the temporary file(%s).

- 1) Access to the attribute of temporary file %s or the folder storing temporary file %s may be limited. Check the attribute of temporary file %s and the folder storing temporary file %s.

E0311: Failed to delete the temporary file(%s).

- 1) Access to the attribute of temporary file %s or the folder storing temporary file %s may be limited. Check the attribute of temporary file %s and the folder storing temporary file %s.

E0314: The running debugger(%s) was not specified in the project.

- 1) Debugger %s undefined by a project (e.g., the SM850 if the ID850 is specified by the project) may have been started. The TW850 cannot execute correct processing if an undefined debugger is started.

E0400: Failed to build the execution module for original analysis.

- 1) Generation processing of an execution module for original analysis has been terminated abnormally. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.
- 2) A C compiler package of a version not supported by the TW850 may be being used. Check that the version is supported by the TW850.

E0401: Failed to build the execution module for profiling.

- 1) Generation processing of an execution module for profile has been terminated abnormally. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.
- 2) The memory area necessary for inserting a probe code or saving a profile table may be insufficient. Change the specified link directive file to secure sufficient memory space.
- 3) A C compiler package of a version not supported by the TW850 may be being used. Check that the version is supported by the TW850.

E0402: Failed to build the tuned execution module.

- 1) Generation processing of a tuned execution module has been terminated abnormally. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.
- 2) The memory area necessary for relocating a function subject to tuning may be insufficient. Change the specified link directive file to secure sufficient memory space.
- 3) A C compiler package of a version not supported by the TW850 may be being used. Check that the version is supported by the TW850.

E0403: Failed to generate an assembly source file.

- 1) Generation processing of an assembly source file has been terminated abnormally. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.
- 2) A C compiler package of a version not supported by the TW850 may be being used. Check that the version is supported by the TW850.

E0404: Failed to analyze callpair information.

- 1) Analysis processing of callpair information has been terminated abnormally. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.

E0405: Failed to execute the execution module for original analysis.

- 1) Execution processing of an execution module for original analysis has been terminated abnormally. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.

E0406: Failed to execute the execution module for profiling.

- 1) Execution processing of an execution module for profile has been terminated abnormally. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.
- 2) An illegal execution module for profile may have been output. Check whether a message (such as a warning message) is displayed when generating a tuned execution module.
- 3) An assembly source file in an illegal format may have been specified. Change the contents of the assembly source file to a format that the C compiler package supported by the TW850 can support.

E0407: Failed to execute the tuned execution module.

- 1) Execution processing of a tuned execution module has been terminated abnormally. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.
- 2) An illegal tuned execution module may have been output. Check whether a message (such as a warning message) is displayed when generating a tuned execution module.
- 3) An assembly source file in an illegal format may have been specified. Change the contents of the assembly source file to a format that the C compiler package supported by the TW850 can support.

E0408: Failed to analyze function information.

- 1) Analysis processing of function information has been terminated abnormally. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.

E0409: Failed to tune an execution module.

- 1) Tuning processing of an execution module has been terminated abnormally. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.

E0411: Failed to analyze the assembly source file.

- 1) Analyzing the assembly source file has failed. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.

E0412: Failed to analyze the execution module for original analysis.

- 1) The execution module for original analysis could not be analyzed. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.

E0500: The number of analyzable functions exceeded limited number(%s).

- 1) The number of functions subject to performance analysis/tuning may exceed the maximum (%s). Decrease the total number of functions subject to performance analysis/tuning to %s or less by assigning functions that do not have to be analyzed or tuned to other segments or sections from the TEXT segment or .text section.

E0501: The format of the system library information file(%s) is illegal.

- 1) The contents of system library information file %s may have been destroyed. Restore the system library information file to the original status by uninstalling and then reinstalling the TW850.
- 2) The version of system library information file %s may not be supported by the TW850. Check that the version of system library information file %s is supported by the TW850.

E0502: Mismatching data exist in the system library information file(%s).

- 1) The contents of system library information file %s may have been destroyed. Restore the system library information file to the original status by uninstalling and then reinstalling the TW850.

E0503: Syntax error or arithmetic error.(%s)

- 1) The assembly source file may have a description of an illegal syntax/evaluation expression. Change corresponding location %s of the assembly source file to a syntax/evaluation expression that the C compiler package supported by the TW850 can support.

E0504: Too many pseudo instructions to terminate conditional assemble.(%s)

- 1) The assembly source file may not have a description of a conditional assemble start quasi directive that corresponds to a conditional assemble end quasi directive. Add a description of a conditional assemble start quasi directive to corresponding location %s of the assembly source file, or delete the description at the corresponding location.

E0505: The system reservation macro name(%s) is used.

- 1) Macro name %s may match a name of system-reserved macro reserved for the TW850. Change macro name %s from a name of system-reserved macro reserved for the TW850.

E0506: Could not evaluate because the unsolved symbol is in the expression.(%s)

- 1) Static analysis processing of symbol information has been terminated abnormally. Define the corresponding symbol included in the evaluation expression of assembly source file %s.

E0507: The execution module is not read.

- 1) Access to the attribute of the execution module or the folder storing the execution module may be limited. Check the attribute of the execution module and the folder storing the execution module.

E0508: Could not get sufficient data from the profile memory.

- 1) The memory area necessary as a profile memory area may be insufficient. In the link directive file, increase the size of the segment or section to which the profile memory area is assigned.
- 2) Data of the profile memory area cannot be read from the execution engine (such as the debugger or simulator) because the file generated by the TW850 has been manipulated (moved, deleted, or attribute changed). Restore the file generated by the TW850 to the original status by re-executing performance analysis processing/tuning processing.

E0509: The symbol(%s) is not found in the execution module.

- 1) Detection processing of symbol %s has been terminated abnormally because the execution module generated by the TW850 has been manipulated (moved, deleted, or attribute changed). Restore the execution module generated by the TW850 to the original status by re-executing performance analysis processing/tuning processing.
- 2) A non-function symbol of the same name as the corresponding function symbol exists in the execution module. If the symbol can be built in the execution module by a link directive, the symbol name built in the execution module by a link directive must not be the same as a function name. If there is a description to declare a symbol of the same name as a symbol in the program by a link directive (for example, the linker script command PROVIDE of GCC), however, this error can be avoided by describing using that description method.

E0510: The label(%s) is defined multiply.(%s)

- 1) Two or more of the same label name %s may have been defined in the assembly source file. Correct the corresponding label of assembly source file %s.

E0511: Too many pseudo instructions to terminate macro definition or repetition.(%s)

- 1) The assembly source file may not have a description of a macro definition/repeat start quasi directive that corresponds to a definition/repeat end quasi directive. Add a description of a macro definition/repeat start quasi directive to corresponding location %s of the assembly source file, or delete the description at the corresponding location.

E0512: The number of macro nesting exceeded limited level(%s).(%)

- 1) The number of times of nest expansion of the macro may exceed the limit %s. Change the number of times of nest expansion of the macro of assembly source file %s to the limit %s or less.

E0513: The instruction to start makes no match of the instruction to terminate about macro definition or repetition.(%s)

- 1) The assembly source file may have mismatching between a macro definition/repeat start quasi directive and a macro definition/repeat end quasi directive. Change assembly source file %s to match the macro definition/repeat start quasi directive and macro definition/repeat end quasi directive.

E0514: Used macro name is same as an assembler instruction(%s).

- 1) The macro name matching an assembler instruction mnemonic cannot be used. Change the macro name that is different from an assembler instruction mnemonic, or exclude the source file using that macro name from analysis.

E0600: Tuning was aborted, because illegal data was found in the profile information.

- 1) The contents of the profile information necessary for generating a tuned execution module may have been destroyed. Check that performance analysis of an execution module for original analysis has been executed correctly.

E0700: Failed to save the result of tuning.(%s)

- 1) Access to the attribute of the tuned execution module or the folder storing the tuned execution module may be limited. Check the attribute of the tuned execution module and the folder storing the tuned execution module in accordance with the contents of displayed message %s.

E0701: Failed to restore the result of tuning.(%s)

- 1) Access to the attribute of the tuned execution module or the folder storing the tuned execution module may be limited. Check the attribute of the tuned execution module and the folder storing the tuned execution module in accordance with the contents of displayed message %s.

E0705: Failed to generate TW project backup file(%s).

- 1) Access to a project file for the TW850, or the folder storing the file, and write a project backup file for the TW850 may be permitted. Check the attribute of the project file or the folder storing the file, and the project backup file.
- 2) The disk area necessary for saving the project file may be insufficient. Allocate sufficient disk area and perform the operation again.

E0706: Failed to restore from the TW project backup file(%s).

- 1) Access to a project backup file for the TW850 may be permitted. Check the attribute of the project backup file. If recovery from the project backup file for the TW850 has failed, information will be recovered from project file for the TW850.
- 2) The project backup file for the TW850 may have been destroyed, or an incompatible file may have been read. Delete the file. If recovery from the project backup file for the TW850 has failed, information will be recovered from project file for the TW850.

E0802: Found the option(%s) which is not supported.

- 1) Option %s not supported by the TW850 may have been specified. Delete the specification of option %s from the project.

E0803: Failed to access to the project file folder(%s).

- 1) Access to the attribute of folder %s storing project file may be limited. Check the attribute of folder %s storing the project file.

E0804: Failed to read the TW project file(%s).

- 1) Access to the attribute of project file %s for the TW850 or the project data folder for the TW850 that stores project file %s for the TW850 may be limited. Check the attribute of project file %s for the TW850 and the project data folder for the TW850 storing project file %s.
- 2) The contents of project file for the TW850 %s may have been destroyed. Check that the project file can be correctly read.

E0805: Failed to write the TW project file(%s).

- 1) Access to the attribute of project file %s for the TW850 or the project data folder for the TW850 that stores project file %s for the TW850 may be limited. Check the attribute of project file %s for the TW850 and the project data folder for the TW850 storing project file %s.

E0806: Failed to create the TW project data folder(%s).

- 1) Access to the attribute of the folder storing the project data folder for the TW850 may be limited. Check the attribute of the folder storing the project data folder for the TW850.
- 2) The existing folder name may match project data folder name for the TW850 %s. Change the existing folder name from project data folder %s for the TW850.

E0807: Failed to delete the TW temporary file(%s).

- 1) Access to the attribute of temporary file for the TW850 %s or the folder storing temporary file for the TW850 %s may be limited. Check the attribute of temporary file for the TW850 %s and the folder storing temporary file for the TW850 %s.

E0808: Illegal project file(%s).

- 1) The contents of project file %s may have been destroyed. Check that the project file can correctly generate an execution module.

E0810: A link directive file is not specified.

- 1) An illegal project file may have been specified. The TW850 cannot recognize a project file for which a link directive file has not been specified, as subject to performance analysis or tuning.

E0811: Unknown execution engine.

- 1) The execution engine (such as the debugger or simulator) could not be detected. Check that the execution engine supported by the TW850 is set using the [Debugger Settings...] menu of the PM plus.

E0812: Environment variable(%s) is not set.

- 1) Environmental variable %s may not have been described. Check that %s has been set as an environmental variable of the host machine in an environmental variable setting file (such as autoexec.bat).

E0816: The active project is not supported by TW.
%s

- 1) An illegal project (active project) may have been specified. The TW850 cannot recognize a project for which a C source file and an assembly source file have not been specified, as subject to performance analysis or tuning.
- 2) The type of the active project may not be supported by the TW850. Check that the type of the active project is supported by the TW850.

E0817: There is no information(%s) in the project file(%s).

- 1) The TW850 cannot detect information %s necessary for the TW850 to execute performance analysis or tuning processing from project file %s. Set information %s necessary for the TW850 to execute performance analysis or tuning processing by using the PM plus.

E0818: There is no project information about debugger.

- 1) Information on the debugger could not be detected from the project file. Save the debug environment for the project file by using the debugger.

E0819: There is no specification about debugger.

- 1) A debugger may not have been set. Set a debugger using the [Debugger Settings...] menu of the PM plus.

E0820: Source file is not registered.

- 1) The TW850 cannot detect the information (source file name) necessary for the TW850 to execute performance analysis or tuning processing from project file. Set the source file name necessary for the TW850 to execute performance analysis or tuning processing by using the PM plus.

E0821: Debug target file is not specified.

- 1) The TW850 cannot detect the information (debug target file name) necessary for the TW850 to execute performance analysis or tuning processing from project file. Set the output file name by using a linker option of the PM plus, and then execute processing to save a workspace.

E0824: Failed to get default include path for compiler.

- 1) Other tool (the PM plus/CA850) may not have been installed correctly. Uninstall and then reinstall other tool.
- 2) Windows may be unstable. Restart Windows.

E0900: The execution module(%s) is unsupported format.

- 1) Execution module %s in an illegal format may have been specified. Check that execution module %s has been generated by a C compiler package supported by the TW850.

E0901: Could not convert from the line number(%s) to the address.

- 1) Illegal line number %s may have been specified. The TW850 cannot recognize a line number that does not exist or a line number outside a function, which cannot be converted into an address, as subject to performance analysis or tuning.

E0902: Could not convert from the address information(%s) to the address.

- 1) Illegal symbol number %s may have been specified. The TW850 cannot recognize a symbol that does not exist, as subject to performance analysis or tuning.

E1001: Illegal device file(%s).

- 1) Device file %s in an illegal format may have been specified. Check that the device file can correctly generate an execution module.
- 2) The device file may not have been installed correctly. Uninstall and then reinstall the device file.

E1100: Not exist string constant sign(%s) which makes a pair.

- 1) The assembly source file may not have a description of an end symbol condition to declare a character string constant. Add a description of an end signal to corresponding location %s of the assembly source file, or delete the description at the corresponding location.

E1101: Illegal operand which is not convertible to a string.

- 1) The assembly source file may have a description of an illegal character string constant in an operand that declares a character string constant. Change the corresponding location of the assembly source file to "a declaration of a character string constant" that the C compiler package supported by the TW850 can support.

E1102: Could not to evaluate the expression.

- 1) The assembly source file may have a description of an illegal evaluation expression. Change the corresponding location of the assembly source file to an evaluation expression that the C compiler package supported by the TW850 can support.

E1103: The macro definition does not have the declaration of a macro name.

- 1) The assembly source file may not have a description of a macro name in a macro definition. Change the corresponding location of the assembly source file to a macro name that the C compiler package supported by the TW850 can support.

E1104: The number of arguments at the use of the macro is mismatch.

- 1) The number of arguments when a macro is defined may have been different from the number of arguments when the macro is used in the assembly source file. Correct the corresponding location of the assembly source file so that the number of arguments is the same.

E1105: The specification of the file symbol is illegal.

- 1) The assembly source file may have a description of an illegal file symbol declaration. Check that the file symbol declaration in the assembly source file is described correctly.

E1200: Not exist bracket which makes a pair.

- 1) The assembly source file may have mismatching parentheses in the evaluation expression. Change the assembly source file to match the parentheses included in the evaluation expression.

E1201: Not exist an expression to evaluate.

- 1) The assembly source file may have a description of an illegal evaluation expression. Change the corresponding location of the assembly source file to an evaluation expression that the C compiler package supported by the TW850 can support.

E1202: Could not evaluate the numerical expression(%s).

- 1) The assembly source file may have a description of an illegal evaluation expression including illegal numeric expression %s. Change the corresponding location of the assembly source file to a numeric expression that the C compiler package supported by the TW850 can support.

E1203: Could not evaluate the symbol(%s).

- 1) The assembly source file may have a description of an illegal evaluation expression including illegal symbol %s. The TW850 cannot evaluate a symbol whose value is not defined.
- 2) The assembly source file may have a description of an illegal evaluation expression including label %s. The TW850 cannot evaluate a label described in an evaluation expression in the assembly source file.

E1204: The number of operations makes no match of the number of numerical values.

- 1) The assembly source file may have a description of an illegal evaluation expression including an operator with an illegal number of items. Change the corresponding location of the assembly source file to an evaluation expression that the C compiler package supported by the TW850 can support.

E1205: Illegal usage of the operator(%s).

- 1) The assembly source file may have a description of an illegal evaluation expression including an illegal operator. Change the corresponding location of the assembly source file to an evaluation expression that the C compiler package supported by the TW850 can support.

E1206: Divided by zero in the expression.

- 1) The assembly source file may have a description of an illegal evaluation expression (division by zero). Change the corresponding location of the assembly source file to an evaluation expression that the C compiler package supported by the TW850 can support.

E1300: Failed to build the execution module.(%s)

- 1) Generation processing of an execution module for original analysis has been terminated abnormally. Individually start the build tool from the original project file in accordance with the contents of displayed message %s, and check that generation processing of an execution module for original analysis has been executed correctly.
- 2) Generation processing of an execution module for profile has been terminated abnormally. Individually start the build tool from the project file of an execution module for profile in accordance with the contents of displayed message %s, and check that generation processing of an execution module for profile has been executed correctly.
- 3) Generation processing of a tuned execution module has been terminated abnormally. Individually start the build tool from the project file of a tuned execution module in accordance with the contents of displayed message %s, and check that generation processing of a tuned execution module has been executed correctly.
- 4) A link directive file in an illegal format may have been specified. Check that the link directive file can correctly generate an execution module in accordance with the contents of displayed message %s.

E1302: Failed to delete temporary files for build.(%s)

- 1) Access to the attribute of the temporary file or the folder storing the temporary file may be limited. Check the attribute of the temporary file and the folder storing the temporary file in accordance with the contents of displayed message %s.
- 2) The temporary file may have been referenced from another application (such as the editor). Check that the temporary file is referenced by another application in accordance with the contents of displayed message %s.

E1303: Found the unsupported symbol in the execution module.

- 1) An illegal execution module may have been specified. The TW850 cannot recognize an execution module including a monitor program for system performance analyzer AZ850, as subject to performance analysis or tuning.

E1400: Failed to execute the execution module(%s).

- 1) Execution processing of execution module %s has been terminated abnormally. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method. Also refer to "4.6.3 [Description format of assembly source file](#)".
- 2) An illegal project may have been specified. Individually start the debugger from the original project file, and check that execution processing in the specified range has been executed correctly.
- 3) Assignment of a relocation function, profile routine, or profile table may be incorrect. Check that the relocation function, profile routine, or profile table is assigned to the memory area specified on the [Allocation Destination Selection Panel](#).

E1401: Failed to stop executing the execution module(%s).

- 1) Execution stop processing of execution module %s has been terminated abnormally. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.

E1403: Failed to analyze the profile information.

- 1) The TW850 may have a bug. Contact NEC Electronics.

E1404: Failed to analyze the trace data.

- 1) Trace data could not be detected. Click the <Detail>>> button on the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.

E1407: Error occurred while executing process of TW timer initialization.

- 1) Execution of the timer initialization processing for the TW850 has been stopped unexpectedly. A specification not supported by the TW850 may have been made for the tw_init_timer processing or tw_init_timer_timing_address of the timer control code block. Refer to the User's Manual to check that the operation is performed within a range supported by the TW850.
- 2) The allocation destination of the profile routine or profile table may be incorrect. Refer to the User's Manual to check that the specification is made within a range supported by the TW850.
- 3) The target system may be unstable. Check the target system, reset it, and then execute again.
- 4) The debugger may be unstable. Check that the execution module to be analyzed can be correctly executed with the debugger.
- 5) Windows may be unstable. Restart Windows.

E1408: Error occurred while executing from start execution position to start analysis position.

- 1) Execution from the execution start position to analysis start position has been stopped unexpectedly. A part not supported by the TW850 may have been specified as an execution start position, or an analysis function/a analysis start position/a trace on event. Refer to the User's Manual to check that the part is specified within a range supported by the TW850.
- 2) The allocation destination of the profile routine or profile table may be incorrect. Refer to the User's Manual to check that the specification is made within a range supported by the TW850.
- 3) The target system may be unstable. Check the target system, reset it, and then execute again.
- 4) The debugger may be unstable. Check that the execution module to be analyzed can be correctly executed with the debugger.
- 5) Windows may be unstable. Restart Windows.

E1409: Error occurred while executing from start analysis position to end analysis position.

- 1) Execution from the analysis start position to analysis end position has been stopped unexpectedly. A part not supported by the TW850 may have been specified as an analysis function/a analysis start position/a trace on event, or a analysis end position/a trace off event. Refer to the User's Manual to check that the part is specified within a range supported by the TW850.
- 2) The allocation destination of the profile routine or profile table may be incorrect. Refer to the User's Manual to check that the specification is made within a range supported by the TW850.
- 3) The target system may be unstable. Check the target system, reset it, and then execute again.
- 4) The debugger may be unstable. Check that the execution module to be analyzed can be correctly executed with the debugger.
- 5) Windows may be unstable. Restart Windows.

E1410: Error occurred while executing from start execution position to timer initialization position(%s).

- 1) Execution from the execution start position to timer initialization position has been stopped unexpectedly. A part not supported by the TW850 may have been specified as an execution start position. Refer to the User's Manual to check that the specification is made within a range supported by the TW850.
- 2) If the timer initialization position is specified in timer control code block, there may be errors in settings. Refer to the User's Manual to check that the specification is made within a range supported by the TW850.
- 3) The allocation destination of the profile routine or profile table may be incorrect. Refer to the User's Manual to check that the specification is made within a range supported by the TW850.
- 4) The target system may be unstable. Check the target system, reset it, and then execute again.
- 5) The debugger may be unstable. Check that the execution module to be analyzed can be correctly executed with the debugger.
- 6) Windows may be unstable. Restart Windows.

E1411: Error occurred while executing from timer initialization position(%s) to start analysis position.

- 1) Execution from the timer initialization position to analysis start position has been stopped unexpectedly. A part not supported by the TW850 may have been specified as an analysis function/a analysis start position. Refer to the User's Manual to check that the specification is made within a range supported by the TW850.
- 2) If the timer initialization position is specified in timer control code block, there may be errors in settings. Refer to the User's Manual to check that the specification is made within a range supported by the TW850.
- 3) The target system may be unstable. Check the target system, reset it, and then execute again.
- 4) The debugger may be unstable. Check that the execution module to be analyzed can be correctly executed with the debugger.
- 5) Windows may be unstable. Restart Windows.

E1412: Error occurred while executing process of TW timer prologue.

- 1) Execution of the time profile preprocessing for the TW850 has been stopped unexpectedly. A part not supported by the TW850 may have been specified as an analysis function/a analysis start position or an analysis end position. Refer to the User's Manual to check that the specification is made within a range supported by the TW850.
- 2) If the timer initialization position is specified in timer control code block, there may be errors in settings. Refer to the User's Manual to check that the specification is made within a range supported by the TW850.
- 3) The allocation destination of the profile routine or profile table may be incorrect. Refer to the User's Manual to check that the specification is made within a range supported by the TW850.
- 4) The target system may be unstable. Check the target system, reset it, and then execute again.
- 5) The debugger may be unstable. Check that the execution module to be analyzed can be correctly executed with the debugger.
- 6) Windows may be unstable. Restart Windows.

E1413: Error occurred while executing process of TW timer epilogue.

- 1) Execution of the time profile post-processing for the TW850 has been stopped unexpectedly. A part not supported by the TW850 may have been specified as an analysis function/a analysis start position or an analysis end position. Refer to the User's Manual to check that the specification is made within a range supported by the TW850.
- 2) If the timer initialization position is specified in timer control code block, there may be errors in settings. Refer to the User's Manual to check that the specification is made within a range supported by the TW850.
- 3) The allocation destination of the profile routine or profile table may be incorrect. Refer to the User's Manual to check that the specification is made within a range supported by the TW850.
- 4) The target system may be unstable. Check the target system, reset it, and then execute again.
- 5) The debugger may be unstable. Check that the execution module to be analyzed can be correctly executed with the debugger.
- 6) Windows may be unstable. Restart Windows.

E1414: Error occurred while executing from end analysis position to start analysis position.

- 1) Execution from the analysis end position to analysis start position has been stopped unexpectedly. A part not supported by the TW850 may have been specified as an analysis a trace on event, or an analysis end position/a trace off event. Or, there may be errors in settings of the number of times of trace. Refer to the User's Manual to check that the specification is made within a range supported by the TW850.
- 2) The target system may be unstable. Check the target system, reset it, and then execute again.
- 3) The debugger may be unstable. Check that the execution module to be analyzed can be correctly executed with the debugger.
- 4) Windows may be unstable. Restart Windows.

E1500: Illegal link directive file(%s).

- 1) Analysis processing of link directive file %s has been terminated abnormally. Individually start the build tool from the original project file, and check that generation processing of an execution module has been executed correctly.
- 2) Access to the attribute of link directive file %s or the folder storing link directive file %s may be limited. Check the attribute of link directive file %s and the folder storing link directive file %s.
- 3) The link directive file may have been described in a format not supported by the TW850. Check that the version of the C compiler package is supported by the TW850. Also check notes on the link directive file by referring to the release note.

E1501: Failed to generate the link directive file.

- 1) Access to the attribute of the project data folder for the TW850 storing the link directive file may be limited. Check the attribute of project data folder for the TW850 that stores the link directive file.
- 2) A link directive file in an illegal format may have been specified. Check that the link directive file can correctly generate an execution module.

E1502: Could not get link directive information to create new link directive file.

- 1) Acquiring the information for generating a new link directive file has failed. Check whether the execution module to be generated is in the relocatable format, in which a link directive file cannot be created. The TW850 cannot generate a new link directive file from a project that generates a relocatable format. Change the project to the one that generates an execution module whose addresses are resolved.

E1503: Could not find section mapping option(%s).

- 1) An option that specifies section allocation must be specified in the link directive file. Specify an option that specifies section allocation.

E1504: Multiple specified section mapping option(%s).(%)s)

- 1) Two or more options that specify section allocation must be specified in the link directive file. Specify only one option that specifies section allocation.

E1505: Illegal start of (%s) area with the string(%s).(%)s)

- 1) A syntax error has been found in the link directive file. Correct a syntax error.

E1506: Illegal usage of the symbol(%s).(%)s)

- 1) A syntax error has been found in the link directive file. Correct a syntax error.

E1507: Illegal parameter(%s).(%)s)

- 1) A syntax error has been found in the link directive file. Correct a syntax error.

E1508: Could not convert to number from the parameter(%s).(%)s)

- 1) A syntax that specifies a character that cannot be converted into a numeral has been found in the link directive file. Describe so that the character can be converted into a numeral.

E1602: Failed to download the execution module(%s) to the target.

- 1) Downloading processing of an execution module has been terminated abnormally. Check that the mapping size of the execution module has been specified correctly.
- 2) Access to the attribute of execution module %s or the folder storing execution module %s may be limited. Check the attribute of execution module %s and the folder storing execution module %s.
- 3) Execution module %s may have been downloaded to a memory to which access is limited. Check that allocation of the memory area to which the execution module is downloaded and setting of the peripheral I/O register is performed on the RAM area.
- 4) Execution module %s in an illegal format may have been specified. Check that execution module %s has been generated by a C compiler package supported by the TW850.
- 5) The version of other tool may not be supported by the TW850. Check that the version of other tool is supported by the TW850.
- 6) Other tool may not be operating correctly. Restart the other tool.
- 7) Windows may be unstable. Restart Windows.

E1603: Failed to read the register(%s) value.

- 1) The version of other tool may not be supported by the TW850. Check that the version of other tool is supported by the TW850.
- 2) Other tool may not be operating correctly. Restart the other tool.
- 3) Windows may be unstable. Restart Windows.

E1604: Failed to write a register(%s) value.

- 1) The version of other tool may not be supported by the TW850. Check that the version of other tool is supported by the TW850.
- 2) Other tool may not be operating correctly. Restart the other tool.
- 3) Windows may be unstable. Restart Windows.

E1605: Failed to read from memory(%s).

- 1) Read processing may have been performed on a memory to which access is limited. Check that allocation of the memory area on which read processing is to be performed and setting of the peripheral I/O register is performed on the RAM area.
- 2) The version of other tool may not be supported by the TW850. Check that the version of other tool is supported by the TW850.
- 3) Other tool may not be operating correctly. Restart the other tool.
- 4) Windows may be unstable. Restart Windows.

E1606: Failed to write to memory(%s).

- 1) Write processing may have been performed on a memory to which access is limited. Check that allocation of the memory area on which write processing is to be performed and setting of the peripheral I/O register is performed on the RAM area.
- 2) The version of other tool may not be supported by the TW850. Check that the version of other tool is supported by the TW850.
- 3) Other tool may not be operating correctly. Restart the other tool.
- 4) Windows may be unstable. Restart Windows.

E1607: Failed to fill to memory(%s).

- 1) Fill processing may have been performed on a memory to which access is limited. Check that allocation of the memory area on which fill processing is to be performed and setting of the peripheral I/O register is performed on the RAM area.
- 2) The version of other tool may not be supported by the TW850. Check that the version of other tool is supported by the TW850.
- 3) Other tool may not be operating correctly. Restart the other tool.
- 4) Windows may be unstable. Restart Windows.

E1608: Failed to cooperate with the other tools because of the following reason.
%s

- 1) The version of other tool may not be supported by the TW850. Check that the version of other tool is supported by the TW850.
- 2) Other tool may not be operating correctly. Restart the other tool.
- 3) Windows may be unstable. Restart Windows.

E1631: Failed to read the project file(%s).

- 1) Access to the attribute of project file %s or the folder storing project file %s may be limited. Check the attribute of project file %s and the folder storing project file %s.
- 2) The version of other tool may not be supported by the TW850. Check that the version of other tool is supported by the TW850.
- 3) Other tool may not be operating correctly. Restart the other tool.
- 4) Windows may be unstable. Restart Windows.

E1632: Failed to get the current project file name.

- 1) The version of other tool may not be supported by the TW850. Check that the version of other tool is supported by the TW850.
- 2) Other tool may not be operating correctly. Restart the other tool.
- 3) Windows may be unstable. Restart Windows.

E1636: Failed to open the help file (%s).

- 1) Help file %s could not be detected. Check that help file %s exists.
- 2) The TW850 may not have been installed correctly. Uninstall and then reinstall the TW850.

E1637: Failed to get the list of registry value.(%s)

- 1) Windows may be unstable. Restart Windows.
- 2) The TW850 or other tool may not have been installed correctly. Uninstall and then reinstall the TW850 or the other tool.

E1638: Failed to get the string of the registry value.(%s)

- 1) Windows may be unstable. Restart Windows.
- 2) The TW850 or other tool may not have been installed correctly. Uninstall and then reinstall the TW850 or the other tool.

E1639: Failed to get the numerical value of the registry value.(%s)

- 1) Windows may be unstable. Restart Windows.
- 2) The TW850 or other tool may not have been installed correctly. Uninstall and then reinstall the TW850 or the other tool.

E1640: Failed to execute the external command.

- 1) Starting processing of the external command has been terminated abnormally. Click the <Detail>>> button in the [Error Dialog Box](#) to check the cause of the occurrence and corrective method.

E1646: Could not start the debugger within the specific time(%s sec).

- 1) The debugger does not become operable within the specified time. Change the timeout value %s specified by the TW850 start option -Dtw.DebuggerTimeout to a proper value.
- 2) The debugger may not be operating correctly. Individually start the debugger and check that the debugger operates normally.
- 3) A debugger of a version not supported by the TW850 may have been specified. Check the version of the debugger.
- 4) Windows may be unstable. Restart Windows in accordance with the contents of displayed message %s.

E1700: Failed to execute the external command.(%s)

- 1) Starting processing of the external command has been terminated abnormally. Check that corresponding external command %s has been correctly started when it has been started from command prompt.
- 2) The external command could not be detected. Check that "search path to corresponding external command %s" has been set as an environmental variable of the host machine in an environmental variable setting file (such as autoexec.bat).
- 3) The memory area necessary for the external command to operate may be insufficient. Terminate unnecessary applications to secure a sufficient memory area.

E1801: The device information(%s) is not set in registry.

- 1) The device file may not have been installed correctly. Uninstall and then reinstall the device file.

E1803: Could not find the device file(%s).

- 1) Device file %s could not be detected. Check that "search path to device file %s" has been set as an environmental variable of the host machine in an environmental variable setting file (such as autoexec.bat).
- 2) Access to the attribute of device file %s or the folder storing device file %s may be limited. Check the attribute of device file %s and the folder storing device file %s.

E1804: The device information is not set at start up.

- 1) The device when the debugger is not specified was not correctly specified. Specify a correct device on the [TW850 Options Dialog Box](#). When a device file is used, check that the device file has been correctly installed.

E1900: Can not connect to ToolLinkManager.

- 1) The TW850 may be unstable. Restart the TW850.
- 2) Windows may be unstable. Restart Windows.
- 3) The ToolLinkManager may not have been installed correctly. Uninstall and then reinstall the ToolLinkManager.

E1901: ToolLinkManager is not running.

- 1) The ToolLinkManager may not be operating correctly. Terminate the NEC Electronics tool (the PM plus, ID850, ID850NW, SM850, SM plus for V850, or TW850), and restart the ToolLinkManager.

E9000: Could not start TW.

- 1) The Java execution environment may not have been installed correctly. Install the Java execution environment (Java2 Runtime Environment Standard Edition V1.4.2_04) supported by the TW850.
- 2) The Java execution environment could not be detected. Check that "search path to Java execution environment" has been set as an environmental variable of the host machine in an environmental variable setting file (such as autoexec.bat).

E9001: Invalid options(%s).

- 1) A project file could not be detected. Check that a project file exists.
- 2) Access to the attribute of the project file or the folder storing the project file may be limited. Check the attribute of the project file and the folder storing the project file.
- 3) An option not supported by the TW850 may have been specified for a project. Delete the specification of option %s from the project.

E9002: The fatal error occurred. Could not start TW.

- 1) The memory area necessary for JavaVM to operate may be insufficient. Terminate unnecessary applications to secure a sufficient memory area.
- 2) Windows may be unstable. Restart Windows.

E9003: Could not find the Java application launcher(%s).

- 1) The Java execution environment may not have been installed correctly. Install the Java execution environment (Java2 Runtime Environment Standard Edition V1.4.2_04) supported by the TW850.
- 2) Java application start command %s could not be detected. Check that "search path to Java application start command" has been set as an environmental variable of the host machine in an environmental variable setting file (such as autoexec.bat).
- 3) Windows may be unstable. Restart Windows.

E9004: Illegal TW options or the fatal error occurred while executing Java.

- 1) If an option is specified as other option of the start options of the TW850, there may be errors in settings of the option. Refer to "[3.1.2 Start options](#)" and the User's Manual to check that the specification of the option is correct.
- 2) The other tools that operate in conjunction with the TW850 may not be operating correctly. Terminate the tools once and then restart them.
- 3) Windows may be unstable. Restart Windows.
- 4) Java may not be operating correctly. Uninstall and then reinstall Java. Check that the version of Java is supported by the TW850.
- 5) The other tools that operate in conjunction with the TW850 may not have been installed correctly. Uninstall and then reinstall other tool. Check that the version of other tools that operate in conjunction is supported by the TW850.

E9100: Failed to copy the file from %s to %s.

- 1) Copy source file %s could not be detected. Check that copy source file %s exists.
- 2) Access to the attribute of file %s or the folder storing file %s may be limited. Check the attribute of file %s and the folder storing file %s.
- 3) The disk area necessary for copying the file may be insufficient. Delete unnecessary files to secure a sufficient disk area.

E9101: Failed to check the wrapper command(%s).

- 1) Access to the attribute of the wrapper command AS850.EXE for the assembler supplied by the TW850 may be limited. Check the attribute of wrapper command AS850.EXE for the assembler supplied by the TW850.
- 2) Access to the attribute of wrapper command %s or the folder storing wrapper command %s may be limited. Check the attribute of wrapper command %s and the folder storing wrapper command %s.
- 3) The memory area necessary for wrapper command %s to operate may be insufficient. Terminate unnecessary applications to secure a sufficient memory area.

E9102: Failed to access to the registry.

- 1) Access to the attribute of registry information may be limited. Have a user with Administrator privileges start the TW850.
- 2) Windows may be unstable. Restart Windows.

E9103: Illegal TWCommon.dll or registry.

- 1) Windows may be unstable. Restart Windows.
- 2) The external command common library TWCommon.dll supplied by the TW850 or the contents of the registry information may have been destroyed. Uninstall and then reinstall the TW850.

APPENDIX B RESTRICTION AND LIMIT

This appendix explains restrictions and limits of the TW850.

B.1 Restriction

Performance analysis processing and tuning processing cannot be executed if the restrictions of the TW850 are exceeded.

[Table B-1](#) shows the restrictions of the TW850.

Table B-1 Restrictions of TW850

Item	Restriction
Number of user functions	64,511
Total of number of callpairs of user functions and number of callpairs of system library functions	2,147,483,647
Nesting of macros in an assembly source file	32 stages

B.2 Limit

[Table B-2](#) shows the limits of the TW850.

Table B-2 Limits of TW850

Item	Limit
Number of times of calling callpair (Trace data analysis method)	9,223,372,036,854,775,807
Number of times of calling callpair (Probe code insertion method)	2,147,483,646
Number of times of execution	9,223,372,036,854,775,8072
Execution time measurement clock of function in time measurement processing using probe code insertion method	2,147,483,646 * number of CPU clocks per timer count

Remark The number of CPU clocks per timer count is specified by the user in the time measurement file (timer control code block). Refer to "[4.4 Timer Control Code Block](#)" for details of a file for measuring time (timer control code block).

APPENDIX C INTERMEDIATE FILES

This appendix explains intermediate files output when the TW850 analyzes or tunes the performance of the user program or the execution module.

The intermediate files output are listed below.

Ultimately, the tuned execution module (original execution module name)_o.out is output.

However, if a hex file is specified as a debug target, a hex format file of the tuned execution module "(original execution module name)_o.hex" is output.

If the (original project file name).tcl exists, a Tcl file for a project file "(original project file name)_o.tcl" is output. If tw.tcl or the (original execution module name).tcl exists, a Tcl file of a tuned execution module "(original execution module name)_o.tcl" is output.

twerrlog.txt,	(original project file name)_s.prj,
(original project file name)_i.tcl,	(original project file name)_i.tcl,
(original project file name)_p.prj,	(original project file name)_p.tcl,
(original project file name)_o.prj,	(original project file name)_o.tcl,
(original user program name).s,	(original user program name).stw,
(original user program name)_i.c2s,	(original user program name)_i.s,
(original user program name)_p.c2s,	(original user program name)_p.s,
(original user program name)_o.c2s,	(original user program name)_o.s,
(original user program name)_i.o,	(original user program name)_p.o,
(original user program name)_o.o,	twprftbl.s,
twprftbl.o,	(timer control code block program file name).o,
(original project file name).pdb,	(original project file name)_*.pdb,
(original execution module name)_i.out,	(original execution module name)_i.xxx,
(original execution module name)_i.tcl,	(original execution module name)_i.tft,
(original execution module name)_p.out,	(original execution module name)_p.xxx,
(original execution module name)_p.tcl,	(original execution module name)_o.out,
(original execution module name)_o.xxx,	(original execution module name)_o.tcl,
(original execution module name)_o.tft,	(original link directive file name)

- twerrlog.txt

This is a log file storing error messages.

This file is output to the same folder as the original project file if the TW850 detects a fatal error.

Remark If no project is specified, this file is output to the same folder as the execution format file of the TW850.

- (original project file name)_s.prj

This is a project file that converts a C source program into an assembly-language source program.

This file is output to the same folder as the original project file when the performance is analyzed.

- (original project file name)_i.prj

This is a project file used to generate an execution module in the original status (execution module for original analysis).

This file is output to the same folder as the original project file when the performance is analyzed.

- (original project file name)_i.tcl

This is a Tcl file for a project file used to execute an execution module in the original status (execution module for original analysis).

This file is output to the same folder as the original project file when the performance is analyzed.

Remark If the (original project file name).tcl exists, this file is output.

- (original project file name)_p.prj

This is a project file used to generate an execution module with probe code inserted (execution module for profile).

This file is output to the same folder as the original project file when the performance is analyzed by the probe code insertion method.

- (original project file name)_p.tcl
This is a Tcl file for a project file used to execute an execution module with probe code inserted (execution module for profile).
This file is output to the same folder as the original project file when the performance is analyzed by the probe code insertion method.

Remark If the (original project file name).tcl exists, this file is output.
- (original project file name)_o.prj
This is a project file used to generate an execution module that has been tuned by the TW850 (tuned execution module).
This file is output to the same folder as the original project file during tuning.
- (original project file name)_o.tcl
This is a Tcl file for a project file used to execute an execution module that has been tuned by the TW850 (tuned execution module).
This file is output to the same folder as the original project file during tuning.

Remark If the (original project file name).tcl exists, this file is output.
- (original user program name).s
This is an intermediate output file in which the original user program (C source file) has been converted into an assembly source program to acquire performance analysis information.
This file is output to the same folder as the original project file when the performance is analyzed.

Remark If the output destination folder of the intermediate output files is defined by the PM plus, this file is output to that folder.
- (original user program name).stw
This is an intermediate output file in which the original user program (assembly source file) has been converted into an assembly source program for acquiring performance analysis information.
This file is output to the same folder as the original project file when the performance is analyzed.

Remark If the output destination folder of the intermediate output files is defined by the PM plus, this file is output to that folder.
- (original user program name)_i.c2s
This is an intermediate output file in which the original user program (C source file) has been converted into an assembly source program to acquire performance analysis information.
This file is output to the same folder as the original project file when an execution module (execution module for original analysis) is generated.

Remark If the output destination folder of the intermediate output files is defined by the PM plus, this file is output to that folder.
- (original user program name)_i.s
This is an intermediate output file in which the original user program (assembly source file) has been converted into an assembly source program for acquiring performance analysis information.
This file is output to the same folder as the original project file when an execution module (execution module for original analysis) is generated.

Remark If the output destination folder of the intermediate output files is defined by the PM plus, this file is output to that folder.
- (original user program name)_p.c2s
This is an intermediate output file in which the original user program (C source file) has been converted into an assembly source program to acquire performance analysis information.
This file is output to the same folder as the original project file when an execution module (execution module for profile) is generated.

Remark If the output destination folder of the intermediate output files is defined by the PM plus, this file is output to that folder.

- (original user program name)_p.s

This is an intermediate output file in which the original user program (assembly source file) has been converted into an assembly source program for acquiring performance analysis information.

This file is output to the same folder as the original project file when an execution module (execution module for profile) is generated.

Remark If the output destination folder of the intermediate output files is defined by the PM plus, this file is output to that folder.

- (original user program name)_o.c2s

This is an intermediate output file in which the original user program (C source file) has been converted into an assembly source program to acquire performance analysis information.

This file is output to the same folder as the original project file when an execution module (tuned execution module) is generated.

Remark If the output destination folder of the intermediate output files is defined by the PM plus, this file is output to that folder.

- (original user program name)_o.s

This is an intermediate output file in which the original user program (assembly source file) has been converted into an assembly source program for acquiring performance analysis information.

This file is output to the same folder as the original project file when an execution module (tuned execution module) is generated.

Remark If the output destination folder of the intermediate output files is defined by the PM plus, this file is output to that folder.

- (original user program name)_i.o

This file is an intermediate file in which the original user program (C source file or assembly source file) has been converted into an object file for acquiring performance analysis information.

This file is output to the same folder as the original project file when an execution module (execution module for original analysis) is generated.

Remark If the output destination folder of the intermediate output files is defined by the PM plus, this file is output to that folder.

- (original user program name)_p.o

This file is an intermediate file in which the original user program (C source file or assembly source file) has been converted into an object file for acquiring performance analysis information.

This file is output to the same folder as the original project file when an execution module (execution module for profile) is generated.

Remark If the output destination folder of the intermediate output files is defined by the PM plus, this file is output to that folder.

- (original user program name)_o.o

This file is an intermediate file in which the original user program (C source file or assembly source file) has been converted into an object file for acquiring performance analysis information.

This file is output to the same folder as the original project file when an execution module (tuned execution module) is generated.

Remark If the output destination folder of the intermediate output files is defined by the PM plus, this file is output to that folder.

- twprftbl.s

This is an assembly source program for acquiring profile information.

This file is output to the same folder as the original project file when the <Calculate> button on the profile table specification area of the [Allocation Destination Selection Panel](#) is selected or when an execution module (execution module for profile) is generated.

- twprftbl.o

This is an intermediate output file in which an assembly source program for acquiring profile information has been converted into an object file for acquiring profile information.

This file is output to the same folder as the original project file when an execution module (execution module for profile) is generated.

Remark If the output destination folder of the intermediate output files is defined by the PM plus, this file is output to that folder.

- (timer control code block program file name).o

This is an intermediate output file in which a timer control code block program file (assembly source program) has been converted into a timer control code block program file (object file).

This file is output to the same folder as the original project file when an execution module (execution module for profile) is generated.

However, note the following.

- Remarks**
1. This file is output only when "probe code analysis method" and "measurement of execution time of function" are specified as performance analysis items.
 2. If the output destination folder of the intermediate output files is defined by the PM plus, this file is output to that folder.

- (original project file name).pdb

This is a file that stores the project information of the TW850.

This file is output to the "(original project file name).tw" folder when [File] menu -> [Save TW Project] selected on the [Main Window](#) of the TW850 or when the TW850 is terminated.

- (original project file name)_*.pdb

This is a backup file of a project file for the TW850 (a file that stores the project information of the TW850).

If a TW850 project file of a version different from that of the TW850 currently being used is read, this file is output to the "(original project file name).tw" folder.

- (original execution module name)_i.out

This is an execution module for original analysis.

This file is output to the "(original project file name).tw" folder.

- (original execution module name)_i.xxx

This is a hex format file of an execution module for original analysis.

This file is output to the "(original project file name).tw" folder.

Remark The extension ".xxx" of this file is the same as the extension of the hex format file of the original execution module.

- (original execution module name)_i.tcl

This is a Tcl file of an execution module for original analysis.

This file is output to the "(original project file name).tw" folder.

Remark If the (original execution module name).tcl exists, this file is output.

- (original execution module name)_i.tft

This is a data file to store function transition information of an execution module for original analysis.

This file is output to the "(original project file name).tw" folder.

Remark If the function transition information is acquired, this file is output.

- (original execution module name)_p.out

This is an execution module for profile.

This file is output to the "(original project file name).tw" folder.

- (original execution module name)_p.xxx

This is a hex format file of an execution module for profile.

This file is output to the "(original project file name).tw" folder.

Remark The extension ".xxx" of this file is the same as the extension of the hex format file of the original execution module.

- (original execution module name)_p.tcl
This is a Tcl file of an execution module for profile.
This file is output to the "(original project file name).tw" folder.

Remark If the (original execution module name).tcl exists, this file is output.
- (original execution module name)_o.out
This is a tuned execution module.
This file is output to the "(original project file name).tw" folder.
- (original execution module name)_o.xxx
This is a hex format file of a tuned execution module.
This file is output to the "(original project file name).tw" folder.

Remark The extension ".xxx" of this file is the same as the extension of the hex format file of the original execution module.
- (original execution module name)_o.tcl
This is a Tcl file of a tuned execution module.
This file is output to the "(original project file name).tw" folder.

Remark If the (original execution module name).tcl exists, this file is output.
- (original execution module name)_o.tft
This is a data file to store function transition information of a tuned execution module.
This file is output to the "(original project file name).tw" folder.

Remark If the function transition information is acquired, this file is output.
- (original link directive file name)
This is a link directive file used to generate an execution module (execution module for original analysis, execution module for profile, or tuned execution module).
This file is output to the "(original project file name).tw" folder.

- Cautions 1. Because the TW850 determines the name of an intermediate file based on the source file name, if a source file having the same base name is in the project (such as source.c and source.s, and source.850), it cannot be tuned.
2. When a file of the same name as an intermediate file is included in a source file (such as when xxx/source.s is included in source.c), that file may not be tuned because the intermediate file may be preferentially included.
 3. If two or more projects contain the same source file (excluding an assembly source file), and if each of these objects is alternately tuned by the TW850, they may not be correctly built or an illegal module may be generated.
In such a case, building can be correctly executed by deleting the intermediate file in the [Option Dialog Box](#) after changing the project.
 4. The project generated by the TW850 or the project including the file generated by the TW850 cannot be tuned.

INDEX

A

Allocation Destination Selection Panel	159
Analysis Source File Selection Panel	123

B

Basic software	17
----------------------	----

C

C compiler package	17
Callgraph Window	103
Callpair	15
Callpair Correction Option Selection Panel	154
CPU	17

D

Debugger	17
Device file	17

E

Error Dialog Box	184
Error Messages	194, 197
Execution Analysis Range Selection Panel	133
Execution Completion Confirmation Panel	166
Execution engine	15
Execution Item Selection Panel	126
Execution module	15
Execution Module Information Acquisition Panel	125
Execution Module Information Window	114
Exiting Method	25

F

Far jump calling	15
Fast access memory area	15
Fast Access Memory Area Selection Panel	144
Fast Access Memory Area Tuning Selection Panel	149
Features	16
File Selection Dialog Box	180
Folder Configuration	20
Function Detail Information Window	116
Function Information Table Window	106
Function Transition Information Window	109
FUNCTIONS	26

G

GUI functions	16
---------------------	----

H

Hardware environment	17
CPU	17
Host machine	17
Main memory	17
Host machine	17
Hot path	15

I

Information Messages	194
Initial Screen Panel	120
Input Dialog Box	188
INSTALLATION	18, 20
Folder Configuration	20
Installation Procedure	18
Uninstallation Procedure	19
Installation Procedure	18
Instruction cache miss	15
Instruction Cache Selection Panel	141
Instruction Cache Tuning Selection Panel	151
INTERMEDIATE FILES	215

J

Java execution environment	17
----------------------------------	----

L

Limit	214
-------------	-----

M

Main memory	17
Main Window	95
Message Dialog Box	183
MESSAGES	193
Display Format	193
Error Messages	194, 197
Information Messages	194
Warning Messages	194, 196
METHOD OF OPERATION	54

O

Operating Environment	17
-----------------------------	----

Hardware environment	17
Software environment	17
Option Dialog Box	175

P

Performance analysis functions	16
Performance Analysis/Tuning Execution Panel	164
Pre-execution Confirmation Panel	163
Priority Allocation Function Selection Panel	156
Project Information Acquisition Panel	122
Project manager	17

R

Restriction	214
RESTRICTION AND LIMIT	214
RTOS(Real-time OS)	17

S

Search Function Transition Information Dialog Box	171
Select Display Function Dialog Box	167
Simulator	17
Software environment	17
Basic software	17
C compiler package	17
Debugger	17
Device file	17
Java execution environment	17
Project manager	17
RTOS(Real-time OS)	17
Simulator	17
Start options	24
Starting Method	23
System Library for TW850	42

T

Timer Control Code Block	38
Trace Event Selection Panel	137
Tuning functions	16
Tuning Method Selection Panel	146
TW850	15
Exiting Method	25
Features	16
FUNCTIONS	26
GENERAL	15
GUI functions	16
INSTALLATION	18
INTERMEDIATE FILES	215
MESSAGES	193
METHOD OF OPERATION	54
Operating Environment	17
Performance analysis functions	16
RESTRICTION AND LIMIT	214

Start options	24
STARTING AND EXITING	22
Starting Method	23
Tuning functions	16
WINDOW REFERENCE	90
TW850 Options Dialog Box	190

U

Uninstallation Procedure	19
--------------------------------	----

V

Verification Dialog Box	186
Version Information Dialog Box	178

W

Warning Messages	194, 196
WINDOW REFERENCE	90
Allocation Destination Selection Panel	159
Analysis Source File Selection Panel	123
Callgraph Window	103
Callpair Correction Option Selection Panel	154
Error Dialog Box	184
Execution Analysis Range Selection Panel	133
Execution Completion Confirmation Panel	166
Execution Item Selection Panel	126
Execution Module Information Acquisition Panel	125
Execution Module Information Window	114
Fast Access Memory Area Selection Panel	144
Fast Access Memory Area Tuning Selection Panel .	149
File Selection Dialog Box	180
Function Detail Information Window	116
Function Information Table Window	106
Function Transition Information Window	109
Initial Screen Panel	120
Input Dialog Box	188
Instruction Cache Selection Panel	141
Instruction Cache Tuning Selection Panel	151
Main Window	95
Message Dialog Box	183
Option Dialog Box	175
Performance Analysis/Tuning Execution Panel	164
Pre-execution Confirmation Panel	163
Priority Allocation Function Selection Panel	156
Probe Code Insertion Analysis Method Option Selection Panel	130
Project Information Acquisition Panel	122
Search Function Transition Information Dialog Box .	171
Select Display Function Dialog Box	167
Trace Data Analysis Method Option Selection Panel	128
Trace Event Selection Panel	137
Tuning Method Selection Panel	146
TW850 Options Dialog Box	190
Verification Dialog Box	186
Version Information Dialog Box	178