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.

RENESAS

NEC

User's Manual

TW Ver. 1.00

Performance Analysis Tuning Tool

Target Device
V850 Series™

Document No. U16104EJ1V0UM00 (1st edition)
Date Published October 2002 CP(K)

Printed in Japan

[MEMO]

V850 Series, V852, V853, V850/SA1, V850/SB1, V850/SB2, V850/SC1, V850/SC2, V850/SC3, V850/SF1, V850/SV1, V850E/MS1, V850E/MS2, V850E/MA1, V850E/MA2, V850E/IA1, and V850E/IA2 are trademarks of NEC Corporation.

Pentium is a trademark of Intel Corporation.

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

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

Green Hills Software and MULTI are trademarks of Green Hills Software, Inc.

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

The export of these products from Japan is regulated by the Japanese government. The export of some or all of these products may be prohibited without governmental license. To export or re-export some or all of these products from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

- **The information in this document is current as of October, 2002. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC's data sheets or data books, etc., for the most up-to-date specifications of NEC semiconductor products. Not all products and/or types are available in every country. Please check with an NEC sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without prior written consent of NEC. NEC assumes no responsibility for any errors that may appear in this document.
- NEC does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC semiconductor products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of customer's equipment shall be done under the full responsibility of customer. NEC assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC endeavours to enhance the quality, reliability and safety of NEC semiconductor products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC semiconductor products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment, and anti-failure features.
- NEC semiconductor products are classified into the following three quality grades:
  "Standard", "Special" and "Specific". The "Specific" quality grade applies only to semiconductor products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of a semiconductor product depend on its quality grade, as indicated below. Customers must check the quality grade of each semiconductor product before using it in a particular application.
  "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots
  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support)
  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.
  The quality grade of NEC semiconductor products is "Standard" unless otherwise expressly specified in NEC's data sheets or data books, etc. If customers wish to use NEC semiconductor products in applications not intended by NEC, they must contact an NEC sales representative in advance to determine NEC's willingness to support a given application.
  (Note)
  (1) "NEC" as used in this statement means NEC Corporation and also includes its majority-owned subsidiaries.
  (2) "NEC semiconductor products" means any semiconductor product developed or manufactured by or for NEC (as defined above).

M8E 00. 4

Regional Information

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

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

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

NEC Electronics Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

NEC do Brasil S.A.
Electron Devices Division
Guarulhos-SP, Brasil
Tel: 11-6462-6810
Fax: 11-6462-6829

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 01
Fax: 0211-65 03 327

- **Sucursal en España**
  Madrid, Spain
  Tel: 091-504 27 87
  Fax: 091-504 28 60

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

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

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

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

- **United Kingdom Branch**
  Milton Keynes, UK
  Tel: 01908-691-133
  Fax: 01908-670-290

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

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

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377
Fax: 02-2719-5951

NEC Electronics Singapore Pte. Ltd.
Novena Square, Singapore
Tel: 253-8311
Fax: 250-3583

J02.4

[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 TW in the organization below.

Organization

This manual consists of the following chapters:

- GENERAL
- INSTALLATION
- STARTING AND TERMINATING
- FUNCTIONS
- OPERATION
- WINDOW REFERENCE
- MESSAGES

How to Use This Manual

It is assumed that the readers of this manual have general knowledge of electrical engineering, logic circuits, microcontrollers, C language, and assemblers.

To understand the functions of the V850 Series

→ Refer to Hardware User's Manual for each product..

To understand the instruction functions of the V850 Series

→ Refer to **V850 Series Architecture User's Manual** or **V850E/MS1™ Architecture User's Manual**.

Conventions

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 … 0xXXXX

Prefix indicating the power of 2 (address space, memory capacity):

K (Kilo): $2^{10} = 1024$

M (Mega): $2^{20} = 1024^2$

Related Documents Refer to the documents listed below when using this manual.
The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such.

Documents related to development tools (User's Manuals)

Document Name		Document No.
IE-703002-MC (In-circuit emulator for V853™, V850/SA1™, V850/SB1™, V850/SB2™, V850/SC1™, V850/SC2™, V850/SC3™, V850/SF1™, V850/SV1™)		U11595E
IE-V850E-MC (In-circuit emulator for V850E/IA1™, V850E/IA2™), IE-V850E-MC-A (In-circuit emulator for V850E/MA1™, V850E/MA2™)		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, V850E/MS2™)		U13875E
IE-703102-MC-EM1, IE-703102-MC-EM1-A (In-circuit emulator option board for V850E/MS1, V850E/MS2)		U13876E
IE-703107-MC-EM1 (In-circuit emulator option board for V850E/MA1)		U14481E
IE-703116-MC-EM1 (In-circuit emulator option board for V850E/IA1)		U14700E
CA850 Ver. 2.50 C Compiler Package	Operation	U16053E
	C Language	U16054E
	PM plus	U16055E
	Assembly Language	U16042E
ID850 Ver. 2.50 Integrated Debugger	Operation	U16217E
SM850 Ver. 2.50 System Simulator	Operation	U16218E
SM850 Ver. 2.00 or Later System Simulator	External Part User Open Interface Specifications	U14873E
RX850 Ver. 3.13 or Later Real-Time OS	Basics	U13430E
	Installation	U13410E
	Technical	U13431E
RX850 Pro Ver. 3.13 Real-Time OS	Fundamental	U13773E
	Installation	U13774E
	Technical	U13772E
RD850 Ver. 3.01 Task Debugger		U13737E
RD850 Pro Ver. 3.01 Task Debugger		U13916E
AZ850 Ver. 3.10 System Performance Analyzer		U14410E
PG-FP3 Flash Memory Programmer		U13502E
PG-FP4 Flash Memory Programmer		U15260E
TW Ver. 1.00 Performance Analysis Tuning Tool		This manual

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 GENERAL

1.1 Overview

TW is software (performance analysis tuning tool) that analyzes or tunes the execution performance of the user program and execution module running on NEC's V850 Series of 32-bit microcontrollers for embedded control.

TW supplies the following Performance analysis functions, Tuning functions, and GUI functions.

- Performance analysis functions
  - Acquiring memory allocation information

    Acquires memory allocation information in function units by statically analyzing the user program or execution module.
  - Acquiring callpair information

    Acquires callpair information (calling relationship between functions) in function units by statically analyzing the user program and execution module.
  - Acquiring call count information

    Acquires call count information (which function has called which function how many times) in function units by analyzing the hardware trace data output by the 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 position at which the function to be analyzed is to be called.
  - Acquiring execution time information

    Acquires execution time information in function units by analyzing the hardware trace data output by the 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 location at which the function to be analyzed is to be called.
  - Acquiring instruction cache information

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

    Acquires RTOS information (such as the number of times of executing processing in the RTOS and the execution time) in function units by analyzing the hardware trace data output by the 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 location of calling or returning in the RTOS (NEC's RX850 or RX850 Pro).

- Tuning functions
  - Tuning of instruction cache

    Improves the execution performance of the execution module by allocating functions so that the number of instruction cache misses decreases.
  - Tuning of fast access memory area

    Raises the execution performance of the execution module by giving priority to and allocating the function that causes the execution performance of the execution module to drop to the fast access memory area (such as internal ROM).
  - Correcting callpair information

    Improves the improvement rate of tuning processing (tuning of instruction cache or tuning of fast access memory area) by correcting the callpair information (calling relationship between functions) of each function.
  - Specifying function to be given priority and allocated

    Raises the improvement rate of tuning processing (tuning of instruction cache or tuning of fast access memory area) by specifying a function that should be given priority and tuned. If "a function to be given priority and allocated" is specified, TW basically assumes this function as the function with the highest priority for tuning of the fast access memory area.

- GUI functions
  - Setting in wizard format

    By providing graphical guidance in wizard format to the settings necessary for using the performance analysis and tuning functions supplied by TW (such as setting the performance analysis zone or execution environment), each setting can be made easily and efficiently.
  - Displaying performance analysis and tuning results

    By graphically displaying the results of performance analysis and tuning performed by TW, these results can be easily compared and evaluated.

1.2 Features

TW 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 the performance analysis functions supplied by TW.

- Easy performance analysis and tuning

  By providing graphical guidance in wizard format to the settings necessary for using the performance analysis and tuning functions supplied by TW (such as setting the performance analysis zone or execution environment), each setting can be made easily and efficiently.

- Supply of detailed performance analysis information

  The location that causes the execution performance of the execution module to drop can be detected in function units.

- Automatic generation of execution module after tuning

  By making settings (inputting necessary information) via the graphical guidance enabled by the wizard format, an execution module that has been tuned (tuned execution module) is automatically generated by TW.

1.3 Operating Environment

1.3.1 Hardware environment

The hardware necessary for TW to operate is as follows.

- Host machine

  IBM-PC/AT-compatible machine

- CPU

  Pentium 450 MHz or over

- Main memory

  128 Mbytes or more (256 Mbytes or more is recommended.)

1.3.2 Software environment

The following software is necessary for TW to operate.

- Basic software

  Windows 2000, Windows 98 Second Edition, Windows Me, Windows XP, Windows NT 4.0 Service Pack 5 or later

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

- Java execution environment

  Java2 Runtime Environment Standard Edition v1.3 : Sun Microsystems, Inc.

- C compiler package

  CA850 Ver.2.50 or later : NEC
  MULTI2000 CCV850/CCV850E Release 6.5.0 : Green Hills Software, Inc.

- Device file

  Device file of target device used : NEC

- Project manager

  PM plus Ver.5.00 or later : NEC
  MULTI2000 builder Release 6.5.0 : Green Hills Software, Inc.

- Debugger

  ID850 Ver.2.50 or later : NEC
  MULTI2000 debugger Release 6.5.0 : Green Hills Software, Inc.

- ICE server

  850eserv : Advanced Data Controls

- Simulator

  SM850 Ver.2.50 or later : NEC

CHAPTER 2 INSTALLATION

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

2.1 Overview

A TW supply medium is available for each type of host machine (Windows-based).
Table 2-1 shows the TW supply medium.

Table 2-1 TW Supply Medium

Host Machine	Format	Medium
Windows-based • IBM-PC/AT-compatible machine	NEC tool chain-supporting version GHS tool chain-supporting version	CD-ROM

Caution Two types of TWs (NEC tool chain-supporting version and GHS tool chain-supporting version) are stored in the supply medium for each type of host machine. To install the files from the supply medium to the host machine, the TW supporting the user's development environment must be installed.

2.2 Installation Procedure

Install the files stored in the TW supply medium in the host machine using the following procedure.

Caution Before re-installing the NEC tool chain-supporting version or GHS tool chain-supporting version of TW, uninstall TW.
When installing the TW supporting the GHS tool chain, first install C compiler package CCV850/CCV850E (Green Hills Software, Inc.).

1) Start Windows

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

2) Set supply medium

Set the TW supply medium in the CD-ROM drive of the host machine. The setup program will be automatically executed.
After that, perform installation in accordance with the messages displayed on the monitor screen.

Caution If the setup program is not automatically executed, start SETUP.EXE stored in the TW\DISK1 folder.

3) Check files

Check that the files stored in the TW supply medium have been installed in the host machine by using a standard application of Windows such as Explorer.
For details of each folder, refer to **2.4 Folder Configuration**.

2.3 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.

4) Delete TW

   From the list of installed software displayed on the [Install/Uninstall] tab in the [Add/Remove Programs Properties] window, select "NEC TW xxxxxxxxx" 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.
   For details of each folder, refer to **2.4 Folder Configuration**.

2.4 Folder Configuration

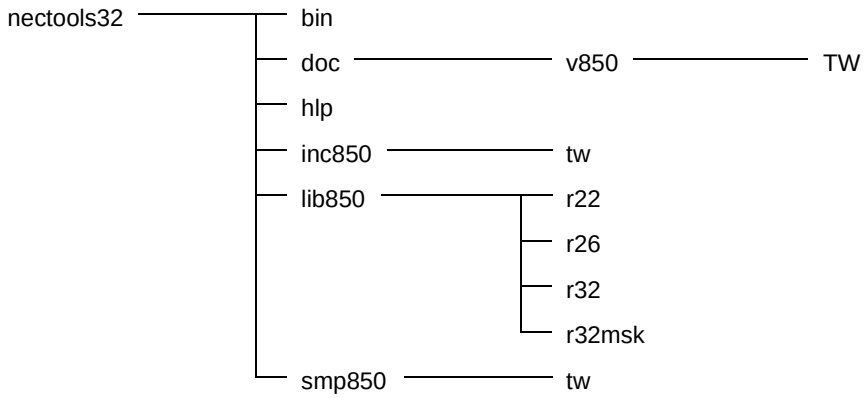
The folder configuration of the files stored in the TW supply medium differs depending on the user's development environment (NEC tool chain or GHS tool chain).

Therefore, the folder configuration for the NEC tool chain and that for the GHS tool chain are explained separately.

2.4.1 NEC tool chain-supporting version

Figure 2-1 shows the folder configuration that is generated when the files stored in the TW supply medium (NEC tool chain-supporting version) are installed in the host machine.

Figure 2-1 Folder Configuration (NEC Tool Chain-Supporting Version)



Each folder is described briefly below.

1) nectools32\bin

This folder stores the executable files of TW.

TW.EXE : Executable file of TW

2) nectools32\doc\v850\TW

This folder stores the release document of TW.

3) nectools32\hlp

This folder stores the online help of TW.

TW.CHM : Online help

4) nectools32\inc850\tw

This folder stores the header files of TW.

5) nectools32\lib850

This folder stores the system library for TW and data files for performance analysis/tuning.

6) nectools32\lib850\r22

This folder stores the system library for TW (22-register mode) and data files for performance analysis/tuning (22-register mode).

7) nectools32\lib850\r26

This folder stores the system library for TW (26-register mode) and data files for performance analysis/tuning (26-register mode).

8) nectools32\lib850\r32

This folder stores the system library for TW (32-register mode) and data files for performance analysis/tuning (32-register mode).

9) nectools32\lib850\r32msk

This folder stores the system library for TW (32-register mask mode) and data files for performance analysis/tuning (32-register mask mode).

10) nectools32\smp850\tw

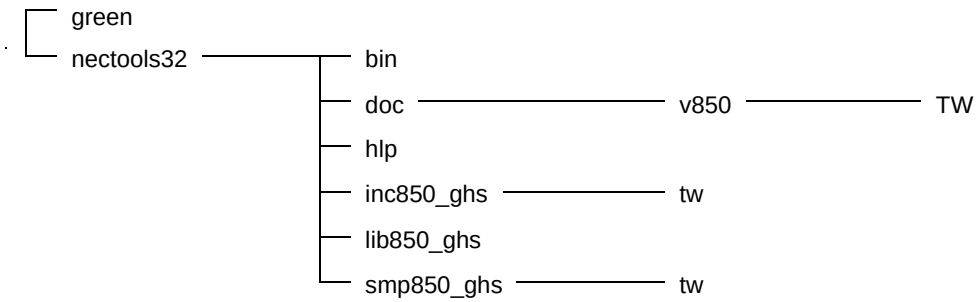
This folder stores the sample program of the timer control code block.
For details of the timer control code block, refer to **4.6 Timer Control Code Block**.

TWTIMER85.S	:	Timer control code block for V853 TM11
TWTIMER85E.S	:	Timer control code block for V850E/MS1 TM10

2.4.2 GHS tool chain-supporting version

Figure 2-2 shows the folder configuration that is generated when the files stored in the TW supply medium (GHS tool chain-supporting version) are installed in the host machine.

Figure 2-2 Folder Configuration (GHS Tool Chain-Supporting Version)



Each folder is described briefly below.

1) green

This folder stores C compiler package CCV850/CCV850E (from Green Hills Software, Inc.).
When TW has been installed, the file name of the original assembler changes from AS850.EXE to AS850ORG.EXE.

AS850.EXE	:	Wrapper command for assembler supplied by TW
AS850ORG.EXE	:	Original assembler supplied by CCV850/CCV850E

2) nectools32\bin

This folder stores the executable files of TW and replacement commands of assembler AS850.EXE.

TW.EXE	:	Executable file of TW
TWWSETUP.EXE	:	Replacement command of assembler AS850.EXE

3) nectools32\doc\v850\TW

This folder stores the release document of TW.

4) nectools32\hlp

This folder stores the online help of TW.

TW.CHM	:	Online help

5) nectools32\inc850_ghs\tw

This folder stores the header files of TW.

6) nectools32\lib850_ghs

This folder stores the system library for TW and data files for performance analysis/tuning.

7) nectools32\smp850_ghs\tw

This folder stores the sample program of the timer control code block.
For details of the timer control code block, refer to **4.6 Timer Control Code Block**.

TMTIMER85.850	:	Timer code control block for V853 TM11
TWTIMER85E.850	:	Timer control code block for V850E/MS1 TM10

CHAPTER 3 STARTING AND TERMINATING

This chapter describes how to start and exit TW.

3.1 Overview

Before starting TW, check the following.

- Java execution environment

  A Java execution environment is necessary for starting TW.
  TW searches for a Java application start command (javaw.exe) in the following sequence.

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

- NEC tool chain

  To start TW with the NEC tool chain, the project that is used with TW must be correctly built and debugged with PM plus, ID850, or SM850.

- GHS tool chain

  To start TW with the GHS tool chain, a path for TW to use must exist in the execution environment of MULTI, and the project that is used with TW must be correctly built and debugged with MULTI.
  The device file to be used with the project that is used with TW must be installed at the specified location by the device file installer.

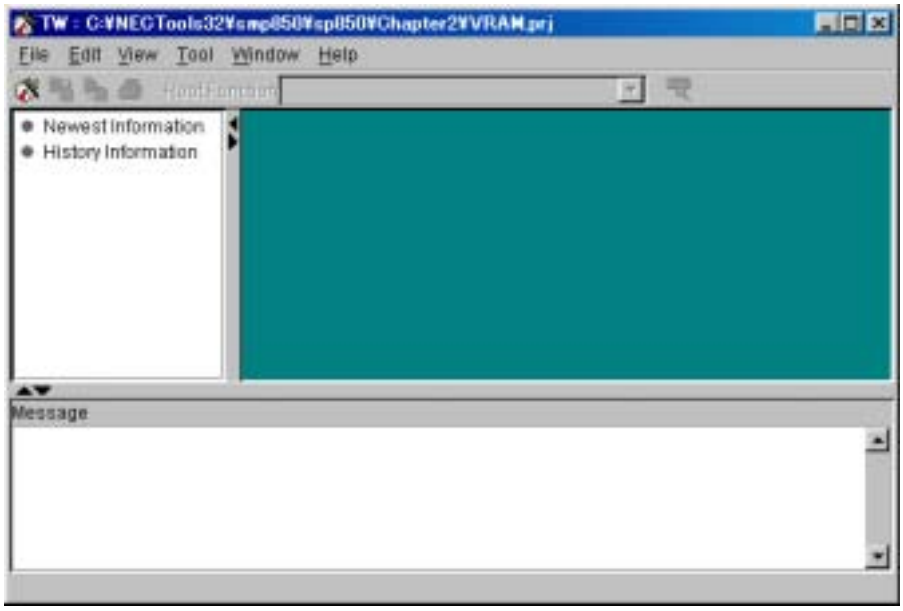
3.2 Starting Method

3.2.1 Starting from PM plus

This section explains how to start TW from the menu bar of PM plus.
When TW has been correctly started, the Main Window shown in Figure 3-1 is automatically opened.

1) Select [Tool] menu -> [Startup TW].

Figure 3-1 Main Window



Cautions 1. Specify the start option in the Start Option Setting Dialog Box.

2. If the <Startup TW> button on the tool bar of PM plus is clicked, processing equivalent to the above operation is performed.

3.2.2 Starting from task bar

Start TW from the task bar of Windows as follows.
Note, however, that only TW supporting the GHS tool chain can be started from the task bar.
When TW has been correctly started, the Main Window is automatically opened.

1) Select [Start] menu -> [Programs] menu -> [NEC Tools32] menu -> [TW for Custom Tool Chain].

Caution Specify the start option in the property dialog box.
The property dialog box is automatically opened when [Properties] is selected from the pop-up menu that is displayed by clicking the right mouse button when the above is selected.
For details of the start options, refer to **3.2.3 Starting from command line**.

3.2.3 Starting from command line

Start TW from the command line of the MS-DOS prompt as follows.
"C>" in the example below indicates the shell prompt and "<Enter>" indicates input of the enter key.
Start options enclosed in "[ ]" may be omitted.
When TW has been correctly started, the Main Window is automatically opened.

[Input example]

```
C> TW.EXE [BldFile] [-Dtw.ToolChain = Tools] [-Dtw.DebuggerTimeout = DbgTmout] [-Dtw.MultiServerTimeout =
   SrvTmout] [-Dtw.MultiShellWaitCount = WaitCnt] <Enter>
```

The start options of TW are as follows.

- *BldFile*

  Specify the file name of the build file required if the user development environment is the GHS tool chain.

- -Dtw.ToolChain=*Tools*

  Specify the type of user development environment.
  The value that can be specified as *Tools* is limited to NEC or GHS.

  NEC : NEC tool chain
  GHS : GHS tool chain

  If this option is omitted, TW executes processing assuming that -Dtw.ToolChain = GHS is specified.

- -Dtw.DebuggerTimeout=*DbgTmout*

  Set the response wait time (in seconds) during which TW 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, TW executes processing assuming that -Dtw.DebuggerTimeout = 300 is specified.

  Caution If -Dtw.ToolChain = GHS is specified, or if the -Dtw.ToolChain option is omitted, specification of this option is invalid.

- -Dtw.MultiServerTimeout=*SrvTmout*

  Specify the timeout value (in seconds) during which TW acquires the hardware trace data output by the execution engine.
  The value that can be specified for *SrvTmout* is limited to 1 to 3600.
  If this option is omitted, TW executes processing assuming that -Dtw.MultiServerTimeout = 1000 is specified.

  Caution This option is invalid if -Dtw.ToolChain = NEC is specified.

- -Dtw.MultiShellWaitCount=*WaiCnt*

  Specify the number of loops (number of loops of vacant processing) of wait processing to be inserted before and after an external command that is called from the MULTI script.
  The value that can be specified as *WaitCnt* is limited to 1 to 10000.
  If this option is omitted, TW executes processing assuming that -Dtw.MultiShellWaitCount = 2000 is specified.

  Caution This option is invalid if -Dtw.ToolChain = NEC is specified.

- -Xmx*HeapSiz***m**

  Specify the maximum size (in Mbytes) that JavaVM can use as a heap memory area.
  If this option is omitted, TW executes processing assuming that -Xmx*HeapSiz*m is not specified.

  Caution To specify 64 Mbytes as the maximum size of a heap memory area, specify -Xmx64m.

- -Xss*StkSiz*k

  Specify the maximum size (in Kbytes) that JavaVM can use as a thread stack memory area.
  If this option is omitted, TW executes processing assuming that -Xss*StkSiz*k is not specified.

  Caution To specify 40 Kbytes as the maximum size of the thread stack memory area, specify -Xss40k.

3.3 Terminating Method

3.3.1 Terminating from menu bar

Terminate TW from the menu bar of the Main Window as follows.
When TW is terminated correctly, 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

3.3.2 Terminating from title bar

Terminate TW from the title bar of the Main Window as follows.
When TW is terminated correctly, 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 the performance analysis, tuning, and GUI functions supplied by TW.

4.1 Overview

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

TW supplies functions such as performance analysis, tuning, and GUI.

4.2 Performance Analysis Function

TW 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

TW supplies the following two methods (trace data analysis method and probe code insertion method) for acquiring performance analysis information.

- Trace data analysis method

  By this method, performance analysis information is acquired by analyzing the hardware trace data output by the execution engine (such as the debugger or simulator) when an execution module is executed.
  TW temporarily stops execution of the execution module, and acquires, analyzes, and clears the contents of the trace buffer (hardware trace data) if the trace buffer of the execution engine used to store the hardware trace data is full, and at the performance analysis start and end positions. Performance analysis cannot be performed using the trace data analysis method for an execution engine for which trace full break (or equivalent processing) cannot be specified, or for an execution module that has been stopped and whose correct operation therefore cannot be guaranteed.

- Probe code insertion method

  By this method, performance analysis information is acquired by executing an execution module with measurement probe code inserted at the location where the function whose performance is to be analyzed is called.
  TW 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 at the performance analysis start or end position cannot be analyzed by using the probe code insertion method.

Caution TW analyzes the performance of a function allocated to the ".text" section.
TW converts a user program in C into an assembly language, and then acquires and analyzes the performance analysis information. Therefore, inline expansion functions in the user program in C are not subject to performance analysis by TW.

4.2.1 Acquiring memory allocation information

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

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

4.2.2 Acquiring callpair information

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

4.2.3 Acquiring call count information

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

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).

4.2.4 Acquiring execution time information

Execution time information is acquired in function units by analyzing the hardware trace data output by the execution engine (such as the 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 whose performance is to be analyzed.

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

4.2.5 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 output by the execution engine (such as the 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).

4.2.6 Acquiring RTOS information

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

- Number of times of execution

  TW 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

  TW 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

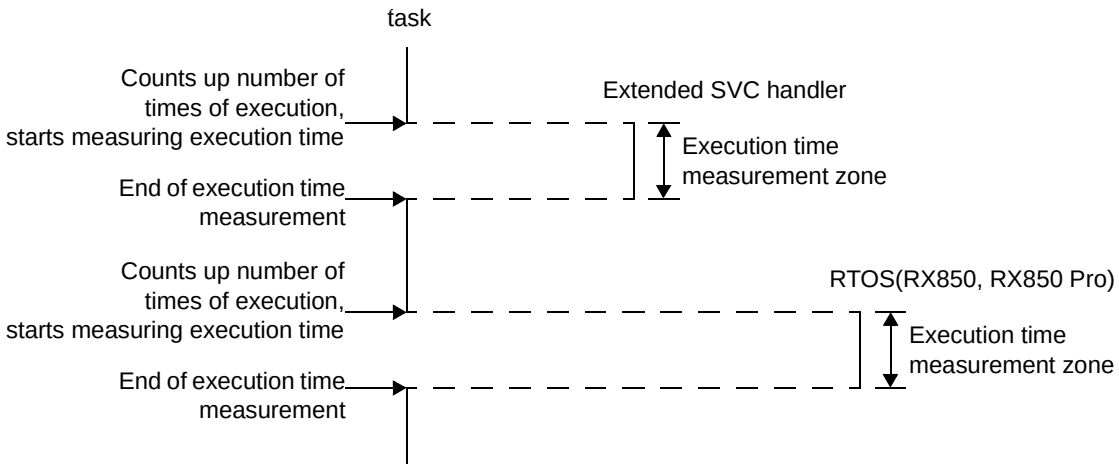
  TW stops measuring the execution time when execution has passed the following locations.

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

- Entrance of idle handler

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



Caution 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.

Caution Inside the extended SVC handler is not subject to trace analysis.

4.3 Tuning Functions

TW supplies the following tuning functions.

- Tuning of instruction cache
- Tuning of fast access memory area
- Correction of callpair information
- Specifying preferentially allocated function

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

tuning of fast access memory area > tuning of instruction cache

4.3.1 Tuning of instruction cache

The execution performance of the execution module can be improved by allocating each function so that the number of instruction cache misses decreases.

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

- Coloring algorithm

  Allocates functions to the instruction cache based on the callpair information (calling relationship between functions) of each function.

- Grouping algorithm

  Groups functions with a strong inter-function relationship and allocates functions in group units, 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.

TW also 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).

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

4.3.2 Tuning of fast access memory area

The execution speed of an execution module is improved by preferentially allocating a function that causes the execution performance of the execution module to drop to the fast access memory area (such as the internal ROM).

TW 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" and "measurement of execution time of a function" are 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

  Based on the instruction cache information of each function (the number of instruction cache hits and the number of instruction cache misses), a function with many instruction cache misses 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 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.

4.3.3 Correction of callpair information

The improvement rate of the tuning processing (tuning of the instruction cache and tuning of the fast access memory area) is raised by correcting the callpair information (calling relationship between functions) of each function.

TW supplies the following three methods for correcting the callpair information (float runtime library, function corresponding to a hot path, and function in loop processing).

- float runtime library

  The execution performance of an execution module is improved by correcting the number of times a callpair related to functions (such as expf and cosf) included in the float runtime library 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 in question is called, based on execution time information including the child function of each 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.

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

float runtime library > function corresponding to a hot path > function in loop processing

4.3.4 Specifying preferentially allocated function

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

If a preferentially allocated function is specified, TW basically gives top priority to the function when tuning the fast access memory area.

Caution If TW 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.

4.4 GUI Functions

TW supplies the following GUI functions.

- Setting in wizard format
- Display of performance analysis and tuning results

4.4.1 Setting in wizard format

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

4.4.2 Display of performance analysis and tuning results

By graphically displaying the results of performance analysis and tuning by TW 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, and Function Detail Information Window).

4.5 Generated Files

TW outputs intermediate files when the performance of the user program and execution module is analyzed or tuned.

However, the intermediate files output differ depending on the user development environment (NEC tool chain or GHS tool chain).

Therefore, the intermediate files that are output if the NEC tool chain is used as the user development environment, and the intermediate files output if GHS tool chain is used are explained below.

4.5.1 NEC tool chain-supporting version

The intermediate files output if the user development environment is the NEC tool chain are listed below.

- twerrlog.txt

  This is a log file storing error messages.
  This file is output to the same folder as the original project file if TW detects a fatal error.

- (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 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 by the probe code insertion method.

- (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 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 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)_o.prj

  This is a project file used to generate an execution module that has been tuned by TW (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 generate an execution module that has been tuned by TW (tuned execution module).
  This file is output to the same folder as the original project file during tuning.

- (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.

- (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.

  Caution If the output destination folder of the intermediate output files is defined by 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 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.

Caution If the output destination folder of the intermediate output files is defined by 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.

  Caution If the output destination folder of the intermediate output files is defined by 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 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.

  Caution If the output destination folder of the intermediate output files is defined by PM plus, this file is output to that folder.

- (original user program name)_p.s

  This file 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.

  Caution If the output destination folder of the intermediate output files is defined by PM plus, this file is output to that folder.

- (original user program name)_o.c2s

  This file is an intermediate output file in which the original user program (C 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.

  Caution If the output destination folder of the intermediate output files is defined by PM plus, this file is output to that folder.

- (original user program name)_o.s

  This file 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.

  Caution If the output destination folder of the intermediate output files is defined by 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.

  Caution If the output destination folder of the intermediate output files is defined by PM plus, this file is output to that folder.

- (original user program name)_p.o

  This is an intermediate output 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.

  Caution If the output destination folder of the intermediate output files is defined by PM plus, this file is output to that folder.

- (original user program name)_o.o

  This is an intermediate output 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.

  Caution If the output destination folder of the intermediate output files is defined by PM plus, this file is output to that folder.

- 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.

  Caution If the output destination folder of the intermediate output files is defined by 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.

  Cautions 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 PM plus, this file is output to that folder.

- (original project file name).pdb

  This is a file that stores the project information of TW.
  This file is output to the "(original project file name).tw" folder when [[File] -> [Save TW Project] selected in the Main Window of TW or when TW is terminated.

- (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.

  Caution 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.

- (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.

  Caution 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.

- (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.

  Caution 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.

- (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.

4.5.2 GHS tool chain-supporting version

The intermediate files that are output if the user's development environment is the GHS tool chain are listed below.

- twerrlog.txt

  This is a log file storing error messages.
  This file is output to the same folder as the original build file if TW detects a fatal error.

- (original build file name)_s.bld

  This is a build file used to convert a user program (C source program or assembly source program) into an assembly source program for acquiring performance analysis information.
  This file is output to the same folder as the original build file when the performance is analyzed.

- (original build file name)_i.bld

  This is a build file used to generate an execution module in the original execution status (execution module for original analysis).
  This file is output to the same folder as the original build file when the performance is analyzed.

- (original build file name)_p.bld

  This is a build 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 build file when the performance is analyzed by the probe code insertion method.

- (original build file name)_o. bld

  This is a build file used to generate an execution module (tuned execution module) that has been tuned by TW.
  This file is output to the same folder as the original build file during tuning.

- (original user program name)_i.o

  This is an intermediate output 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 build file when an execution module (execution module for original analysis) is generated.

- (original user program name)_p.o

  This is an intermediate output 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 build file when an execution module (execution module for profile) is generated.

- (original user program name)_o.o

  This is an intermediate output 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 build file when an execution module (tuned execution module) is generated.

- (original execution module name)_i.*xxx*

  This is an execution module for original analysis.
  This file is output to the "(original build file name).tw" folder.

  Caution The extension of this file ".*xxx*" is the same as the extension of the original execution module.

- (original execution module name)_i.*xxx*.rc

  This is a resource file of an execution module for original analysis.
  This file is output to the "(original build file name).tw" folder.

  Caution The extension of this file ".*xxx*" is the same as the extension of the original execution module.

- (original execution module name)_p.*xxx*

  This is an execution module for profile.
  This file is output to the "(original build file name).tw" folder.

  Caution The extension of this file ".*xxx*" is the same as the extension of the original execution module.

- (original execution module name)_p.*xxx*.rc

  This is a resource file of an execution module for profile.
  This file is output to the "(original build file name).tw" folder.

  Caution The extension of this file ".*xxx*" is the same as the extension of the original execution module.

- (original execution module name)_o.*xxx*

  This is a tuned execution module.
  This file is output to the "(original build file name).tw" folder.

  Caution The extension of this file ".*xxx*" is the same as the extension of the original execution module.

- (original execution module name)_o.*xxx*.rc

  This is a resource file for a tuned execution module.
  This file is output to the "(original build file name).tw" folder.

  Caution The extension of this file ".*xxx*" is the same as the extension of the original execution module.

- (original section map file name)

  This is a section map 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 build file name).tw" folder.

- (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 for acquiring performance analysis information.
  This file is output to the same folder as the original user program when the performance is analyzed.

- (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 user program when the performance is analyzed.

- (original user program name)_i.850

  This is an intermediate output file in which the original user program (C 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 user program when an execution module (execution module for original analysis) is generated.

- (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 user program when execution module (execution module for original analysis) is generated.

- (original user program name)_p.850

  This is an intermediate output file in which the original user program (C 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 user program when execution module (execution module for profile) is generated.

- (original user program name)_p.s

  This is an intermediate output file in which the original user program (C 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 user program when execution module (execution module for profile) is generated.

- (original user program name)_o.850

  This is an intermediate output file in which the original user program (C 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 user program when execution module (tuned execution module) is generated.

- (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 user program when execution module (tuned execution module) is generated.

4.6 Timer Control Code Block

TW 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 or tuned, a processing routine (timer control code block) that initializes the timer/counter so that it is used with TW is necessary.

Caution TW supplies a sample program of the timer control code block.

[NEC tool chain-supporting version]

nectools32\smp850\tw\TWTIMER85.S	:	Timer control code block for V853 TM11
nectools32\smp850\tw\TWTIMER85E.S	:	Timer control code block for V850E/MS1 TM10

[GHS tool chain-supporting version]

nectools32\smp850_ghs\tw\TWTIMER85.850	:	Timer control code block for V853 TM11
nectools32\smp850_ghs\tw\TWTIMER85E.850	:	Timer control code block for V850E/MS1 TM10

4.7 System Library for TW

With the NEC tool chain, TW reallocates not only the user program but also the standard library functions (included in libc.a and libm.a) during tuning. It therefore supplies libtw.a, which can take the place of libc.a and libm.a. This libtw.a is necessary for generating an execution module with TW.

With the GHS tool chain, TW uses the library functions supplied with NEC's C compiler package CA850, except the functions related to memory allocation, file manipulation, and those not included in the system library of C compiler package CCV850/CCV850E of Green Hills Software, Inc., as libtw.a.

Table 4-1 lists the functions included in the system library (for GHS tool chain) of TW.

Table 4-1 System Library for TW

Functional outline	Function name					
Character string/memory management	bcmp, memset, strcspn, strrchr,	bcopy, index, strerror, strspn,	memchr, rindex, strlen, strstr,	memcmp, strchr, strncmp, strtok	memcpy, strcmp, strncpy,	memmove, strcpy, strpbrk,
Macro/function of character type	toascii, iscntrl, isspace,	tolower, isdigit, isupper,	toupper, isgraph, isxdigit	isalnum, islower,	isalpha, isprint,	isascii, ispunct,
Utility	abs, strtoul, _ _ _ftoi,	labs, rand, _ _ _ftou	bsearch, srand,	div, qsort,	atoi, _ _ _itof,	strtol, _ _ _utof,
Non-local branch	setjmp	longjmp				
Mathematical processing	expf, ldexpf, asinf,	logf, matherr, atanf,	log10f, coshf, atan2f,	powf, sinhf, cosf,	sqrtf, tanhf, sinf,	frexpf, acosf, tanf
Floating-point operation	_ _ _fadd,	_ _ _fsub,	_ _ _fmul,	_ _ _fdiv,	_ _ _fcmp	

Caution The result of operations using float-type infinity or Not-a-Number differs between the system library libtw.a supplied by TW (for GHS tool chain) and the system library supplied by the C compiler package CCV850/CCV850E of Green Hills Software, Inc (U.S.A).

4.8 Notes on Assembly Source File

4.8.1 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, TW recognizes the label declarations that satisfy any of the following conditions as functions subject to performance analysis or tuning.

The functions TW could not recognize are not displayed on the Function Information Table Window.

1) Global label declaration

TW 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.

[With NEC tool chain]

```
        .globl  _func                   --global label declaration
_func :
```

[With GHS tool chain]

```
        .global _func                   # global label declaration
_func :
```

2) Label declaration with function attribute

TW 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.

[With NEC tool chain]

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

[With GHS tool chain]

```
        .type   _func , @function       # 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)

TW 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.

[With NEC tool chain]

```
        .text
        .globl  _func1
_func1 :

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

[With GHS tool chain]

```
        .text
        .global  _func1
_func1 :

        .global  _func2
_func2 :                                # Not recognized as target function
```

2) Instruction immediately before label declaration (other than quasi directive or unconditional branch instruction)

TW 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.

[With NEC tool chain]

```
        .text
        .globl   _func1
_func1 :
        nop

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

[With GHS tool chain]

```
        .text
        .global  _func1
_func1 :
        nop

        .global  _func2
_func2 :                                # Not recognized as target function
```

4.8.2 Detection of location calling function

TW 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 TW could not recognize are not displayed in the Callgraph Window.

1) Issuance of branch instruction jarl

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

[With NEC tool chain]

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

[With GHS tool chain]

```
        .global _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)

TW 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.

[With NEC tool chain]

```
        .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
```

[With GHS tool chain]

```
        .global _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
        movea   xxx , xxx , r31         # Issuance of arithmetic instruction movea
        jmp     REGISTER                # Issuance of branch instruction jmp

_label3 :                               # Label declaration
        movea   xxx , xxx , lp          # Issuance of arithmetic instruction movea
        jmp     REGISTER                # Issuance of branch instruction jmp

_label4 :                               # 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)

TW 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.

[With NEC tool chain]

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

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

_label2 :                               -- Label declaration
```

[With GHS tool chain]

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

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

_label2 :                               # Label declaration
        mov     xxx , r31               # Issuance of arithmetic instruction mov
        jmp     REGISTER                # Issuance of branch instruction jmp

_label3 :                               # Label declaration
        mov     xxx , lp                # Issuance of arithmetic instruction mov
        jmp     REGISTER                # Issuance of branch instruction jmp

_label4 :                               # Label declaration
```

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

4.8.3 Description format of assembly source file

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

Table 4-2 shows the assembly source files described in specific formats TW can and cannot recognize as subject to performance analysis (trace data analysis method and probe code insertion method) or tuning.

Table 4-2 Restrictions on Assembly Source File

Control Item	Trace	Probe	Tuning
Accessing higher address of function label by offset	O	X	X
Assembly quasi directive with changing conditions	X	X	X
Formal parameter outside target range (with NEC tool chain)	X	X	X
Defined macro call (with NEC tool chain)	X	X	X
Multiple expansion of macro definition exceeding 32 stages	X	X	X
Numeric expression or operation result in quasi directive outside valid range	-	-	-
Block declaration straddling over blocks	X	X	X
Macro definition matching assembler instruction mnemonic or quasi directive (with GHS tool chain)	X	X	X
Boundary of function	-	-	-
Branch to outside of function referencing label in function	O	X	X
Recognition of post-declaration function label with two or more function entrances	X	X	X
Offset branch instruction straddling function calling instructions	O	X	O
Reference/branch of label outside function in instruction that cannot straddle sections	O	O	X
Address calculation using label in different functions	O	O	X
Non-initialization of stack pointer	O	X	O
Use of register that is saved in case of interrupt	O	X	O
Use of I/O register for usage other than that assumed (with NEC tool chain)	X	X	X
Concatenation operator for label declaration, conditional assemble quasi directive, or symbol definition quasi directive	X	X	X
Use of macro reserved by system	X	X	X
Specification of function allocation sequence or function allocation address	O	X	X
Use of label in evaluation expression (with GHS tool chain)	X	X	X

Caution "Function" is the function that satisfies the conditions described in **4.8.1 Detection of functions subject to performance analysis and tuning**.
"Function call" is the function call that satisfies the conditions described in **4.8.2 Detection of location calling function**.

1) Accessing higher address of function label by offset

TW 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.

[With NEC tool chain]

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

[With GHS tool chain]

```
        .text
        .word   0x0
        .global _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

TW 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 .elsif) 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).

[With NEC tool chain]

```
.set    VAR, 0x0

.repeat 0x10
        .if     ( VAR == 0x5 )          -- Assembly quasi directive with changing condition
                nop
        .endif

        .set    VAR, VAR + 0x1
.endm
```

[With GHS tool chain]

```
.set    VAR, 0x0

.rept   0x10
        .if     ( VAR == 0x5 )          # Assembly quasi directive with changing condition
                nop
        .endif

        .set    VAR, VAR + 0x1
.endr
```

3) Formal parameter outside target range (with NEC tool chain)

TW 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).

[With NEC tool chain]

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

MCR1    0x1

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

4) Defined macro call (with NEC tool chain)

TW 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).

[With NEC tool chain]

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

        MCR2
        nop
.endm

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

macro2
```

5) Multiple expansion of macro definition exceeding 32 stages

TW 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.

[With NEC tool chain]

```
.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
```

[With GHS tool chain]

```
.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, TW 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.

[With NEC tool chain]

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

.if     ( ( 0x7fffffff * 0x7fffffff ) == 0x1 )  -- Operation result in quasi directive is outside valid range.
        nop
.endif
```

[With GHS tool chain]

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

.if     ( ( 0x7fffffff * 0x7fffffff ) == 0x1 )  # Operation result in quasi directive is outside valid range.
        nop
.endif
```

7) Block declaration straddling over blocks

TW 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).

[With NEC tool chain]

```
.set    EVAL, 0x1

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

nop

.if     EVAL
        .endm
.endif
```

[With GHS tool chain]

```
.set    EVAL, 0x1

.if     EVAL
        .rept    EVAL           # Repetitive definition block declaration straddling over other blocks
.endif

nop

.if     EVAL
        .endr
.endif
```

8) Macro definition matching assembler instruction mnemonic or quasi directive (with GHS tool chain)

TW 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 macro definition using the same macro name as an assembler instruction mnemonic or quasi directive is made.

[With GHS tool chain]

```
.macro  nop                 # Definition of macro name matching assembler instruction mnemonic
        jarl    _func, lp
.endm

nop
```

9) Boundary of function

TW 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.

[With NEC tool chain]

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

[With GHS tool chain]

```
        .align  0x4         # Boundary of function
        .global _func
_func :
        nop
```

10) Branch to outside of function referencing label in function

TW 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.

[With NEC tool chain]

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

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

[With GHS tool chain]

```
        .global _func1
_func1 :
        jr      _func1 + 0x6    # Branch outside function referencing label in function
        jmp     [ lp ]

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

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

TW 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.

[With NEC tool chain]

```
        .globl  _func1
_func1 :
        nop

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

[With GHS tool chain]

```
        .global _func1
_func1 :
        nop

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

12) Offset branch instruction straddling function calling instructions

TW 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.

[With NEC tool chain]

```
        .globl  _func1
_func1 :
        br      _fun1 + 0x6             -- Offset branch instruction straddling function call instruction
        jarl    _func2, r11
        jmp     [ lp ]

        .globl  _func2
_func2 :
        jmp     [ r11 ]
```

[With GHS tool chain]

```
        .global _func1
_func1 :
        br      _fun1 + 0x6             # Offset branch instruction straddling function call instruction
        jarl    _func2, r11
        jmp     [ lp ]

        .global _func2
_func2 :
        jmp     [ r11 ]
```

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

TW 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.

[With NEC tool chain]

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

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

[With GHS tool chain]

```
        .global _func1
_func1 :
        jbr     _func2              # Branch to label outside function
        jmp     [ lp ]

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

14) Address calculation using label in different functions

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

[With NEC tool chain]

```
        .globl  _func1
_func1 :
        mov     #data - _#func2, r11    -- Address calculation using label in different functions
        jmp     [ lp ]

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

[With GHS tool chain]

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

        .global _func2
_func2 :
        jmp     [ lp ]
data :
        .word   0x0
```

15) Non-initialization of stack pointer

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

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

TW 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.

17) Use of I/O register for usage other than that assumed (with NEC tool chain)

TW 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, TW 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.

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

TW 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 concatenation operator in label declaration, conditional assembly quasi directive, or symbol definition quasi directive.

[With NEC tool chain]

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

[With GHS tool chain]

```
.macro   labeldef  ARG
         .global   _func><ARG
         _func><ARG :             # Concatenation operator for label declaration
.endm
```

19) Use of macro reserved by system

TW treats words starting with "tw_" or ".." and "__e_sysfnc" as system-reserved macros. Therefore, TW 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.

[With NEC tool chain]

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

[With GHS tool chain]

```
         .global   tw_func              # Use of system-reserved macro
tw_func :
         jmp       [ lp ]
```

20) Specification of function allocation sequence or function allocation address

TW 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.

[With NEC tool chain]

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

_func2 :
        jmp     [ lp ]
```

[With GHS tool chain]

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

_func2 :
        jmp     [ lp ]
```

21) Use of label in evaluation expression (with GHS tool chain)

TW cannot execute performance analysis (trace data analysis method or probe code insertion method) and tuning processing on an assembly source file that uses a label as an address in the evaluation expression of a symbol assignment instruction, conditional assemble quasi directive, and repeat quasi directive.

[With GHS tool chain]

```
        .global _func1
_func1 :
        jmp     [ lp ]

_func2 :
        .if     ( ( _func2 - _func1 ) >= 0x2 )      # Use of label in evaluation expression
                nop
        .endif
```

CHAPTER 5 OPERATION

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

5.1 Overview

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

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

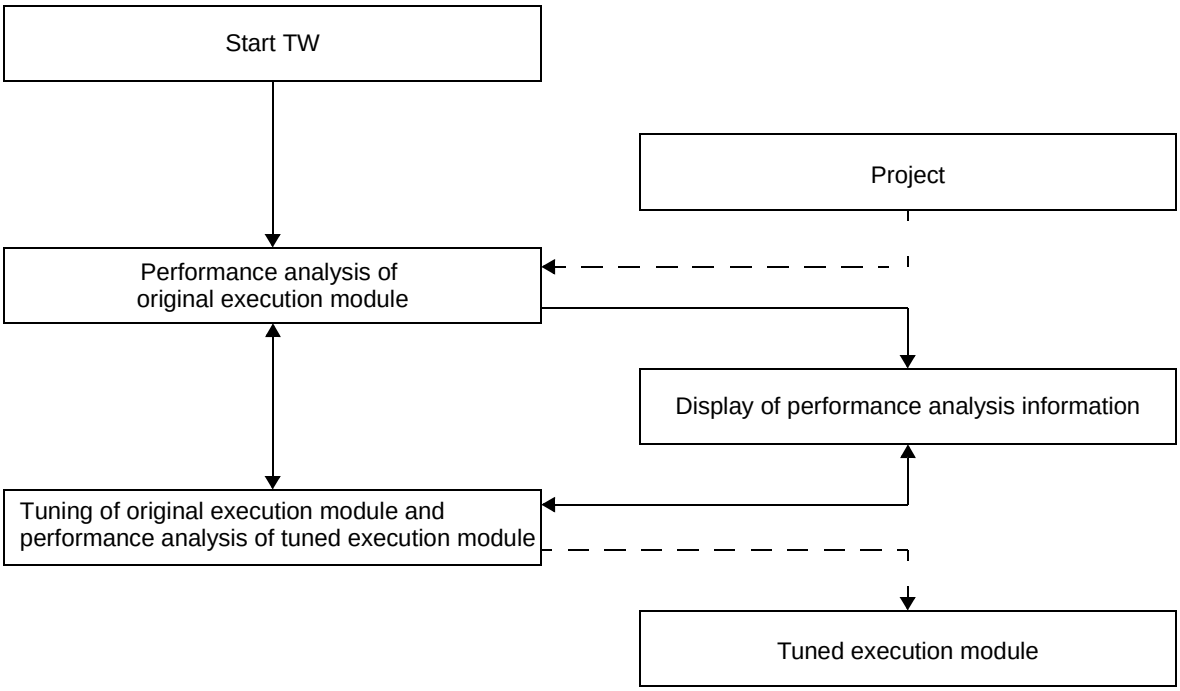
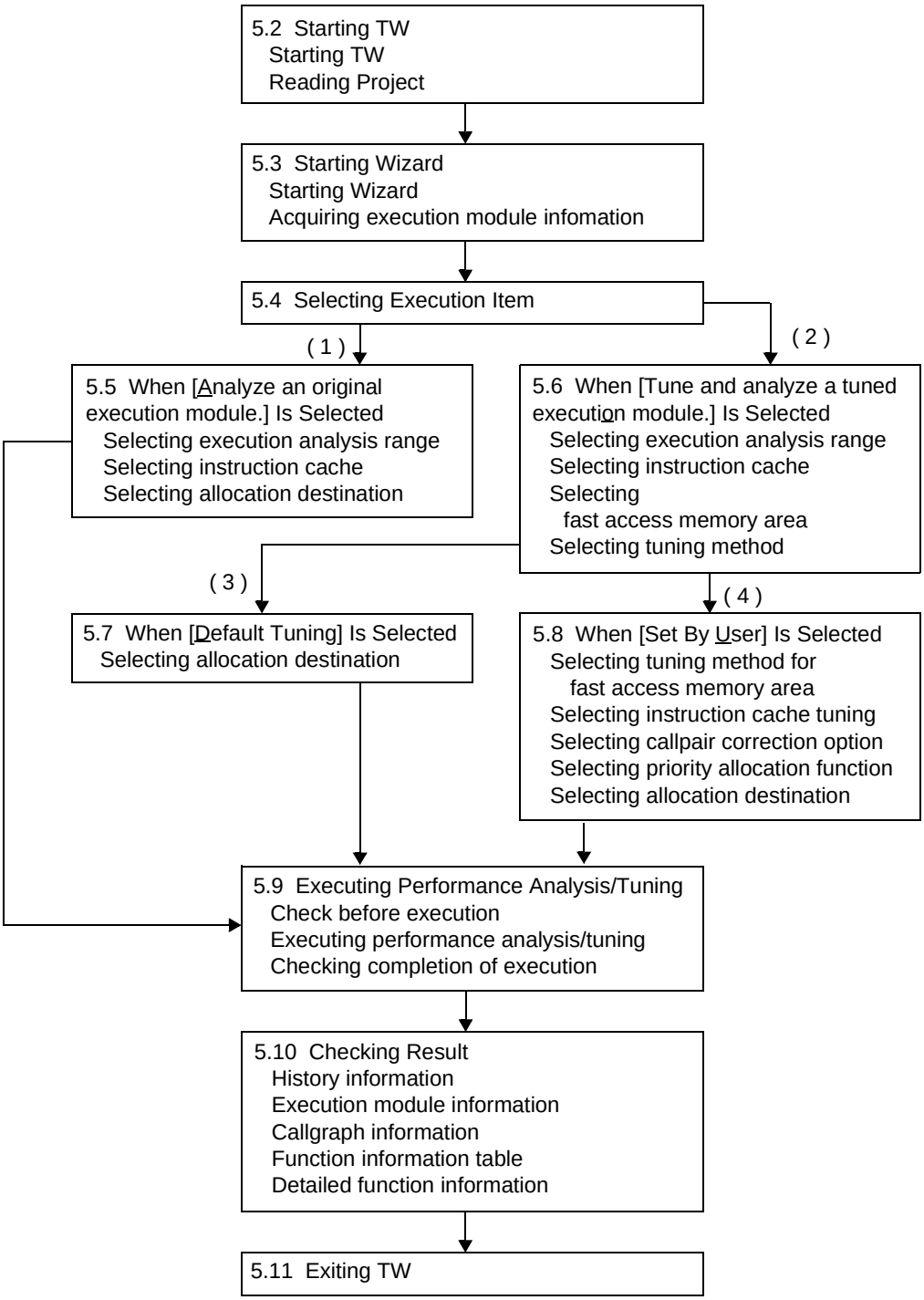


Figure 5-2 shows the procedure for operating TW and checking the performance analysis results.

Figure 5-2 Procedure for Operating TW/Checking Performance Analysis Results



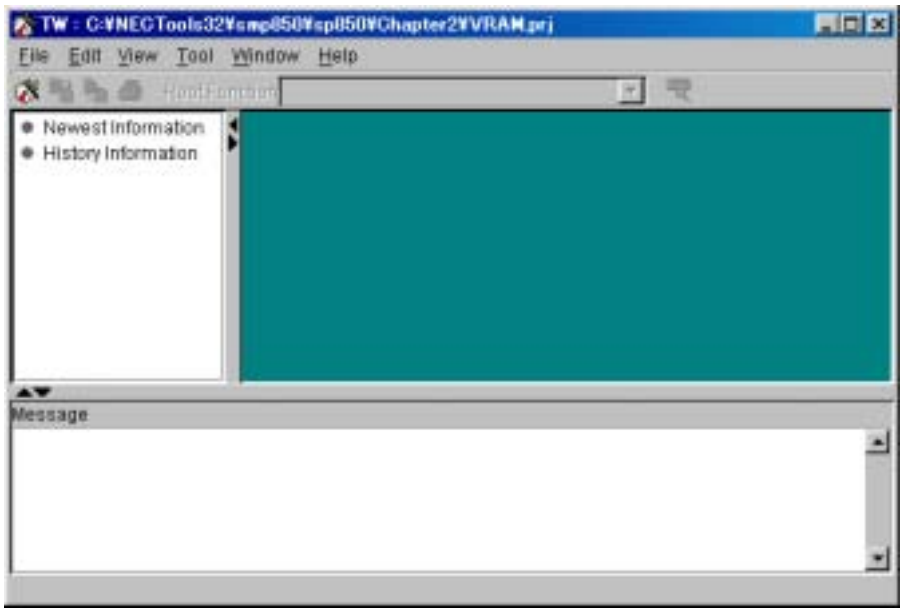
1) If only the [Analyze an original execution module.] check box is checked when selecting an execution item
2) If the [Tune and analyze a tuned execution module.] check box is checked when selecting an execution item
3) If the [Default Tuning] radio button is checked when selecting a tuning method
4) If the [Set By User] radio button is checked when selecting a tuning method

5.2 Starting TW

5.2.1 Starting TW

For details of starting TW, refer to **CHAPTER 3 STARTING AND TERMINATING**.
When TW has been correctly started, the following Main Window is automatically opened.

Figure 5-3 Main Window



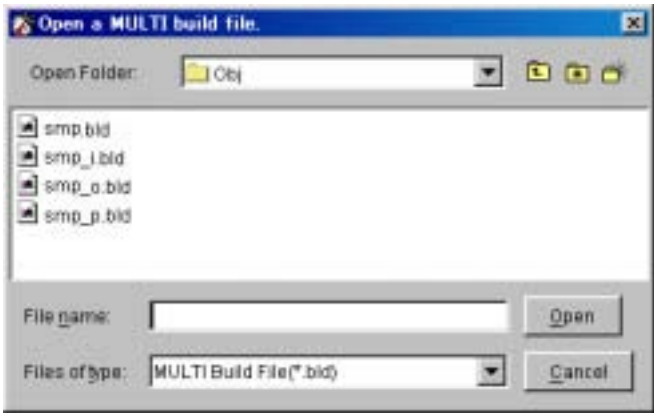
5.2.2 Reading project

If TW is the version supporting the NEC tool chain, TW recognizes a project (project file) that is active in PM plus as the target project.

If TW is the version supporting the GHS tool chain, however, an operation to read a project (build file) is necessary.

To read a build file, select [File] menu -> [Open MULTI Project...] on the Main Window, and open the following File Selection Dialog Box.

Figure 5-4 File Selection Dialog Box



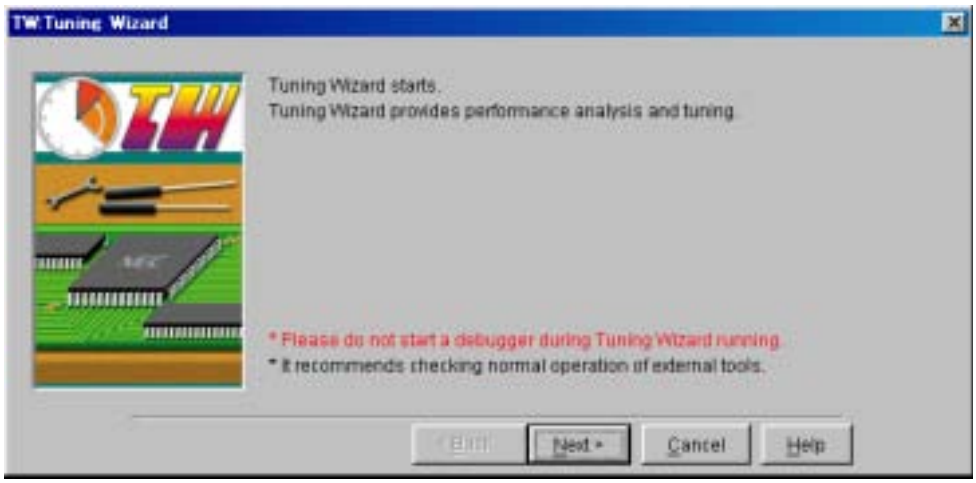
In the above dialog box, specify the file (build file) necessary for TW to execute performance analysis/tuning processing, and click the <Open> button.

5.3 Starting Wizard

5.3.1 Starting wizard

Click the <Tuning Wizard> button or select [Tool] menu -> [Tuning Wizard...] on the Main Window.
If the TW wizard is correctly started, the following Initial Screen Panel is automatically opened.

Figure 5-5 Initial Screen Panel



To continue the wizard processing (performance analysis or tuning), confirm that a message reporting the start of TW wizard processing (performance analysis or tuning) is displayed in the message display area, and click the <Next>> button.

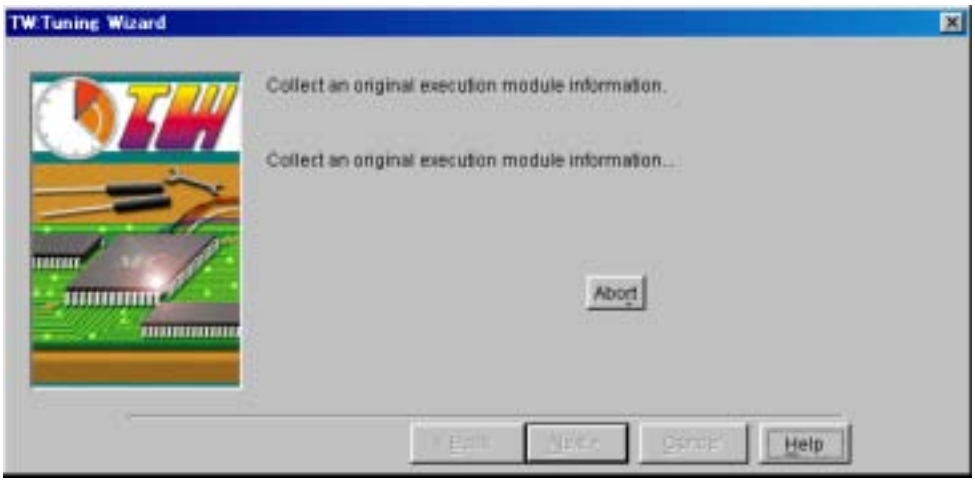
5.3.2 Acquiring execution module information

Acquire the execution module information necessary for TW to execute performance analysis processing from the user program or execution module (execution module for original analysis) described in the project (project file or build file) read in **5.3.1 Starting wizard**.

However, the user does not have to be aware of this phase because it is automatically performed by TW when the <Next>> button on the Initial Screen Panel is clicked in **5.3.1 Starting wizard**.

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

Figure 5-6 Execution Module Information Acquisition Panel

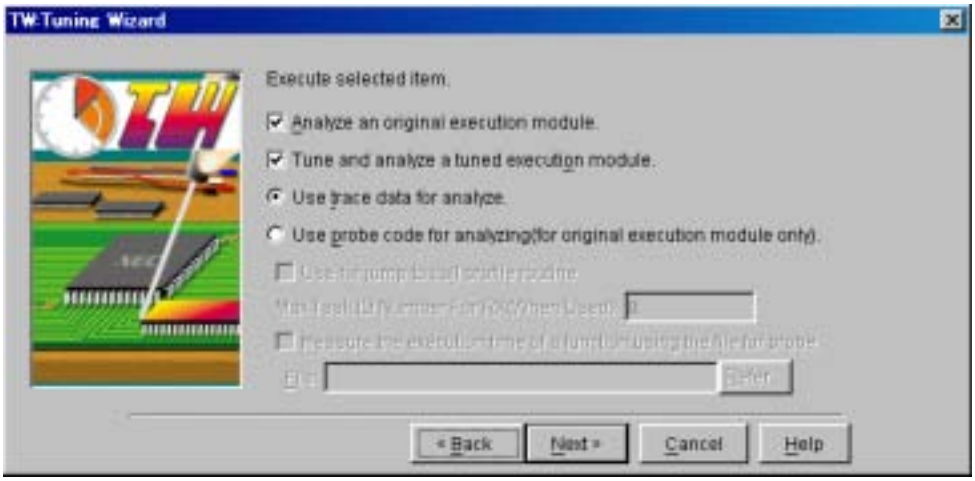


5.4 Selecting Execution Items

Specify the items to be executed in the 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 **5.3.2 Acquiring execution module information** has been terminated correctly.

Figure 5-7 Execution Item Selection Panel



On this panel, the following items can be specified.

(1) 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 the original execution module as wizard processing of TW.

When this check box is checked, performance analysis is executed on the original execution module.

(b) [Tune and analyze a tuned execution module.] check box

This check box is used to specify whether tuning processing (generation processing of tuned execution module) is performed on the original execution module and whether performance analysis processing is performed on the tuned execution module, as wizard processing of TW.

When this check box is checked, tuning processing (generation processing of tuned execution module) is performed on the original execution module, and performance analysis processing is performed on the tuned execution module.

(2) Performance analysis information acquisition method specification area

This area is used to set a method of acquiring performance analysis information of wizard processing (performance analysis or tuning).

This specification area consists of the following radio buttons.

(a) [Use trace data for analyze.] radio button

This button is used to specify whether the trace data analysis method is used to acquire performance analysis information of wizard processing.

If this radio button is checked, the trace data analysis method is used to acquire performance analysis information.

(b) [Use probe code for analyzing(for original execution module only).] radio button

This button is used to specify whether the probe code insertion method is used to acquire performance analysis information of wizard processing.

If this radio button is checked, the probe code insertion method is used to acquire performance analysis information.

(3) Probe code insertion method option specification area

This area is used to set the options of the probe code insertion method required when the [Use probe code for analyzing(for original execution module only).] radio button is checked.
This specification area consists of the following check box, specification area, and buttons.

(a) [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 the profile routine that is output as an intermediate file as a result of the wizard processing (tuning) of TW.
If this check button is checked, the far jump instruction is used to call the profile routine.

(b) [Max Task ID Number For RX(When Used)] specification area

This area is used to specify the maximum task ID number for profile that is required for TW to execute wizard processing (performance analysis processing: RTOS information acquisition processing).
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.

Caution The value specified in this specification area is meaningful only when all the following conditions are satisfied.

- If the [Use probe code for analyzing(for original execution module only).] radio button is checked on the Execution Item Selection Panel
- If the functions supplied RTOS (NEC's RX850 or RX850 Pro) are used in the user program

(c) [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 of a function subject to performance analysis is measured as wizard processing of TW (performance analysis processing).
If this check box is checked, "execution time of the function is measured".

(d) Time measurement file name specification area

This area is used to specify the file name of a file for measuring time (timer control code block) necessary for TW to execute wizard processing (performance analysis processing: execution time measurement of function subject to performance analysis).

Caution 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 checked

(e) <Refer...> button

This button opens the File Selection Dialog Box.

Caution 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

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

5.5 When [Analyze an original execution module.] Is Selected

The operation procedure if only the [Analyze an original execution module.] check box is checked in **5.4 Selecting Execution Items** is explained below.

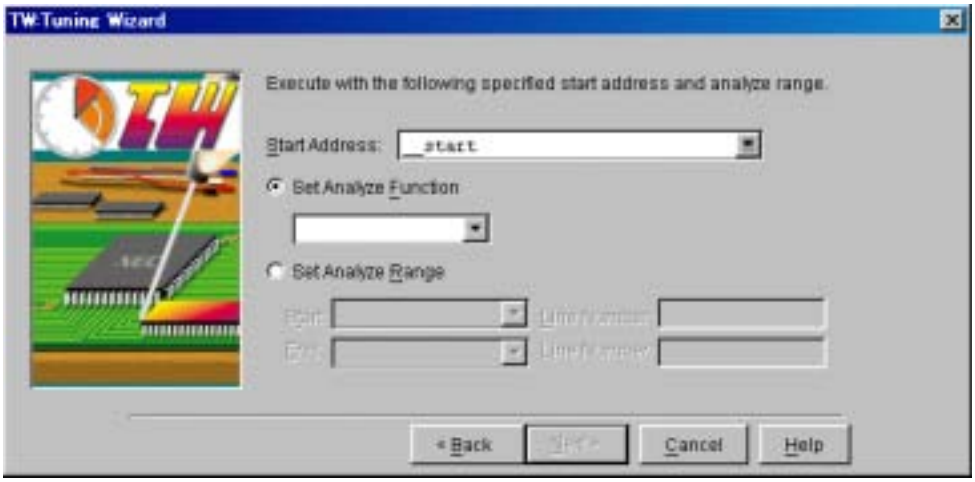
For the operation procedure if the [Tune and analyze a tuned execution module.] check box is checked in **5.4 Selecting Execution Items**, refer to **5.6 When [Tune and analyze a tuned execution module.] Is Selected**.

5.5.1 Selecting execution analysis range

Specify the execution start position of the user program and execution module, and the performance analysis range for wizard processing (performance analysis processing).

The Execution Analysis Range Selection Panel shown in the figure below is opened when each item has been specified in **5.4 Selecting Execution Items** and then <Next>> button has been clicked.

Figure 5-8 Execution Analysis Range Selection Panel



This panel is used to specify the following items.

(1) [Start Address] combo box

This combo box is used to select the execution start position of the user program and execution module.

(2) Performance analysis target function specification area

This area is used to specify the performance analysis target range for wizard processing (performance analysis). This specification area consists of the following radio button and specification area.

(a) [Set Analyze Eunction] radio button

This button is used to specify whether the performance analysis target range for wizard processing (performance analysis) is specified by a function name.
If this button is checked, the performance analysis target range is specified by a function name.

(b) Performance analysis target function name specification combo box

This box is used to select the performance analysis target range (function name) for wizard processing (performance analysis).

Caution TW 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.

(3) Performance analysis target range specification area

This area is used to specify the performance analysis target range for wizard processing (performance analysis).

This specification area consists of the following radio button, combo box, and specification area.

(a) [Set Analyze Range] radio button

This button is used to specify whether the performance analysis target range for wizard processing (performance analysis) is specified by a file name, symbol name, or line number.
If this radio button is checked, the performance analysis target range is specified by a file name, symbol name, or line number.

(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).

Caution This specification area is invalid if the following condition is satisfied.

- If the [Set Analyze Range] radio button is not checked

(c) [Line Number] specification area

This area is used to select the performance analysis start position (line number) for wizard processing (performance analysis).

Caution This specification area is invalid if the following condition is satisfied.

- If nothing is input to the [Start] combo box.

(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).

Caution This specification area is invalid if the following condition is satisfied.

- If the [Set Analyze Range] radio button is not checked

(e) [Line Number] specification area

This area is used to select the performance analysis end position (line number) for wizard processing (performance analysis).

Caution This specification area is invalid if the following condition is satisfied.

- If nothing is input to the [End] combo box

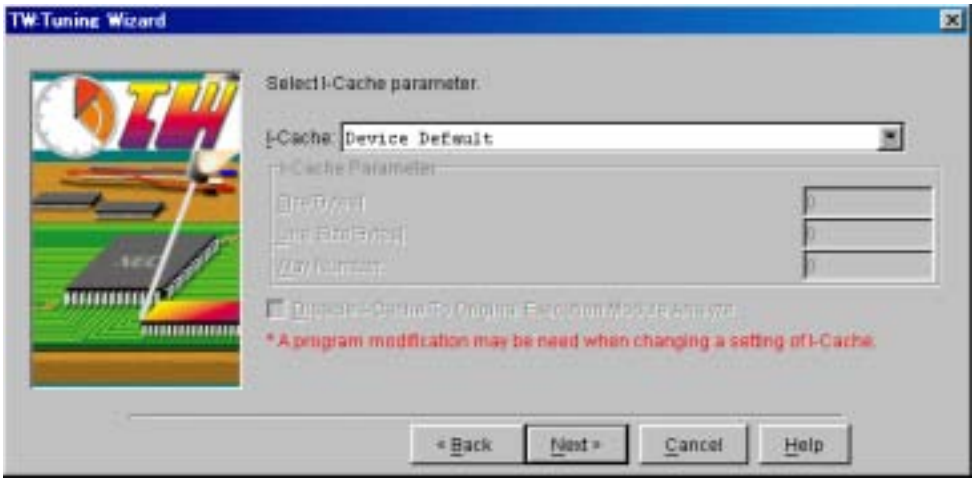
To continue wizard processing (performance analysis), specify each item and then click the <Next> > button.

5.5.2 Selecting instruction cache

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

The Instruction Cache Selection Panel shown in the figure below is opened when each item is specified in **5.5.1 Selecting execution analysis range** and then the <Next>> button is clicked.

Figure 5-9 Instruction Cache Selection Panel



This panel is used to specify the following items.

(1) [I-Cache] combo box

This combo box is used to select the type of the instruction cache used for wizard processing (performance analysis).

Caution TW can set the instruction cache selected in this combo box as the execution engine, but does not set the instruction cache as described in the "Instruction Cache, Data Cache NB85E, NB85ET User's Manual". To use the instruction cache, therefore, processing to set the instruction cache must be added to the startup routine.
With an NEC simulator, the instruction cache does not have to be set.

(2) Instruction cache parameter specification area

This area is used to specify the parameters of the instruction cache used for wizard processing (performance analysis).
This specification area consists of the following specification areas.

(a) [Size] specification area

This area is used to specify the size (in bytes) of the instruction cache used for wizard processing (performance analysis). 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 the number of ways specification area".

Cautions 1. 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

2. 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 simulator.

(b) [Line Size] 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).

Caution 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).

Cautions 1. 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

2. 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 simulator.

(3) [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.
If this check box is checked, the instruction cache is made invalid when the performance of the original execution module is analyzed.

Caution This check box is invalid if the following condition is satisfied.

- If the size of the instruction cache is 0

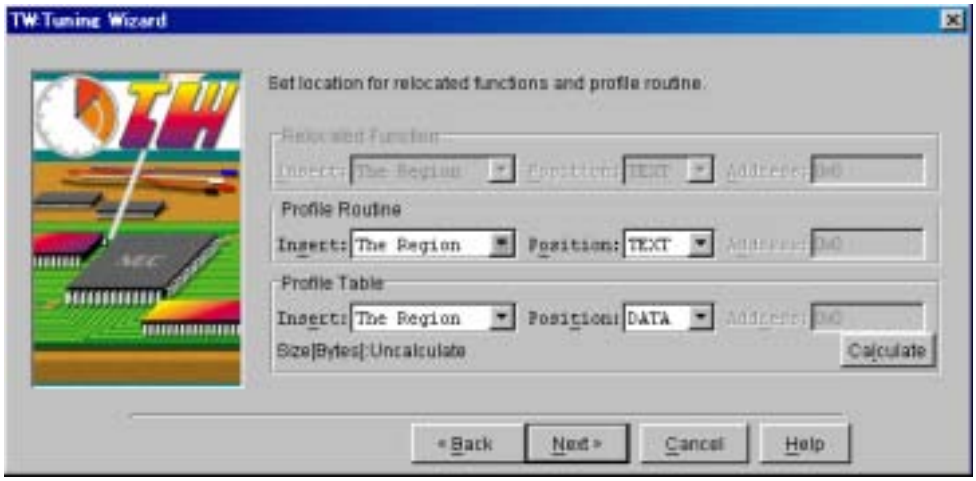
To continue wizard processing (performance analysis), specify each item and then click the <Next> > button.

5.5.3 Selecting allocation destination

Specify the memory area where the profile routine and profile table output as intermediate files when the performance is analyzed are to be allocated.

The Allocation Destination Selection Panel shown in the figure below is opened when each item is set in **5.5.2 Selecting instruction cache** and then the <Next>> button is clicked.

Figure 5-10 Allocation Destination Selection Panel



This panel is used to specify the following items.

(1) Reallocation target function specification area

This specification area becomes invalid if the [Analyze an original execution module.] check box is checked in **5.4 Selecting Execution Items**.

(2) Profile routine specification area

This area is used to specify the memory location where the profile routine output as an intermediate file when the performance is analyzed is to be allocated.
This specification is valid only when the [Use probe code for analyzing(for original execution module only).] radio button is checked in **5.4 Selecting Execution Items**.
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 (in specified area, before specified area, after specified area, or at specified address) of the memory area for allocating the profile routine output as an intermediate file when the performance is analyzed.

Caution The "The Region" menu of this combo box can be selected only when the user's development environment is the NEC tool chain.
Menus other than the "The Region" menu of 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 when the performance is analyzed.

Caution This combo box is invalid if the following condition is satisfied.

- If the TWROUTINE segment is defined for the target link directive file

(c) [Adddress] 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 when the performance is analyzed.

Caution This specification area is invalid if the following condition is satisfied.

- If a menu other than "Set Address" is selected in the [Insert] combo box

(3) 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 when the performance is analyzed.
This specification area is valid only when the [Use probe code for analyzing(for original execution module only).] radio button is checked in **5.4 Selecting Execution Items**.
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 (in specified area, before specified area, after specified area, or at specified address) of the memory area for allocating the profile table output as an intermediate file when the performance is analyzed.

Caution The "The Region" menu of this combo box can be selected only when the user's development environment is the NEC tool chain.
Menus other than the "The Region" menu of this combo box are invalid if the following condition is satisfied.

- 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 when the performance is analyzed.

Caution This combo box is invalid if the following condition is satisfied.

- If the TWTABLE segment is defined for the target link directive file

(c) [Adddress] specification area

This area is used to specify the insertion target area (address) of the memory area for allocating the profile table output as an intermediate file when the performance is analyzed.

Caution This specification area is invalid if the following condition is satisfied.

- If a menu other than "Set Address" is selected in the [Insert] combo box

(d) Size display area

This area displays the size (in bytes) of the profile table that can be acquired by clicking the <Calculate> button.

(e) <Calculate> button

This button calculates the size (in bytes) of the profile table and displays the result of calculation in the size display area.

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

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

This section explains the operating procedure when [Tune and analyze a tuned execution module.] is selected in **5.4 Selecting Execution Items**.

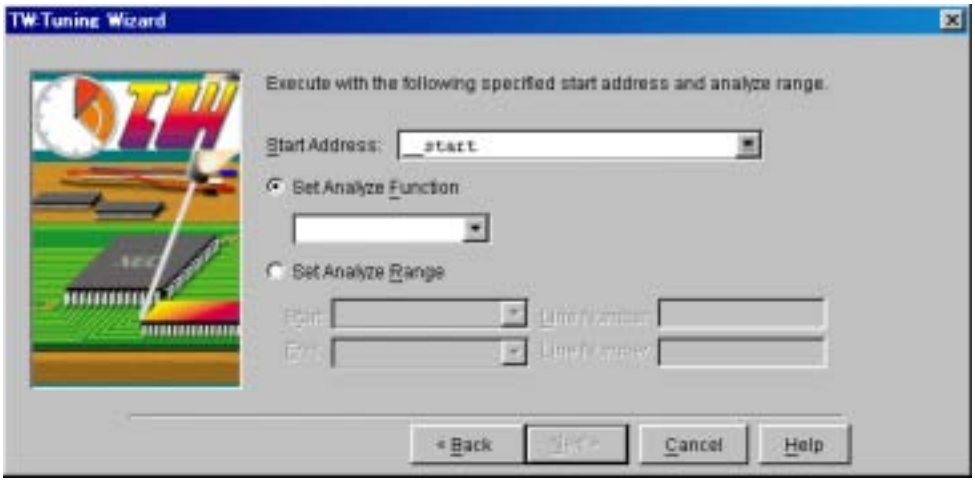
For the operating procedure when the [Analyze an original execution module.] check box is checked in **5.4 Selecting Execution Items**, refer to **5.5 When [Analyze an original execution module.] Is Selected**.

5.6.1 Selecting execution analysis range

Specify the execution start position of the user program and execution module, and the performance analysis range for wizard processing (performance analysis processing).

The Execution Analysis Range Selection Panel shown in the figure below is opened when each item is specified in **5.4 Selecting Execution Items** and then the <Next> > button is clicked.

Figure 5-11 Execution Analysis Range Selection Panel



This panel is used to specify the following items.

(1) [Start Address] combo box

This combo box is used to select the execution start position of the user program and execution module.

Caution 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

(2) Performance analysis target function specification area

This area is used to specify the performance analysis target area for wizard processing (performance analysis). This specification area consists of the following radio button and specification area.

(a) [Set Analyze Function] radio button

This radio button is used to specify whether the performance analysis target range for wizard processing (performance analysis) is specified by a function name.
If this radio button is checked, the performance analysis target range for wizard processing (performance analysis) is specified by a function name.

Caution 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 box is used to select the performance analysis target range (function name) for wizard processing (performance analysis).

Cautions 1. TW 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. This radio 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 range specification area

This area is used to specify the performance analysis target range for wizard processing (performance analysis). This specification area consists of the following radio button, combo box, and specification area.

(a) [Set Analyze Range] radio button

This radio button is used to specify the performance analysis target range for wizard processing (performance analysis) by a file name, symbol name, or line number.
If this radio button is checked, the performance analysis target range for wizard processing (performance analysis) is specified by a file name, symbol name, or line number.

Caution 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).

Caution 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

(c) [Line Number] specification area

This area is used to select the performance analysis start position (line number) for wizard processing (performance analysis).

Caution This combo box is invalid if either of the following conditions is satisfied.

- If nothing is input to the [Start] combo box
- If the [Analyze an original execution module.] check box on the Execution Item Selection Panel is not checked

(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).

Caution 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

(e) [Line Number] specification area

This area is used to select the performance analysis end position (line number) for wizard processing (performance analysis).

Caution This specification area is invalid if either of the following conditions is satisfied.

- If nothing is input to the [End] combo box
- If the [Analyze an original execution module.] check box on the Execution Item Selection Panel is not checked

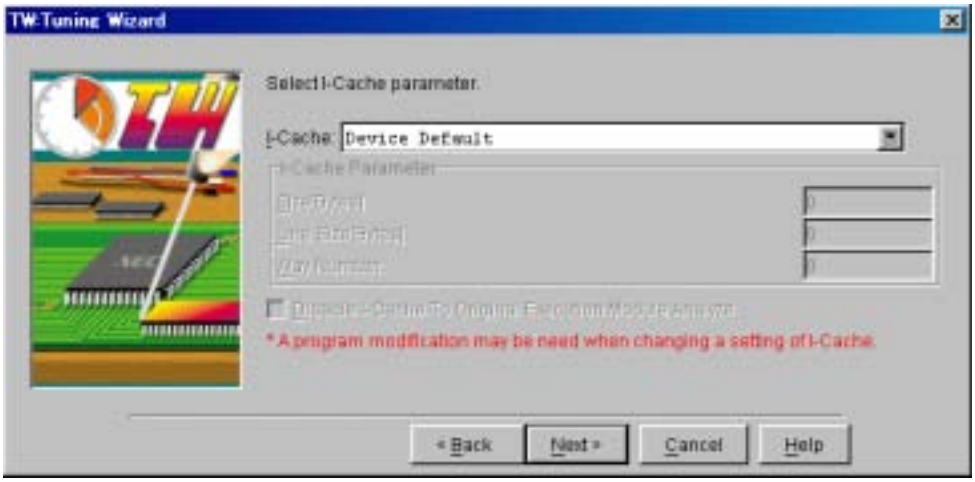
To continue wizard processing (performance analysis), specify each item and then click the <Next>> button.

5.6.2 Selecting instruction cache

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

The Instruction Cache Selection Panel shown in the figure below is opened when each item is specified in **5.4 Selecting Execution Items** and then the <Next>> button is clicked.

Figure 5-12 Instruction Cache Selection Panel



This panel is used to specify the following items.

(1) [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).

Caution TW can set the instruction cache selected in this combo box as the execution engine, but does not set the instruction cache as described in the "Instruction Cache, Data Cache NB85E, NB85ET User's Manual". To use the instruction cache, therefore, processing to set the instruction cache must be added to the startup routine.
With an NEC simulator, the instruction cache does not have to be set.

(2) 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] 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 the number of ways specification area".

Cautions 1. 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

2. 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 simulator.

(b) [Line Size] 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).

Caution 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).

Cautions 1. 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

2. If the number of ways of the instruction cache is set to 1 or less, the value of 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 the tag. This description is irrelevant an NEC simulator.

(3) [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.
If this check box is checked, "the instruction cache is made invalid when the performance of the original execution module is analyzed".

Caution This check box is invalid if either 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

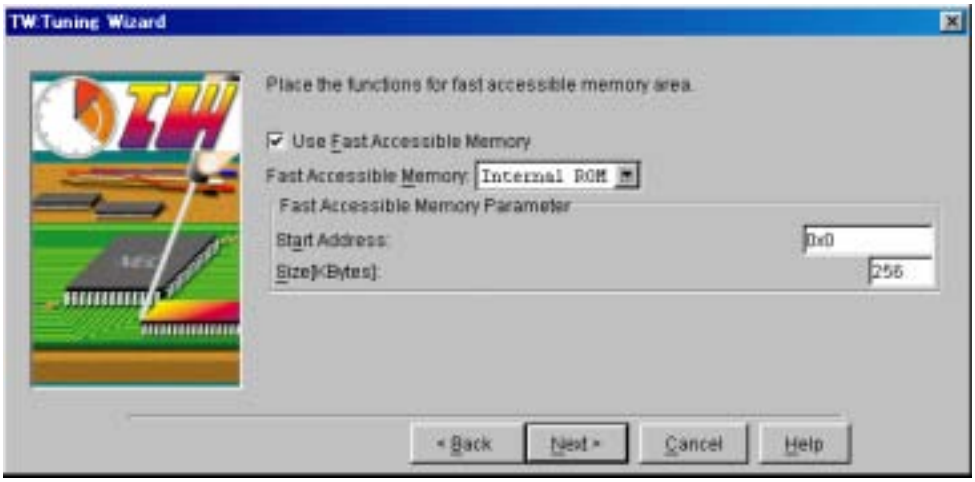
To continue wizard processing (performance analysis), specify each item and then click the <Next>> button.

5.6.3 Selecting fast access memory area

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

The Fast Access Memory Area Selection Panel shown below is opened when <Next>> button is clicked in **5.6.2 Selecting instruction cache**.

Figure 5-13 Fast Access Memory Area Selection Panel



This panel is used to specify the following items.

(1) [Use East Accessible Memory] check box

This check box is used to specify whether a function causing the execution performance of the execution module to drop in wizard processing (tuning) is given priority and allocated to the fast access memory (such as internal ROM). If this check box is checked, "the function causing the execution performance of the execution module to drop in wizard processing (tuning) is given priority and allocated to the fast access memory (such as internal ROM)".

(2) [Fast Accessible Memory] combo box

This combo box is used to specify the type of the fast access memory area used for wizard processing (tuning).

Caution This combo box is invalid if the following condition is satisfied.

- If the [Use East Accessible Memory] check box is not checked

(3) 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).

Caution This specification area is invalid if the following condition is satisfied.

- If the [Use East Accessible Memory] check box is not checked

(b) [Size] specification area

This area is used to specify the size (in Kbytes) of the fast access memory area used for wizard processing (tuning).

Caution This specification area is invalid if the following condition is satisfied.

- If the [Use East Accessible Memory] check box is not checked

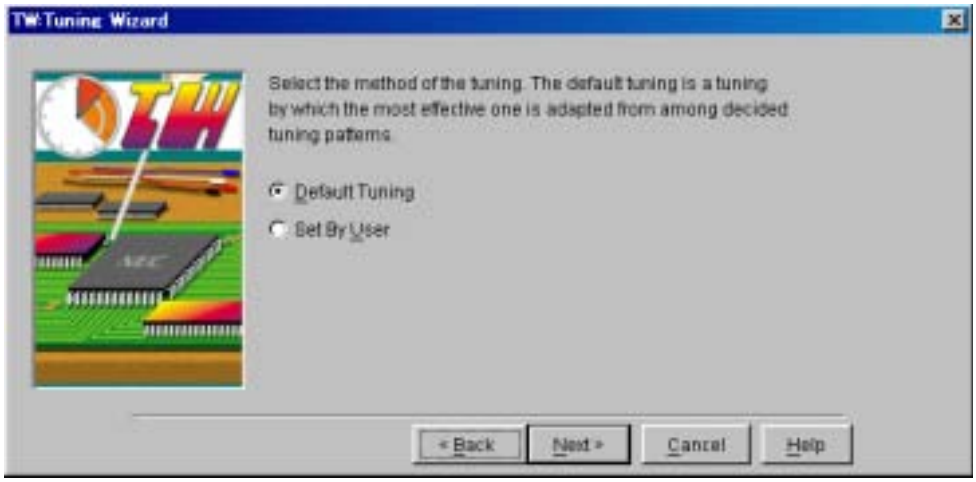
To continue wizard processing (performance analysis or tuning), specify each item and then click the<Next>> button.

5.6.4 Selecting tuning method

Specify the method of tuning the user program and execution module (standard pattern or user-specified pattern).

The Tuning Method Selection Panel shown in the figure below is opened when each item has been selected in **5.6.3 Selecting fast access memory area** and <Next>> button has been clicked.

Figure 5-14 Tuning Method Selection Panel



This panel is used to specify the following items.

(1) Tuning method specification area

This area is used to make setting related to the tuning method of the user program and execution module.
This specification area consists of the following radio buttons.

(a) [Default Tuning] radio button

This radio button is used to specify whether the user program or execution module is tuned by the standard pattern (tuning pattern recommended by TW).
If this radio button is checked, "tuning is performed by the standard pattern (tuning pattern recommended by TW)".

(b) [Set By User] radio button

This radio button is used to specify whether the user program or execution module is tuned by a user-specified pattern. If this radio button is checked, "tuning is performed by a user specified pattern".

Caution "Setting of each tuning item" necessary when this radio button is checked is performed during wizard processing (such as Fast Access Memory Area Tuning Selection Panel, Instruction Cache Tuning Selection Panel, and Callpair Correction Option Selection Panel) that is continued by clicking the <Next>> button.

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

5.7 When [Default Tuning] Is Selected

This section explains the operation procedure when the [Default Tuning] radio button is checked in **5.6.4 Selecting tuning method**.

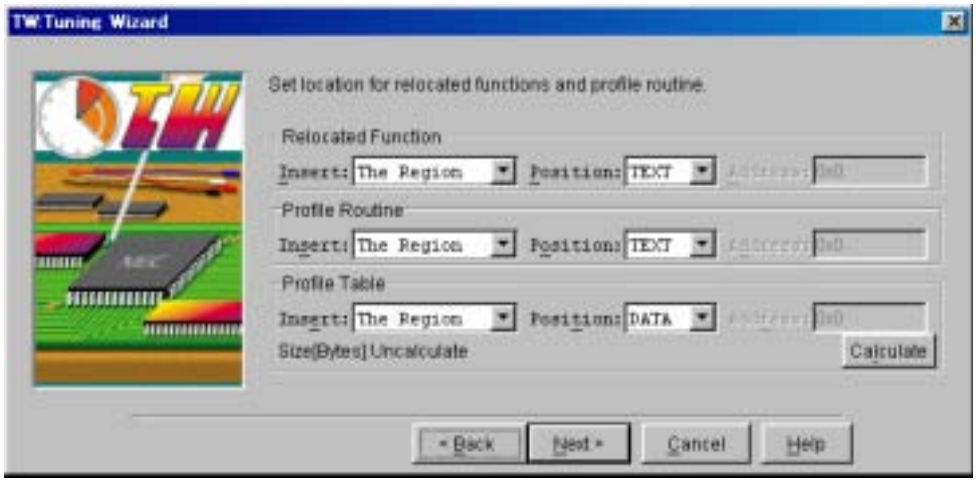
For the operation procedure when the [Set By User] radio button is checked in **5.6.4 Selecting tuning method**, refer to **5.8 When [Set By User] Is Selected**.

5.7.1 Selecting allocation destination

Specify the memory area where the function to be reallocated during tuning, and the profile routine and profile table output as intermediate files during performance analysis or tuning are to be allocated.

The Allocation Destination Selection Panel shown in the figure below is opened when [Default Tuning] check button is checked in **5.6.4 Selecting tuning method** and then <Next>> button has been clicked.

Figure 5-15 Allocation Destination Selection Panel



This panel is used to specify the following items.

(1) Reallocation target function specification area

This specification area is used to make settings related to the reallocation memory area of the function to be reallocated during tuning.

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 (in specified area, before specified area, after specified area, or at specified address) of the memory area for reallocating the function subject to reallocation during tuning

Caution The "The Region" menu of this combo box can be selected only when the user's development environment is the NEC tool chain.
Menus other than the "The Region" menu of 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 reallocating the function subject to reallocation during tuning.

Caution This combo box is invalid if the following condition is satisfied.

- 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 reallocating the function subject to reallocation during tuning.

Caution This specification area is invalid if the following condition is satisfied.

- If a menu other than "Set Address" is selected in the [Insert] combo box

(2) Profile routine specification area

This area is used to make settings related to the memory area for reallocating the profile routine output as an intermediate file during performance analysis or tuning.
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 (in specified area, before specified area, after specified area, or at specified address) of the memory area for allocating the profile routine output as an intermediate file during performance analysis or tuning.

Caution The "The Region" menu of this combo box can be selected only when the user's development environment is the NEC tool chain.
Menus other than the "The Region" menu of this combo box are invalid if either of the following conditions is satisfied.

- If the TWROUTINE segment is defined for the target link directive file
- If the [Use probe code for analyzing(for original execution module only).] radio button on the Execution Item Selection Panel is not checked

(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.

Caution This combo box is invalid if the following condition is satisfied.

- If the TWROUTINE segment is defined for the target link directive file

(c) [Adddress] 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.

Caution This specification area is invalid if the following conditions are satisfied.

- If a menu other than "Set Address" is selected in the [Insert] combo box
- If the [Use probe code for analyzing(for original execution module only).] radio button on the Execution Item Selection Panel is not checked

(3) 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.
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 (in specified area, before specified area, after specified area, or at specified address) of the memory area for allocating the profile table output as an intermediate file during performance analysis or tuning.

Caution The "The Region" menu of this combo box can be selected only when the user's development environment is the NEC tool chain.
Menus other than the "The Region" menu of this combo box are invalid if the following condition is satisfied.

- 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.

Caution This combo box is invalid if the following condition is satisfied.

- If the TWTABLE 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 table output as an intermediate file during performance analysis or tuning.

Caution This specification area is invalid if the following condition is satisfied.

- If a menu other than "Set Address" is selected in the [Insert] combo box

(d) Size display area

This area displays the size (in bytes) of the profile table that can be acquired by clicking the <Calculate> button.

(e) <Calculate> button

This button calculates the size (in bytes) of the profile table and displays the result of calculation in the size display area.

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

5.8 When [Set By User] Is Selected

This section explains the operating procedure when the [Set By User] radio button is checked in **5.6.4 Selecting tuning method**.

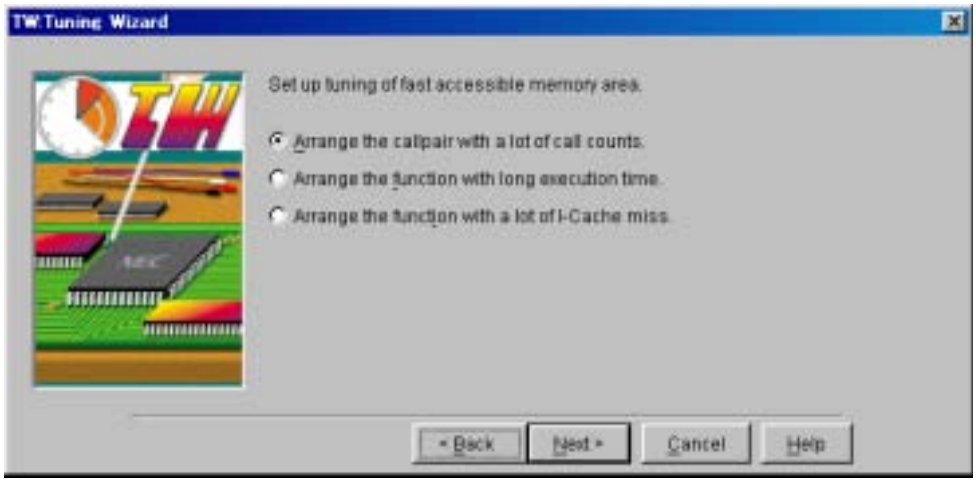
For the operating procedure when the [Default Tuning] radio button is checked in **5.6.4 Selecting tuning method**, refer to **5.7 When [Default Tuning] Is Selected**.

5.8.1 Selecting tuning method for fast access memory area

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

The Fast Access Memory Area Tuning Selection Panel shown in the figure below is opened when the [Set By User] check box is checked in **5.6.4 Selecting tuning method** and then the <Next>> button is clicked.

Figure 5-16 Fast Access Memory Area Tuning Selection Panel



This panel is used to specify the following items.

(1) Function allocation method specification area

This area is used to make settings related to the method of allocating functions to 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 the user program or execution module was analyzed.
This 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, from among the functions that have been called many times (calling functions and called functions), whether a function with a 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 the user program or execution module was analyzed.
When this radio button is checked, from among the functions that have been called many times (calling functions and called functions), a function with a small code size is given priority and allocated to the fast access memory area.

Caution This radio button is invalid if the following condition is satisfied.

- If the [Use East Accessible Memory] radio button 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 the user program or execution module was analyzed.

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.

Cautions 1. This radio button is invalid if all the following conditions are 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.] radio button on the Execution Item Selection Panel is not checked

2. This radio button is invalid if the following condition is satisfied.

- If the [Use Fast Accessible Memory] radio button 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 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 instruction cache hits and misses) of each function that was acquired when the performance of the user program or execution module was analyzed.
If this radio button is checked, a function with many instruction cache misses is given priority and allocated to the fast access memory area.

Caution 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 Fast Accessible Memory] check box on the Fast Access Memory Area Selection Panel is not checked

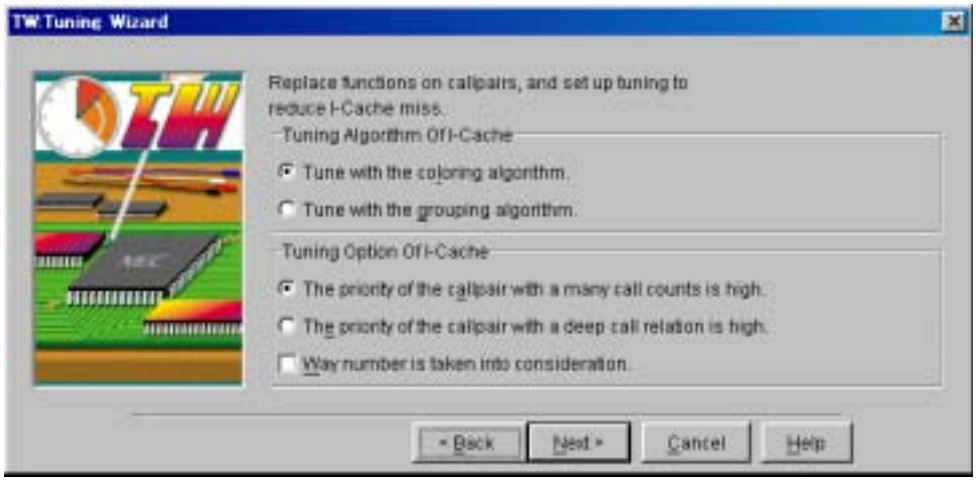
To continue wizard processing (performance analysis or tuning), specify each item and then click the <Next>> button.

5.8.2 Selecting instruction cache tuning

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

The Instruction Cache Tuning Selection Panel shown in the figure below is opened when each item is set in **5.8.1 Selecting tuning method for fast access memory area** and then the <Next>> button is clicked.

Figure 5-17 Instruction Cache Tuning Selection Panel



This panel is used to specify the following items.

(1) 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 the user program or execution module was analyzed.
This specification area consists of the following radio buttons.

(a) [Tune with the coloring algotithm.] radio button

This button is used to specify whether functions are allocated to the instruction cache by a coloring algorithm.
If this radio button is checked, functions are allocated to the instruction cache by a coloring algorithm.

Cautions 1. The coloring algorithm allocates functions to the instruction cache based on the callpair information (calling relationship between functions) of each function.

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.
If this button is checked, functions are allocated to the instruction cache by a grouping algorithm.

Cautions 1. The grouping algorithm groups functions in a close calling relationship and allocates functions to the instruction cache in group units, based on the callpair information (calling relationship between functions) of each function and assuming the size of the instruction cache to be the upper-limit value of the group size.

2. This radio button is invalid if the following condition is satisfied.

- If the size of the instruction cache is 0

(2) 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 the user program or 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 function that is called many times is given priority and allocated to the instruction cache. If this radio button is checked, a function that is called many times is given priority and allocated to the instruction cache.

Caution 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 the function close to the end of a calling relationship is given priority and allocated to the instruction cache.
If this radio button is checked, a function close to the end of a calling relationship is given priority and allocated to the instruction cache.

Caution 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.
If this check box is checked, the cache mechanism (the number of ways) is taken into consideration when a function is allocated to the instruction cache.

Caution This check box is invalid if the following conditions are satisfied.

- If the size of the instruction cache is 0
- If the cache mechanism (number of ways) of the instruction cache is less than 2

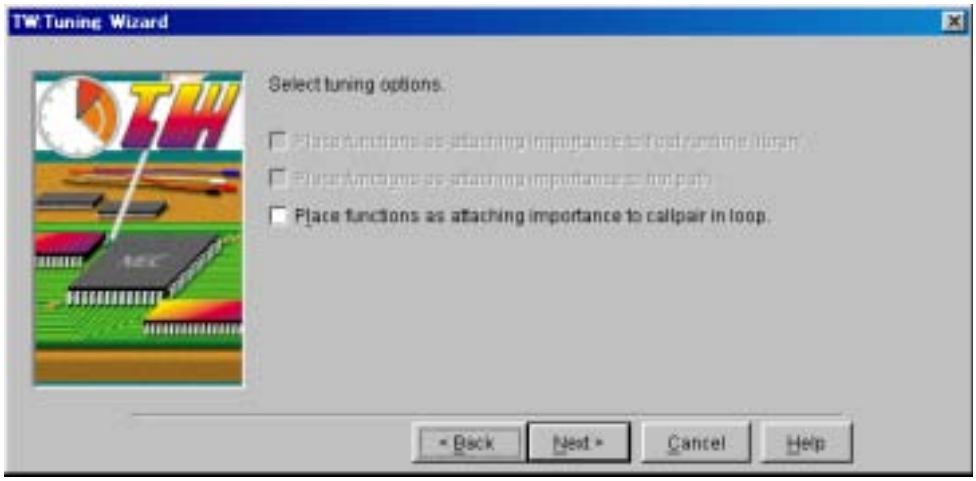
To continue wizard processing (performance analysis or tuning), specify each item and then click the <Next>> button.

5.8.3 Selecting callpair correction option

Specify the tuning method (method of correcting callpair information) of functions to be reallocated, based on the callpair information that was acquired when the performance of the user program or execution module was analyzed.

The Callpair Correction Option Selection Panel shown in the figure below is opened when each item is set in **5.8.2 Selecting instruction cache tuning** and then the <Next>> button is clicked.

Figure 5-18 Callpair Correction Option Selection Panel



This panel is used to specify the following items.

(1) Callpair correction option specification area

This area is used to make settings related to the tuning method (method of correcting callpair information) of the functions to be reallocated, based on the callpair information that was acquired when the performance of the user program or 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 functions (such as expf and cosf) included in the float runtime library are given priority and allocated to the instruction cache or fast access memory area.
If this check box is checked, the functions (such as expf and cosf) included in the float runtime library are given priority and allocated to the instruction cache or fast access memory area.

Cautions 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 all the following conditions are satisfied.

- If the user development environment is the GHS tool chain
- If the [Use libtw] check box in the Option Dialog Box box is not checked.

3. This check box is invalid if the following condition is satisfied.

- If the functions (such as expf and cosf) included in the float runtime library are not included in the user program or execution module

(b) [Place functions as attaching importance to hot path.] check box

This check box specifies whether a function corresponding to a hot path (callpair function related to the applicable function, based on the execution time information of each function, including child functions) is given priority and allocated to the instruction cache or fast access memory area. 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.

Cautions 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.
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.

Caution 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 Fast Access Memory Area Tuning Selection Panel
- If the size of the instruction cache is 0

Caution If two or more check boxes are specified at the same time, TW corrects the callpair information in the following order.

float > hot path > loop processing

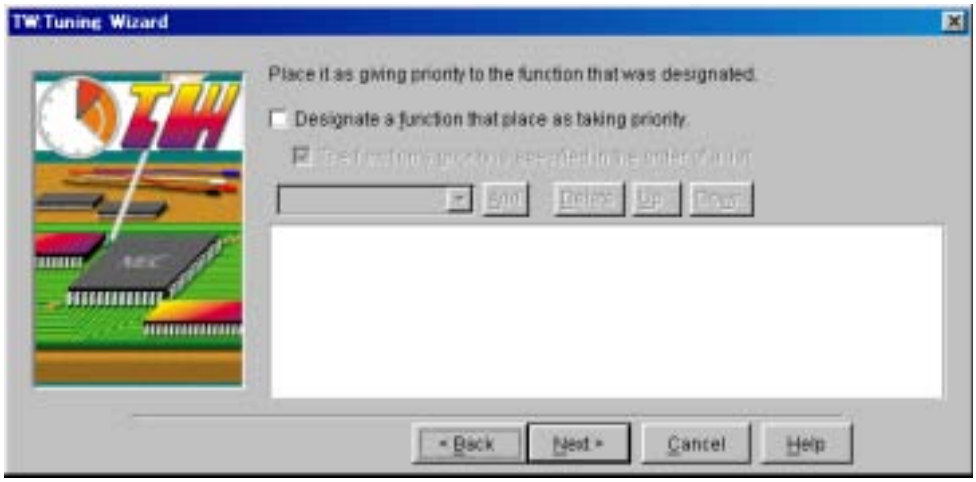
To continue wizard processing (performance analysis or tuning), specify each item and then click the <Next>> button.

5.8.4 Selecting priority allocation function

Specify the function that is given priority and allocated during wizard processing (tuning).

The Priority Allocation Function Selection Panel shown in the figure below is opened when each item is set in **5.8.3 Selecting callpair correction option** and the <Next>> button is clicked.

Figure 5-19 Priority Allocation Function Selection Panel



This panel is used to specify the following items.

(1) [Designate a function that place as taking priority.] check box

This check box is used to specify whether a function that is given priority and allocated is specified for wizard processing (tuning).
If this check box is checked, a function that is given priority and allocated is specified.

(2) [The function's priority is specified in the order of a list.] check box

This check box is used to specify whether function reallocation processing is executed during wizard processing (tuning) in the order in which the functions to be reallocated are displayed on this panel (top: highest priority, bottom: lowest priority).
If this check box is checked, function reallocation processing is executed during wizard processing (tuning) in the order in which the functions to be reallocated are displayed on this panel (top: highest priority, bottom: lowest priority).

Caution 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

(3) Reallocation target function specification area

This area is used to specify the function that is given priority and allocated during wizard processing (tuning).
This specification area consists of the following combo box and buttons.

(a) Reallocation target function name specification combo box

This box is used to select the name of the function that is given priority and allocated during wizard processing (tuning).

Caution 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 registers the function selected in the reallocation target function name specification combo box as a function that is given priority and allocated.

By clicking this button, the name of the function specified in the reallocation target function name specification combo box is added to the lowest position in the reallocation target function display area.

Caution 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 reallocation target function name specification combo box as a function that is given priority and allocated.
By clicking this button, the name of the function specified in the reallocation target function name specification combo box is deleted from the reallocation target function display area.

Caution 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 reallocation target function name specification combo box.
By clicking this button, the position of the name of the function selected in the reallocation target function name specification combo box is moved one level up in the reallocation target function display area.

Caution 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 reallocation target function name specification combo box.
By clicking this button, the position of the name of the function selected in the reallocation target function name specification combo box is moved one level down in the reallocation target function display area.

Caution 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

(4) Reallocation target function display area

This area displays a list of the names of the functions to be given priority and allocated during wizard processing (tuning).

Caution 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

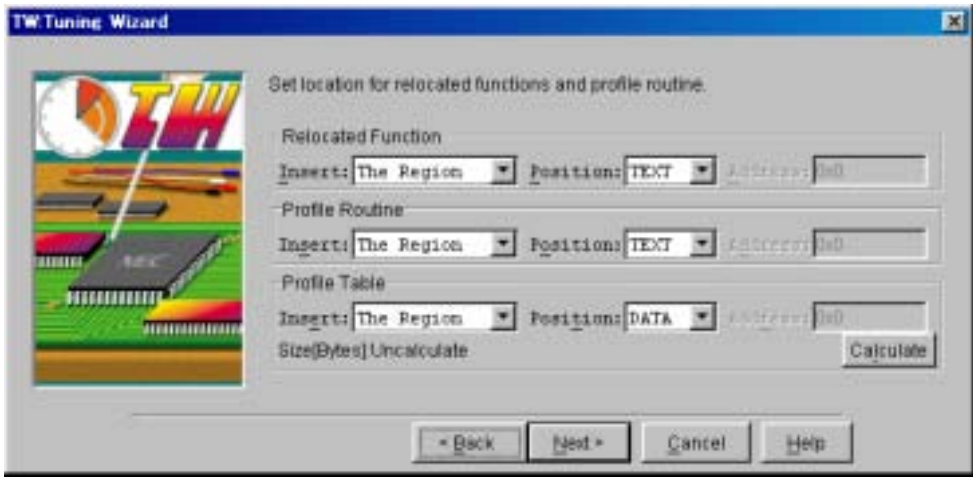
To continue wizard processing (performance analysis or tuning), specify each item and then click the <Next>> button.

5.8.5 Selecting allocation destination

Specify the memory area where the function to be reallocated during tuning, and the profile routine and profile table output as intermediate files during performance analysis are to be allocated.

The Allocation Destination Selection Panel shown in the figure below is opened when each item is specified in **5.8.4 Selecting priority allocation function** and then the <Next>> button is clicked.

Figure 5-20 Allocation Destination Selection Panel



This panel is used to specify the following items.

(1) Reallocation target function specification area

This specification area is used to make settings related to the reallocation memory area of the function to be reallocated during tuning.
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 (in specified area, before specified area, after specified area, or at specified address) of the memory area for reallocating the function subject to reallocation during tuning.

Caution The "The Region" menu of this combo box can be selected only when the user's development environment is the NEC tool chain.
Menus other than the "The Region" menu of 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 reallocating the function subject to reallocation during tuning.

Caution This combo box is invalid if the following condition is satisfied.

- 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 reallocating the function subject to reallocation during tuning.

Caution This specification area is invalid if the following condition is satisfied.

- If a menu other than "Set Address" is selected in the [Insert] combo box

(2) Profile routine specification area

This area is used to make settings related to the memory area for reallocating the profile routine output as an intermediate file during performance analysis or tuning.
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 (in specified area, before specified area, after specified area, or at specified address) of the memory area for allocating the profile routine output as an intermediate file during performance analysis or tuning.

Caution The "The Region" menu of this combo box can be selected only when the user's development environment is the NEC tool chain.
Menus other than the "The Region" menu of this combo box are invalid if either of the following conditions are satisfied.

- If the TWROUTINE segment is defined for the target link directive file
- If the [Use probe code for analyzing(for original execution module only).] radio button on the Execution Item Selection Panel is not checked

(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.

Caution This combo box is invalid if the following condition is satisfied.

- If the TWROUTINE segment is defined for the target link directive file

(c) [Adddress] 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.

Caution This specification area is invalid if either of the following conditions is satisfied.

- If a menu other than "Set Address" is selected in the [Insert] combo box
- If the [Use probe code for analyzing(for original execution module only).] radio button on the Execution Item Selection Panel is not checked

(3) 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.
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 (in specified area, before specified area, after specified area, or at specified address) of the memory area for allocating the profile table output as an intermediate file during performance analysis or tuning.

Caution The "The Region" menu of this combo box can be selected only when the user's development environment is the NEC tool chain.
Menus other than the "The Region" menu of this combo box are invalid if the following condition is satisfied.

- 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.

Caution This combo box is invalid if the following condition is satisfied.

- If the TWTABLE segment is defined for the target link directive file

(c) [Adddress]specification area

This area 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.

Caution This specification area is invalid if the following condition is satisfied.

- If a menu other than "Set Address" is selected in the [Insert] combo box

(d) Size display area

This area displays the size (in bytes) of the profile table that can be acquired by clicking the <Calculate> button.

(e) <Calculate> button

This button calculates the size (in bytes) of the profile table and displays the result of calculation in the size display area.

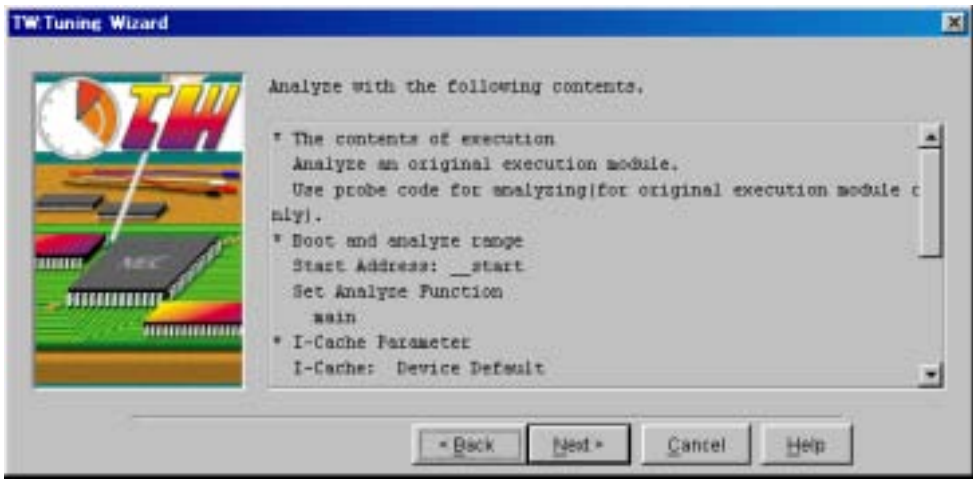
To continue wizard processing (performance analysis or tuning), specify each item and then click the <Next>> button.

5.9 Executing Performance Analysis/Tuning

5.9.1 Check before execution

The following Pre-execution Confirmation Panel is opened when the performance analysis or tuning items necessary for TW to execute wizard processing (performance analysis or tuning) (i.e., setting operations up to **5.5.3 Selecting allocation destination**, **5.7.1 Selecting allocation destination** or **5.8.5 Selecting allocation destination**) have been correctly set.

Figure 5-21 Pre-execution Confirmation Panel



To execute wizard processing (performance analysis or tuning), check the "message reporting completion of setting the performance analysis or tuning items necessary for TW to execute wizard processing (performance analysis or tuning)" displayed in the message display area, and then click the <Next>> button.

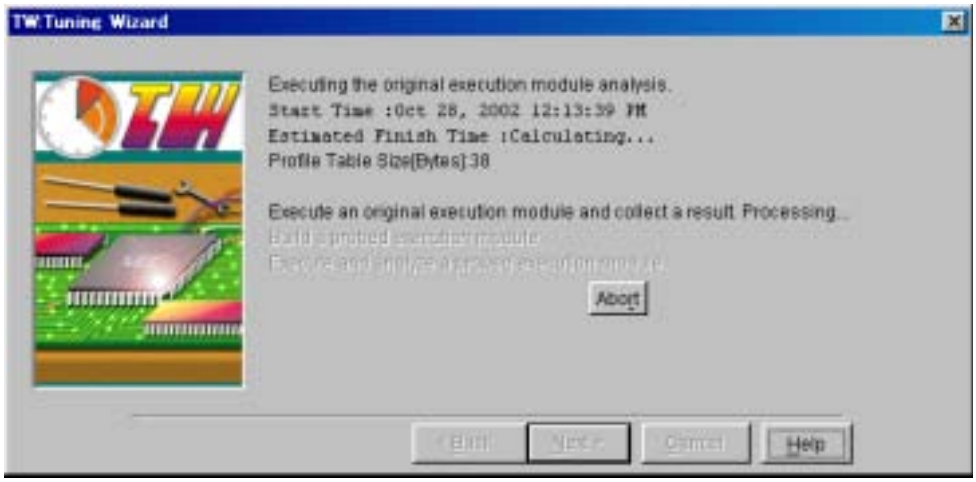
5.9.2 Executing performance analysis/tuning

Wizard processing (performance analysis or tuning) is executed in accordance with the performance analysis or tuning items displayed in **5.9.1 Check before execution**.

However, the user does not have to aware of this phase because TW automatically executes it when the message displayed in **5.9.1 Check before execution** is checked and the <Next>> button is clicked.

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

Figure 5-22 Performance Analysis/Tuning Execution Panel

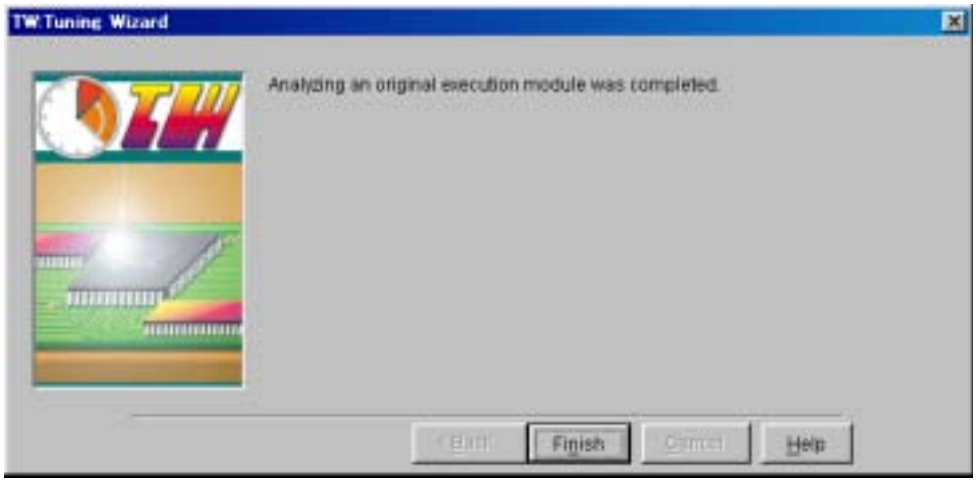


5.9.3 Checking completion of execution

Check that wizard processing (performance analysis or tuning) by TW has been completed.

The following Execution Completion Confirmation Panel is automatically displayed when TW has correctly completed wizard processing (performance analysis or tuning) in **5.9.2 Executing performance analysis/tuning**.

Figure 5-23 Execution Completion Confirmation Panel



To complete wizard processing (performance analysis or tuning), check the "message reporting that TW has completed wizard processing (performance analysis or tuning)" displayed in the message display area, and then click the <Finish> button.

By clicking the <Finish> button, the TW wizard that was started in **5.3.1 Starting wizard** is completed.

Caution If the [Tune and analyze a tuned execution module.] check box is checked in **5.4 Selecting Execution Items**, the tuned execution module is output to the "(original project file name).tw" folder.

5.10 Checking Results

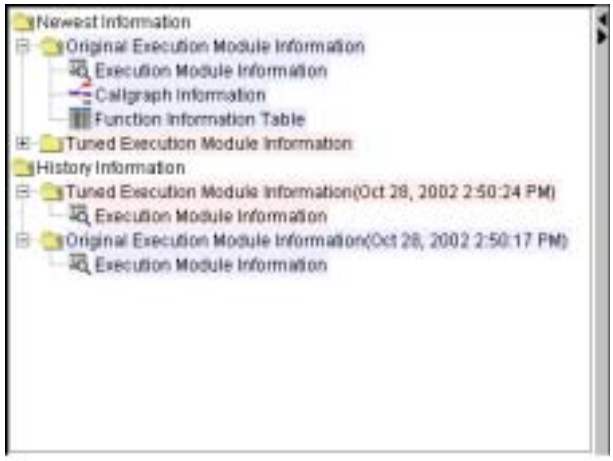
The results of analyzing the performance of the user program or execution module, and the results of tuning can be checked on the Main Window.

5.10.1 History information

The results of analyzing the performance of the original execution module and tuned execution module, or the results of tuning can be checked in the history display area of the Main Window.

The history display area shown in the figure below can be displayed by selecting [View] menu -> [History] on the Main Window.

Figure 5-24 History Display Area



This area consists of the following items.

(1) [Newest Information] item

This displays the latest results of wizard processing (performance analysis or tuning) executed by TW.

(2) [History Information] item

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

(3) [Original Execution Module Information] item

This is a sub-item of [Newest Information] and [History Information].
This item displays the results of wizard processing (performance analysis) TW has executed on the execution module.

(4) [Tuned Execution Module Information] item

This is a sub-item of [Newest Information] and [History Information].
This item displays the results of wizard processing (performance analysis) TW has executed on the tuned execution module.

(5) [Execution Module Information] item

This is a sub-item of [Original Execution Module Information] and [Tuned Execution Module Information]. By clicking the right mouse button or selecting the [View] menu -> [Open] menu on the Main Window after selecting this item, a window (Execution Module Information Window) that displays execution module information (execution result and setting) on the applicable execution module can be opened.

(6) [Callgraph Information] item

This is a sub-item of [Original Execution Module Information] and [Tuned Execution Module Information]. By clicking the right mouse button or selecting the [View] menu -> [Open] menu on the Main Window after selecting this item, a window (Callgraph Window) that displays the performance analysis information (such as callpair information and call count information) on the applicable execution module in graph form can be opened.

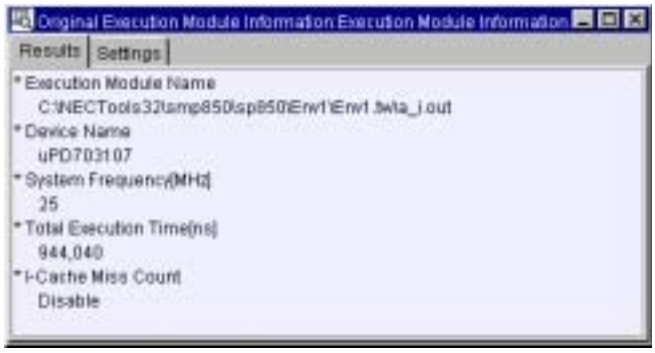
(7) [Function Information Table] item

This is a sub-item of [Original Execution Module Information] and [Tuned Execution Module Information]. By clicking the right mouse button or selecting the [View] menu -> [Open] menu on the Main Window after selecting this item, a window (Function Information Table Window) that displays detailed function information on the applicable execution module can be opened.

5.10.2 Execution module information

Execution module information (execution result and setting) on the execution module is displayed.
The Execution Module Information Window shown in the figure below is opened when [Execution Module Information] is selected in the history display area of the Main Window and then the right mouse button is clicked.

Figure 5-25 Execution Module Information Window



On this window, information on the following items can be checked.

(1) [Results] tab

This tab is used to display information on the execution results of performance analysis (execution result information: such as execution module name, occupied text area size, and device name).
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) Device name display area

This area displays the target device name of the execution module.

(c) Operating frequency display area

This area displays the operating frequency (in MHz) of the execution module.

(d) Total execution time display area

This area displays the total execution time (in ns) in the range subject to performance analysis.
If the module in question is a tuned execution module, this area displays in color the results of comparing the module with an execution module for original analysis.

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.

(e) Instruction cache miss information display area

This area displays the number of instruction cache misses and rate of occurrence of 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 misses in the range subject to performance analysis.

Number of instruction cache misses / (Number of instruction cache hits + Number of instruction cache misses)

(2) [Settings] tab

This tab is used to display information related to performance analysis settings (setting contents information: such as execution contents, execution start position, and execution analysis range).
This tab consists of the following display areas.

(a) Execution contents display area

This area displays the execution items in wizard processing (performance analysis or tuning) specified on the Execution Item Selection Panel.

(b) Execution start position and execution analysis range display area

This area displays the execution start position of the execution module, and range subject to performance analysis specified on the Execution Analysis Range Selection Panel.

(c) Instruction cache type display area

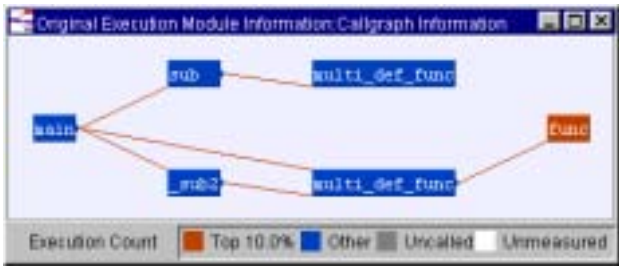
This area displays the type of the instruction cache and instruction cache parameters specified on the Instruction Cache Selection Panel.

5.10.3 Callgraph information

The performance analysis information (such as callpair information and call count information) of the execution module is displayed in graph form.

The Callgraph Window shown in the figure below is opened when [Callgraph Information] is selected in the history display area of the Main Window and then the right mouse button is clicked.

Figure 5-26 Callgraph Window



This window is used to check the following items of information.

(1) Callgraph display area

This area displays information on the execution results of performance analysis (such as callpair information and call count information) in graph form.
This 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 the target function calls a function (calling a child function), a black circle is shown on the right of the function box.

White : Unknown function (function not subject to performance analysis)
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

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

2. "Higher" means the "higher" sort result of the color display format (such as coloring by NET time or GROSS time) specified using the [Callgraph] menu 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.

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

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

(2) Color reference display area

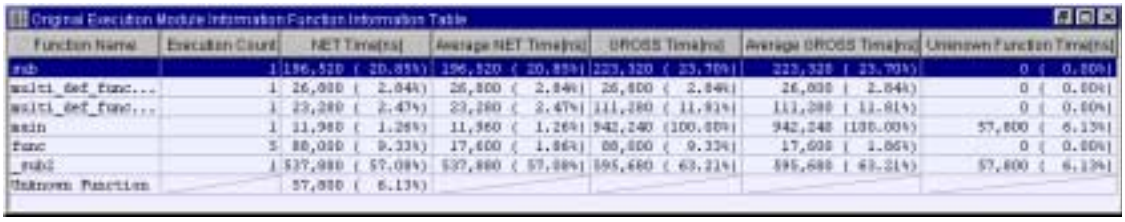
This area displays the color reference (such as coloring by NET time, coloring by GROSS time, or coloring by the number of times of execution) in the function box.
The color reference in the function box can be changed on the menu (such as for coloring by NET time, coloring by GROSS time, and coloring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

5.10.4 Function information table

Function information of the 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 is clicked.

Figure 5-27 Function Information Table Window



This window is used to check the following items of information.

(1) Function information display area

This area displays information on the execution results of performance analysis (function information: such as function name, number of times of execution, NET time) in table form.
This display area consists of the following display areas.

(a) Function name display area

This area displays the name of the target function.

(b) Number of times of execution display area

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

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

This area display the NET time (in ns) of the target function and the proportion of the performance analysis target range to the total NET time (in %).
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.

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

(d) Average NET time display area

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

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.

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

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

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.

(f) Average GROSS time display area

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

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.

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

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

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.

(h) Instruction cache miss display area

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

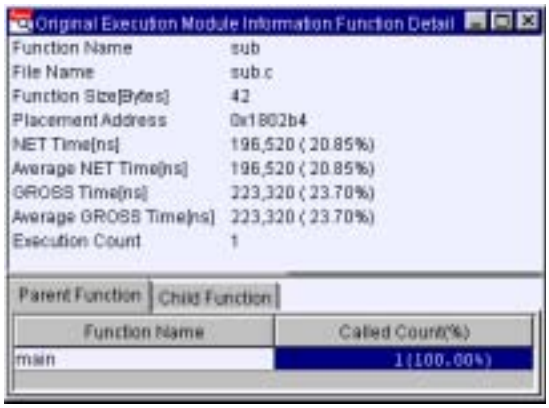
Number of instruction cache misses / (Number of instruction cache hits + Number of instruction cache misses)

5.10.5 Detailed function information

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

The Function Detail Information Window shown in the figure below is opened when the <Function Detail> button on the Main Window is clicked after selecting the 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 the 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 the function name displayed on the [Parent Function] tab or [Child Function] tab in the Function Detail Information Window.

Figure 5-28 Function Detail Information Window



This window is used to check the following items of information.

(1) Performance analysis information display area

This area displays information on the execution results of performance analysis (performance analysis information: such as function name, file name, function size).
This area consists of the following display areas.

(a) Function name display area

This area displays the function 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.

(c) Function size display area

This area displays the size of the text area (in bytes) occupied by the target function.

(d) Function allocation address display area

This area displays the address (first address) to which the text area of the target function is assigned.

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

This area display the NET time (in ns) of the target function and the proportion of the performance analysis target range to the total NET time (in %).
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.

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

(f) Average NET time display area

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

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.

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.

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

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

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.

(h) Average GROSS time display area

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

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.

(i) Number of times of execution display area

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

(j) Instruction cache miss display area

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

Number of instruction cache misses / (Number of instruction cache hits + Number of instruction cache misses)

(2) [Parent Function] tab

This tab displays information on the execution result 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) Parent function name display area

This area displays the parent function of the target function.

(b) Number of times called display area

This area displays the number of times the target function has been called by the parent function, and the proportion (in %) to the total number of times the function has been called.

(3) [Child Function] tab

This tab displays information on the execution result of performance analysis (child function information: such as child function name, number of times function is called) 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.

(b) Number of times calling display area

This area displays the number of times the target function has called by the child function, and the proportion (in %) to the total number of times the child function has been called.

5.11 Exiting TW

For details of how to exit TW, refer to **3 STARTING AND TERMINATING**.
If an exit operation has been executed for TW, the following Verification Dialog Box is automatically opened.

Figure 5-29 Verification Dialog Box

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

CHAPTER 6 WINDOW REFERENCE

This chapter details the functions of the windows, panels and dialog boxes of TW.

6.1 Overview

Table6-1 lists the windows, panels, and dialog boxes of TW.

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

Window name, Panel name, or Dialog Box name	Functional Outline
Main Window	Window that controls execution of wizard processing (performance analysis or tuning) and displays performance analysis information (such as callpair information and call count information)
Callgraph Window	Window that lists and displays in graph form the performance analysis information of the execution module (such as callpair information and call count information)
Function Information Table Window	Window that lists and displays in table form the function information of the execution module (such as the function name, the number of times of execution, and NET time)
Execution Module Information Window	Window that displays the execution module information of the execution module (such as execution results and settings)
Function Detail Information Window	Window that displays the detailed function information of the execution module (such as performance analysis information, parent function information, and child function information)
Tuning wizard Initial Screen Panel	Panel that reports that TW has started wizard processing (performance analysis or tuning)
Tuning wizard Execution Module Information Acquisition Panel	Panel that reports that TW is executing wizard processing (performance analysis: processing to generate an execution module for original analysis, and to analyze its performance)
Tuning wizard Execution Item Selection Panel	Panel that is used to specify execution items for wizard processing (performance analysis or tuning)
Tuning wizard Execution Analysis Range Selection Panel	Panel that is used to specify the execution start position of the user program or execution module, and the performance analysis range of wizard processing (performance analysis)
Tuning wizard Instruction Cache Selection Panel	Panel that is used to specify the type and parameters of the instruction cache used for wizard processing (performance analysis or tuning)
Tuning wizard Fast Access Memory Area Selection Panel	Panel that is used to specify the type and parameters of the fast access memory area used for wizard processing (tuning)
Tuning wizard Tuning Method Selection Panel	Panel that is used to specify the method of tuning the user program or execution module (standard pattern or user-specified pattern)
Tuning wizard Fast Access Memory Area Tuning Selection Panel	Panel that is used to specify the method of allocating functions to the fast access memory area (such as the internal ROM) based on the callpair information that was acquired when the performance of the user program or execution module was analyzed

Window name, Panel name, or Dialog Box name	Functional Outline
Tuning wizard Instruction Cache Tuning Selection Panel	Panel that is used to specify the method of tuning the instruction cache (tuning algorithm and allocation priority correction option) based on the callpair information that was acquired when the performance of the user program or execution module was analyzed
Tuning wizard Callpair Correction Option Selection Panel	Panel that is used to specify the method of tuning the callpair information (method of correcting callpair information) that was acquired when the performance of the user program or execution module was analyzed
Tuning wizard Allocation Destination Selection Panel	Panel that is used to specify the memory area where the specified function is to be reallocated during tuning, and the memory area where the profile routine and profile table output as intermediate files during tuning are to be allocated
Tuning wizard Priority Allocation Function Selection Panel	Panel that is used to specify the function that is given priority and reallocated during wizard processing (tuning)
Tuning wizard Pre-execution Confirmation Panel	Panel that reports that the items of performance analysis and tuning required by TW to execute wizard processing (performance analysis or tuning) have been set
Tuning wizard Performance Analysis/Tuning Execution Panel	Panel that reports that TW is executing wizard processing (performance analysis or tuning)
Tuning wizard Execution Completion Confirmation Panel	Panel that reports that TW has correctly completed wizard processing (performance analysis or tuning)
Option Dialog Box	Dialog box used to specify the method of printing the Callgraph Window displayed in the internal window display area of the Main Window, and whether the intermediate files (temporary files) output while TW was executing performance analysis or tuning are to be deleted
Version Information Dialog Box	Dialog box that displays the version information of TW (such as version and copyright)
File Selection Dialog Box	Dialog box that is used to specify the file (such as build file or timer control code block) required by TW to execute performance analysis or tuning, or the file (such as a project information saving file) to save the execution status of TW
Message Dialog Box	ialog box that displays the information to be reported to the user (such as progress of processing and occurrence of warning) detected by TW while it was executing performance analysis or tuning
Error Dialog Box	Dialog box that displays the information to be reported to the user (such as occurrence of a fatal error) detected by TW while it was executing performance analysis or tuning
Verification Dialog Box	Dialog box that displays the inquiry information (such as problem in continuing processing) detected by TW while it was executing performance analysis or tuning
Input Dialog Box	Dialog box that is used to specify the information requested by TW (such as color boundary, display magnification rate, and root function)
Start Option Setting Dialog Box	Dialog box used to specify the start options of TW

6.2 Input Rules

This section explains the rules for inputting characters (alphanumeric characters) in the window, panel, or dialog box of TW.

- File name

  TW generates an intermediate file 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 original file name the user can specify is a name that is five characters fewer than the maximum number of characters defined by the C compiler package.

- Path name

  TW generates an intermediate file in the folder that stores the original file, or a folder specified by TW while TW is executing performance analysis or tuning of a user program or an execution module.

  Therefore, the maximum number of characters in the user-specified path name is the number defined by the C compiler package.

- Function name and symbol name

  TW outputs performance analysis information with the original function name or original symbol name plus up to 12 characters when it is executing performance analysis or tuning of a user program or an execution module.

  Therefore, the original function name or original symbol name the user can specify is a name that is 12 characters fewer than the maximum number of characters defined by the C compiler package.

- Numeric value

  TW treats a word starting with a number (0 to 9) as a numeric value.

  TW distinguishes numeric values as follows.

  XXXX : Decimal
  0x*XXXX* : Hexadecimal

- Keywords

  TW treats a word starting with "tw_" or "..", and "__e_sysfnc" as a system-reserved macro.

  Therefore, using these words for a purpose other than that specified is prohibited.

6.3 Window Explanation

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

Window name, Panel name, Dialog Box name

The functional outline of the window, panel, or dialog box is described, followed by a display image.

How to open this window

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

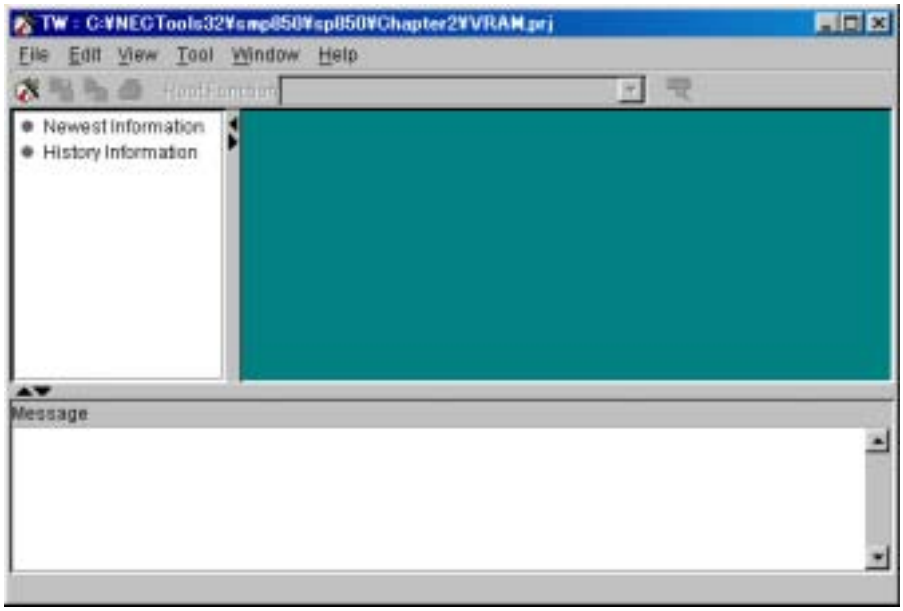
Area of this window

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

Main Window

This window controls execution of 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 explains the following items.

- How to open this window
- Areas of this window
- Pop-up menu

How to open this window

This window opens automatically when TW is started.

Areas of this window

This window consists of the following items.

- Title bar
- Menu bar
- Toolbar
- History display area
- Internal window display area
- Message display area
- Status bar

(1) Title bar

This title bar has the following buttons.

(a) <Minimize> button

Minimizes this window (to an icon), and displays it as a button on the task bar.

(b) <Maximize> button

Maximizes and displays this window fully on the monitor screen.

(c) <Restore> button

Reduces and displays the window in size before maximization.

(d) <Close> button

Closes this window.

(2) Menu bar

This menu bar consists of the following menus.

(a) [File] menu

This menu consists of the following menus.

- [Open MULTI Project...] menu (when using GHS tool chain)

  Opens a dialog box (File Selection Dialog Box) used to specify the file (build file) required by TW to execute performance analysis or tuning.

- [Recent MULTI Projects] menu (when using GHS tool chain)

  Opens a menu (of five types of build files read in the past) to specify the file (build file) required by TW to execute performance analysis or tuning.

- [Save TW Project] menu

  Opens a dialog box (File Selection Dialog Box) used to specify the file (such as a project information saving file) used to save the execution status of TW.

- [Export...] menu

  By selecting this menu after selecting an internal window (such as the Function Detail Information Window, Function Information Table Window, or Execution Module Information Window), a dialog box (File Selection Dialog Box) used to specify the file to save the selected window in text format is opened.

- [Print...] menu

  By selecting this menu after selecting an internal window (such as the Callgraph Window or Function Information Table Window), the selected window is printed.

- [Exit...] menu

  Closes this window.

(b) [Edit] menu

This menu consists of the following menus.

- [Cut] menu

  Cuts and pastes the text selected in the text input area to the copy buffer.

  Caution This menu is invalid if the following condition is satisfied.

  - If TW is executing wizard processing

- [Copy] menu

  Copies the text selected in the text input area to the copy buffer.

  Caution This menu is invalid if the following condition is satisfied.

  - If TW is executing wizard processing

- [Paste] menu

  Pastes the text stored in the copy buffer using the [Cut] or [Copy] menu (text cut or copied) to the specified text input area.

  Caution This menu is invalid if the following condition is satisfied.

  - If TW is executing wizard processing

- [Delete] menu

  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.

  Caution This menu is invalid if the following condition is satisfied.

  - If TW is executing wizard processing

- [Add To History] menu

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

- [Restore From History...] menu

  By selecting this menu after selecting an item displayed in the history display area (such as [Original Execution Module Information] or [Tuned Execution Module Information]), the setting of the selected item (such as performance analysis information and tuning information) is reflected on TW.

(c) [View] menu

This menu consists of the following menus.

- [Toolbar] menu

  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 menu is checked, the toolbar is displayed.

- [History] menu

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

- [Message] menu

  Specifies whether the message display area (that displays the information to be reported to the user detected by TW) is to be displayed.
  If this menu is checked, the message display area is displayed.

- [Open] menu

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

- [Function Detail] menu

  By selecting this menu after selecting a function name displayed in the Callgraph Window, Function Information Table 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.

- [Callgraph] menu

  This menu consists of the following menus.

  - [NET Time] menu

    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 menu is checked, the NET time is used as a reference for coloring the function box.

  - [GROSS Time] menu

    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 menu is checked, the GROSS time is used as a reference for coloring the function box.

- [Execute Count] menu

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

- [I-Cache Miss] menu

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

- [Coloring Threshoud...] menu

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

- [Expaned], [Shrink] menu

  By selecting these menus after selecting a function box displayed in the callgraph display area of the Callgraph Window, the selected function box is expanded or folded.

- [Zoom...] menu

  By selecting this menu after selecting the Callgraph Window, a window (Input Dialog Box) used to change the display magnification rate of the callgraph display area is opened.

- [Change Root Funsion...] menu

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

- [Hide Excecution Time 0 Functions]menu

  Specify whether a function with execution time of 0 or an execution count of 0 is displayed or not when an internal window (such as the Callgraph Window, Function Information Table Window, and Execution Module Information Window) is displayed.

(d) [Tool] menu

This menu consists of the following menus.

- [Tuning Wizard...] menu

  Starts TW wizard processing (performance analysis or tuning).
  When this menu is selected, a panel (Initial Screen Panel) reporting that TW has started wizard processing (performance analysis or tuning) is opened.

- [Option...] menu

  Starts option setting processing.
  When this menu is selected, a dialog box (Option Dialog Box) that is used to specify the printing method of the Callgraph Window, whether intermediate files (temporary files) output when TW has executed performance analysis or tuning are to be deleted, and whether TW system library libtw*.a is used to execute performance analysis or tuning.

(e) [Window] menu

This menu consists of the following menus.

- [Cascade] menu

  Overlaps and displays the internal window display areas (cascade display).

- [Tile] menu

  Arranges and displays the internal window display areas (tile display).

(f) [Help] menu

This menu consists of the following menus.

- [TW Help] menu

  Starts the HTML help related to TW.

- [Main Window] menu

  Starts the HTML help related to this window.

- [Current Window F1] menu

  Starts the HTML help related to the window, panel, or dialog box in the active status.

- [About...] menu

  Opens a dialog box (Version Information Dialog Box) used to display the version information of TW (such as version and copyright).

(3) Toolbar

This toolbar consists of the following buttons and combo box.

(a) <Tuning Wizard> button

Starts TW wizard processing (performance analysis or tuning).
When this button is clicked, a panel (Initial Screen Panel) reporting that TW has started wizard processing (performance analysis or tuning) is opened.

(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 can be added and registered to [History Information].

(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 [Newest Information] in the history display area, the setting of the selected item (such as performance analysis or tuning) is reflected on TW.

(d) <Print> button

By selecting this menu after selecting an internal window (such as the Callgraph Window or Function Information Table Window), the selected window is printed.

(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.

(f) <Function Detail> button

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

(4) History display area

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

(a) [Newest Information] item

Displays the latest result of wizard processing executed by TW (performance analysis or tuning).

(b) [History Information] item

Displays the results (up to 20 results in the past) of wizard processing executed by TW (performance analysis or tuning).

(c) [Original Execution Module Information] item

This is a sub-item of the [Newest Information] and [History Information] items.
It displays the result of wizard processing (performance analysis) TW has executed on the original execution module.

(d) [Tuned Execution Module Information] item

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

(e) [Execution Module Information] item

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

(f) [Callgraph Information] item

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

(g) [Function Information Table] item

By clicking the right mouse button on this window or selecting [View] menu -> [Open] after selecting this item, a window (Function Information Table Window) used to display the detailed function information of the selected execution module is opened.

(5) Internal window display area

This area displays an internal window (such as the Callgraph Window, Function Information Table Window, Execution Module Information Window, and Function Detail Information Window).
The display format of this display area can be selected from the menu (overlapped or arranged) that is displayed by selecting the [Window] menu.

(6) Message display area

This area displays the information to be reported to the user detected by TW.
Whether this area is to be displayed or not can be selected by [View] menu -> [Message].

(7) Status bar

This area displays the progress of the processing by TW.
If TW is executing processing that takes time, a progress bar is displayed.

Pop-up menu

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

The pop-up menu consists of the following menus.

(1) [Open] menu

Opens the window (Callgraph Window, Function Information Table Window, or Execution Module Information Window) corresponding to the selected item.

(2) [Add To History] menu

Adds and registers the selected item to [History Information].

(3) [Restore From History...] menu

Reflects the setting of the selected item (such as performance analysis information and tuning information) on TW.

(4) [Delete] menu

Deletes the selected item.

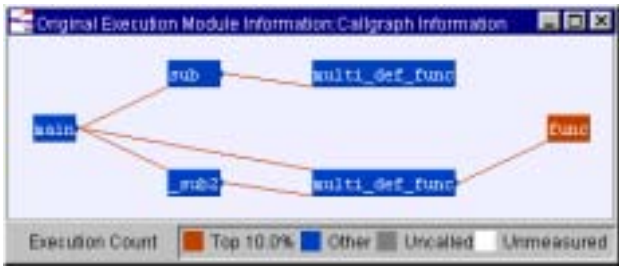
(5) [Rename] menu

Opens a dialog box (Input Dialog Box) used to change the name of the selected item.

Callgraph Window

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

Figure 6-2 Callgraph Window



This section explains the following items.

- How to open this window
- Areas of this window
- Pop-up menu

How to open this window

This panel can be opened as follows.

- On the Main Window,
  - Select (double-click) [Callgraph Information] in the history display area.
- After selecting [Callgraph Information] in the history display area of the Main Window,
  - Click the right mouse button and select [Open] menu.
  - Select [View] menu -> [Open].

Areas of this window

This window consists of the following items.

- Title bar
- Callgraph display area
- Coloring reference display area

(1) Title bar

This title bar has the following buttons.

(a) <Minimize> button

Minimizes this window (to an icon), and displays it as a button on the task bar.

(b) <Maximize> button

Maximizes and displays this window fully on the monitor screen.

(c) <Restore> button

Reduces and displays the window in the size before maximization.

(d) <Close> button

Closes this window.

(2) Callgraph display area

This area displays information on the results of performance analysis execution (such as callpair information and call count information) in graph form.
This 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 the target function calls a function (calls a child function), a black circle is shown on the right of the function box.

White : Unknown function (function not subject to performance analysis)
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

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

2. If the function box is clicked, the function box is selected, and the calling relational line connecting the parent function box of the target function, child function box, 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, TW executes a selection operation for each target function.

3. If the function box is double-clicked, the Function Detail Information Window for displaying 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 the function box and calling relational line is clicked, all the function boxes are unselected.

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] menu 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.

Cautions 1. Color boundary value xxx% can be changed in the Input Dialog Box that is opened by selecting [View] menu -> [Callgraph] menu -> [Coloring Threshoud...] menu 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.

3. If the calling relational line is clicked, all the function boxes are unselected.

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

(3) Coloring reference display area

This area displays the color reference (such as coloring by NET time, coloring by GROSS time, or coloring by the number of times of execution) 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, and coloring by the number of times of execution) that is displayed by selecting [View] menu -> [Callgraph] menu on the Main Window.

Pop-up menu

A pop-up menu is displayed if the right mouse button is clicked in the following areas.

(1) Function box

The pop-up menu displayed when the right mouse button is clicked in the function box consists of the following menus.

(a) [Set To Root Function] menu

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

(b) [Open Function Detail] menu

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] menu

Expands or folds the function box at the selected position.

(2) Areas other than function box

The pop-up menu displayed when the right mouse button is clicked in an area other than the function box consists of the following menus.

(a) [NET Time] menu

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

(b) [GROSS Time] menu

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

(c) [Execution Count] menu

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

(d) [I-Cache Miss] menu

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

(e) [Coloring Threshold...] menu

Displays a dialog box (Input Dialog Box) to specify the coloring boundary value of the function box.

(f) [Zoom...] menu

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

(g) [Change Root Function...] menu

Displays a dialog box (Input Dialog Box) to change the root function displayed in the callgraph display area.

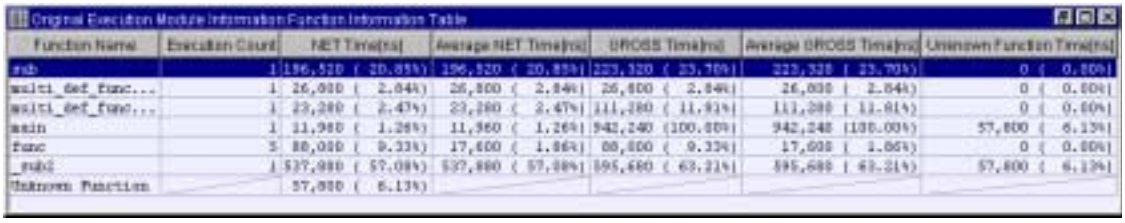
(h) [Hide Execution Time 0 Functions] menu

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

Function Information Table Window

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

Figure 6-3 Function Information Table Window



This section explains the following items.

- How to open this window
- Areas of this window
- Pop-up menu

How to open this window

This window can be opened as follows.

- On the Main Window,
  - Select (double-click) [Callgraph Information] in the history display area.
- Select [Function Information Table] in the history display area of the Main Window,
  - Click the right mouse button and select [Open] menu.
  - Select [View] menu -> [Open].

Areas of this window

This window consists of the following items.

- Title bar
- Function information display area

(1) Title bar

This title bar has the following buttons.

(a) <Minimize> button

Minimizes this window (to an icon), and displays it as a button on the task bar.

(b) <Maximize> button

Maximizes and displays this window fully on the monitor screen.

(c) <Restore> button

Reduces and displays the window in the size before maximization.

(d) <Close> button

Closes this window.

(2) Function information display area

This area displays information on the results of performance analysis execution (function information: such as function name, number of times of execution, NET time) in table form.
This display area consists of the following display areas.

(a) Function name display area

This area displays the name of the target function.

(b) Number of times of execution display area

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

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

This area display the NET time (in ns) of the target function and the proportion of the performance analysis target range to the total NET time (in %).
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.

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

(d) Average NET time display area

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

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.

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

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

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.

(f) Average GROSS time display area

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

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.

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

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

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.

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.

(h) Instruction cache miss display area

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

Number of instruction cache misses / (Number of instruction cache hits + Number of instruction cache misses)

Cautions 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 another Callgraph Window or Function Information Table Window is displaying the target function, TW executes a selection operation for each target function.
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 that displays the detailed information of the target function is opened.
By dragging the function box, the target function box can be moved.

Pop-up menu

A pop-up menu is displayed if a function is selected in the function information table area and the right mouse button is clicked.
The pop-up menu consists of the following menu.

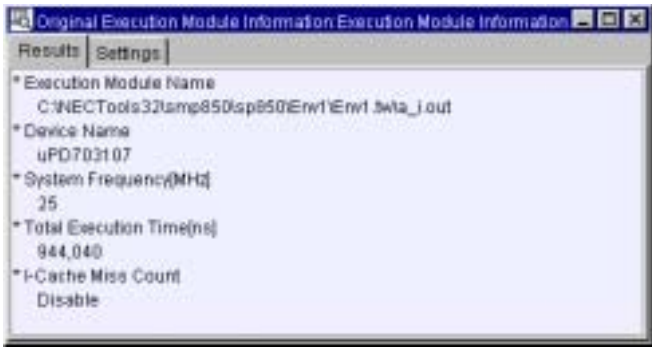
(1) [Open Function Detail] menu

Opens a window (Function Detail Information Window) that shows the detailed information of the target function (performance analysis information, parent function information, and child function information).

Execution Module Information Window

This window displays the execution module information of the execution module (such as the execution results and setting contents).

Figure 6-4 Execution Module Information Window



This section explains the following items.

- How to open this window
- Areas of this window

How to open this window

This window can be opened as follows.

- On the Main Window,
  - Select (double-click) [Callgraph Information] in the history display area.
- After selecting [Execution Module Information] in the history display area of the Main Window,
  - Click the right mouse button and select [Open] menu.
  - Select [View] menu -> [Open].

Areas of this window

This window consists of the following items.

- Title bar
- [Result] tab
- [Settings] tab

(1) Title bar

This title bar has the following buttons.

(a) <Minimize> button

Minimizes this window (to an icon), and displays it as a button on the task bar.

(b) <Maximize> button

Maximizes and displays this window fully on the monitor screen.

(c) <Restore> button

Reduces and displays the window in the size before maximization.

(d) <Close> button

Closes this window.

(2) [Result] tab

This tab is used to display information on the results of performance analysis execution (execution result information: such as execution module name, occupied text area size, device name).
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) Device name display area

This area displays the target device name of the execution module.

(c) Operating frequency display area

This area displays the operating frequency (in MHz) of the execution module.

(d) Total execution time display area

This area displays the total execution time (in ns) in the range subject to performance analysis.
If the applicable module is a tuned execution module, this area displays in color the results of comparing the module with the execution module for original analysis.

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.

(e) Instruction cache miss information display area

This area displays the number of instruction cache misses and rate of occurrence of 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 misses in the range subject to performance analysis.

Number of instruction cache misses / (Number of instruction cache hits + Number of instruction cache misses)

(3) [Settings] tab

This tab is used to display information related to performance analysis settings (setting contents information: such as execution contents, execution start position, execution analysis range).
This tab consists of the following display areas.

(a) Execution contents display area

This area displays the execution items for wizard processing (performance analysis or tuning) specified on the Execution Item Selection Panel.

(b) Execution start position and execution analysis range display area

This area displays the execution start position of the execution module, and range subject to performance analysis specified on the Execution Analysis Range Selection Panel.

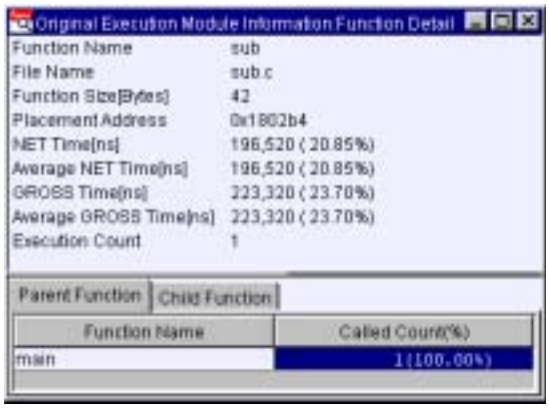
(c) Instruction cache type display area

This area displays the type of instruction cache and instruction cache parameters specified on the Instruction Cache Selection Panel.

Function Detail Information Window

This window displays the detailed function information of the execution module (performance analysis information, parent function information, and child function information).

Figure 6-5 Function Detail Information Window



This section explains the following items.

- How to open this window
- Areas of this window

How to open this window

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 [View] menu -> [Function 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 [View] menu -> [Function 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 name displayed on the [Parent Function] tab or [Child Function] tab of this window,
  - Click the <Function Detail> button.
  - Select [View] menu -> [Function Detail].
- On this window,
  - Select (double-click) a function name displayed on the [Parent Function] tab or [Child Function] tab.

Areas of this window

This window consists of the following items.

- Title bar
- Performance analysis information display area
- [Parent Function] tab
- [Child Function] tab

(1) Title bar

This title bar has the following buttons.

(a) <Minimize> button

Minimizes this window (to an icon), and displays it as a button on the task bar.

(b) <Maximize> button

Maximizes and displays this window fully on the monitor screen.

(c) <Restore> button

Reduces and displays the window in the size before maximization.

(d) <Close> button

Closes this window.

(2) Performance analysis information display area

This area displays information on the results of performance analysis execution (performance analysis information: such as function name, file name, function size).
This area consists of the following display areas.

(a) Function name display area

This area displays the function 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.

(c) Function size display area

This area displays the size of the text area (in bytes) 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 display the NET time (in ns) of the target function and the proportion of the performance analysis target range to the total NET time (in %).
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.

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

(f) Average NET time display area

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

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.

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

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

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.

(h) Average GROSS time display area

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

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.

(i) Number of times of execution display area

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

(j) Instruction cache miss display area

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

Number of instruction cache misses / (Number of instruction cache hits + Number of instruction cache misses)

(3) [Parent Function] tab

This tab displays information on the results of performance analysis execution (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) Parent function name display area

This area displays the parent function of the target function.

(b) Number of times called display area

This area displays the number of times the target function has been called by the parent function, and the proportion (in %) to the total number of times the function has been called.

Cautions 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 that displays the detailed information of the target function is opened.

(4) [Child Function] tab

This tab displays information on the results of performance analysis execution (child function information: such as child function name, number of times function is called) 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.

(b) Number of times of calling display area

This area displays the number of times the target function has called by the child function, and the proportion (in %) to the total number of times the child function has been called.

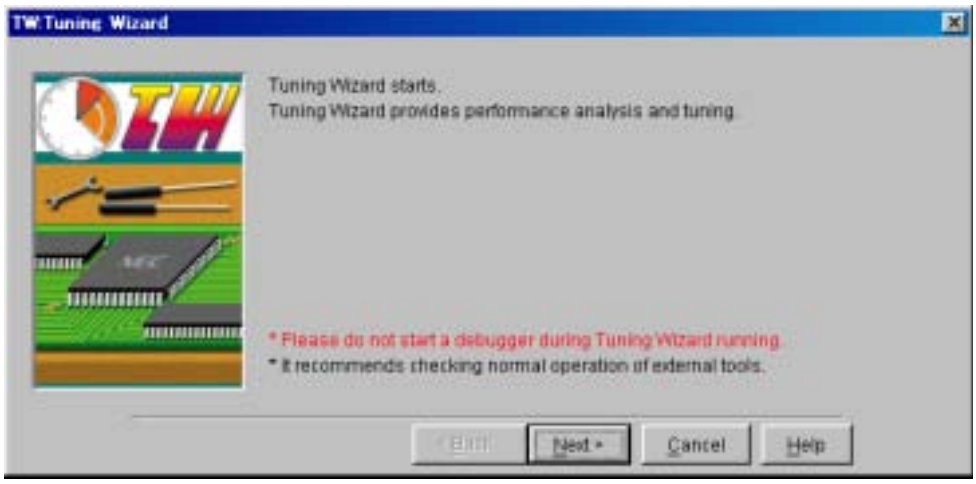
Cautions 1. By clicking the header of each item, the function information can be sorted and displayed on the basis of the specified item.

2. If the information of each function is double-clicked, the Function Detail Information Window that displays the detailed information of the target function is opened.

Initial Screen Panel

This panel reports that TW has started wizard processing (performance analysis or tuning).

Figure 6-6 Initial Screen Panel



This section explains the following items.

- How to open this panel
- Areas of this panel

How to open this panel

This panel consists of the following items.

- On the Main Window,
  - Click the <Tuning Wizard> button.
  - Select [Tool] menu -> [Tuning Wizard...].
- On the Execution Module Information Acquisition Panel,
  - Click the <<Back> button.
- On the Execution Item Selection Panel,
  - Click the <<Back> button.

Areas of this panel

This panel consists of the following items.

- Title bar
- Message display area
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Closes this panel without executing wizard processing (performance analysis or tuning).

(2) Message display area

This area displays a message reporting that TW has started wizard processing (performance analysis or tuning).

(3) Function buttons

The following function buttons are available.

(a) <<Back> button

This button cannot be clicked because it is invalid.

(b) <Next>> button

Closes this panel and then opens the Execution Module Information Acquisition Panel.

(c) <Cancel> button

Closes this panel without executing wizard processing (performance analysis or tuning).

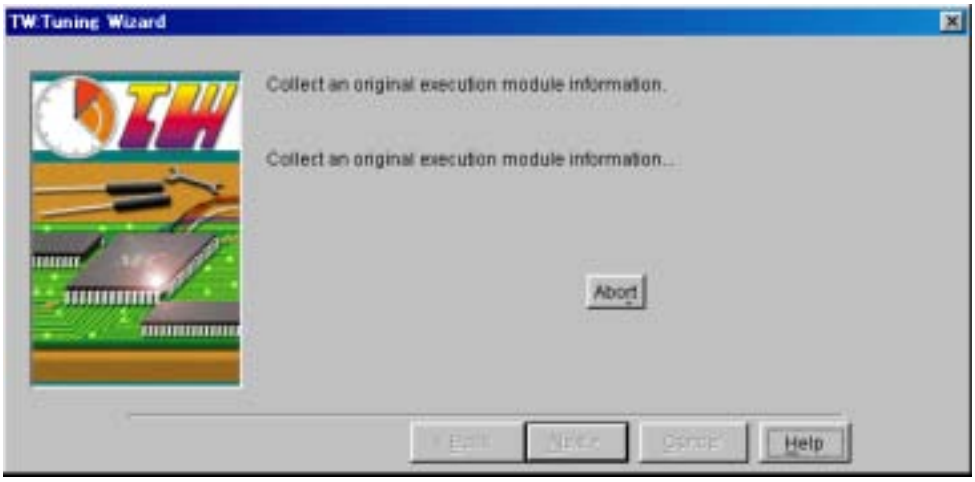
(d) <Help> button

Starts the HTML help.

Execution Module Information Acquisition Panel

This panel reports that TW is executing wizard processing (performance analysis processing: processing to generate an execution module for original analysis and to analyze its performance).

Figure 6-7 Execution Module Information Acquisition Panel



This section explains the following items.

- How to open this panel
- Areas of this panel

How to open this panel

This panel can be opened as follows.

- On the Initial Screen Panel,
  - Click the <Next>> button.

Areas of this panel

This panel consists of the following items.

- Title bar
- Message display area
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Closes this panel without executing wizard processing (performance analysis processing: processing to generate an execution module for original analysis and to analyze its performance).

Caution This button is invalid if the following condition is satisfied.

- If TW is acquiring information on the original execution module

(2) Message display area

This area displays a message reporting that TW is executing wizard processing (performance analysis processing: processing to generate an execution module for original analysis and to analyze its performance).

(a) <Abort> button

Stops wizard processing (performance analysis processing: processing to generate an execution module for original analysis and to analyze its performance).

Caution It may take time for wizard processing to actually stop after this button is clicked.

(b) <Retry> button

Re-executes wizard processing (performance analysis processing: processing to generate an execution module for original analysis and to analyze its performance).

(3) Function buttons

The following buttons are available.

(a) <<Back> button

Closes this panel and opens the Initial Screen Panel.

Caution This button is invalid if the following condition is satisfied.

- If TW is acquiring information on the original execution module

(b) <Next>> button

This button cannot be clicked because it is invalid.

(c) <Cancel> button

Closes this panel without executing wizard processing (performance analysis processing: processing to generate an execution module for original analysis and to analyze its performance).

Caution This button is invalid if the following condition is satisfied.

- If TW is acquiring information on the original execution module

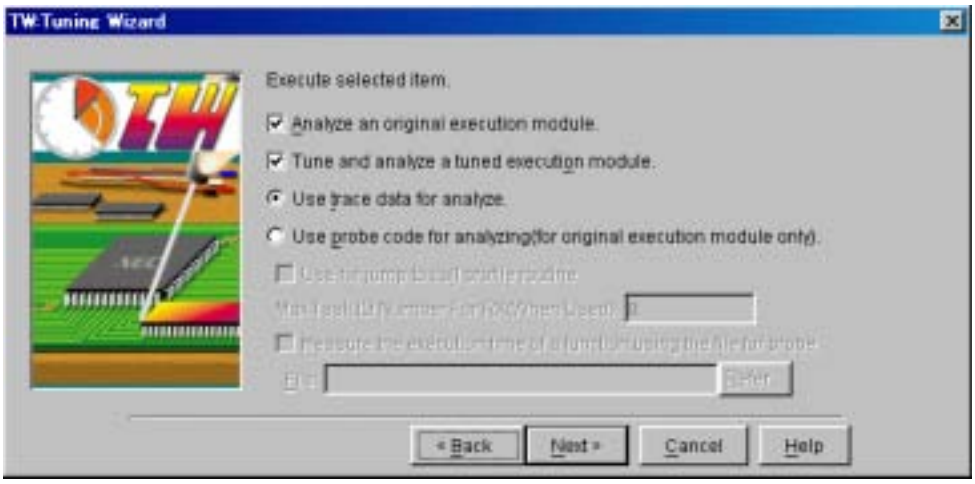
(d) <Help> button

Starts the HTML help.

Execution Item Selection Panel

This panel is used to specify execution items for wizard processing (performance analysis or tuning).

Figure 6-8 Execution Item Selection Panel



This section explains the following items.

- How to open this panel
- Areas of this panel

How to open this panel

This panel is automatically opened when TW has correctly completed wizard processing (performance analysis processing: processing to generate an execution module for original analysis and analyze its performance).
This panel can also be opened as follows.

- On the Execution Analysis Range Selection Panel,
  - Click the <<Back> button.

Areas of this panel

This panel consists of the following items.

- Title bar
- Message display area
- Execution item specification area
- Performance analysis information acquisition method specification area
- Probe code insertion method option specification area
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Closes this panel without executing wizard processing (performance analysis or tuning).

(2) Message display area

This area displays messages explaining this panel.

(3) Execution item specification area

This area is used to make settings related to the execution items of wizard processing (performance analysis and 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 the original execution module as TW wizard processing.
When this check box is checked, performance analysis is executed on the original execution module.

(b) [Tune and analyze a tuned execution module.] check box

This check box is used to specify whether tuning processing (generation of tuned execution module) is performed on the original execution module and whether performance analysis processing is performed on the tuned execution module, as TW wizard processing.
When this check box is checked, tuning processing (generation of tuned execution module) is performed on the original execution module, and performance analysis processing is performed on the tuned execution module.

(4) Performance analysis information acquisition method specification area

This area is used to set a method of acquiring performance analysis information in the wizard processing (performance analysis and tuning).
This specification area consists of the following radio buttons.

(a) [Use trace data for analyze.] radio button

This button is used to specify whether the trace data analysis method is used to acquire performance analysis information in the wizard processing.
If this radio button is checked, the trace data analysis method is used as the method to acquire performance analysis information.

(b) [Use probe code for analyzing(for original execution module only).] radio button

This button is used to specify whether the probe code insertion method is used to acquire performance analysis information in the wizard processing.
If this radio button is checked, the probe code insertion method is used as the method to acquire performance analysis information.

(5) Probe code insertion method option specification area

This area is used to set the probe code insertion method options required when the [Use probe code for analyzing(for original execution module only).] radio button is checked.
This specification area consists of the following check box, specification area, and buttons.

(a) [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 the profile routine that is output as an intermediate file as a result of TW wizard processing (tuning).
If this check button is checked, the far jump instruction is used to call the profile routine.

(b) [Max Task ID Number For RX(When Used)] specification area

This area is used to specify the maximum task ID number for profile that is required for TW to execute wizard processing (performance analysis processing: RTOS information acquisition).
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.

Caution The value specified in this specification area is meaningful only when all the following conditions are satisfied.

- If the [Use probe code for analyzing(for original execution module only).] radio button is checked on the Execution Item Selection Panel
- If the functions supplied by the RTOS (NEC's RX850 or RX850 Pro) are used in the user program

(c) [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 of a function subject to performance analysis is measured as TW wizard processing (performance analysis processing).
If this check box is checked, the execution time of the function is measured.

(d) Time measurement file name specification area

This area is used to specify the file name of a file for measuring time (timer control code block), required for TW to execute wizard processing (performance analysis processing: execution time measurement of function subject to performance analysis).

Caution 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 checked

(e) <Refer...> button

This button opens the File Selection Dialog Box.

Caution 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

(6) Function buttons

The following buttons are available.

(a) <<Back> button

Closes this panel and then opens the Initial Screen Panel.

(b) <Next>> button

Closes this panel and then opens the Execution Analysis Range Selection Panel.

Caution This button is invalid if the following condition is satisfied.

- If data input to the parameters on this panel is illegal

(c) <Cancel> button

Closes this panel without executing wizard processing (performance analysis or tuning).

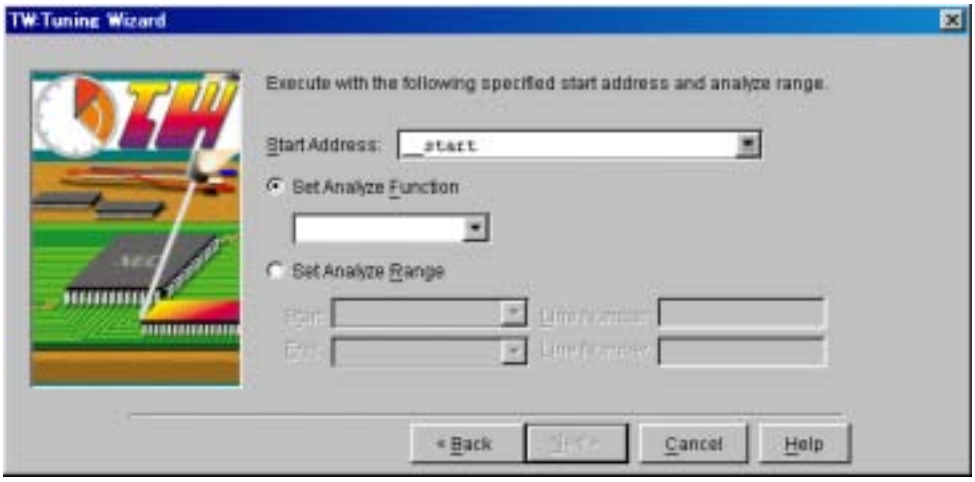
(d) <Help> button

Starts the HTML help.

Execution Analysis Range Selection Panel

This panel is used to specify the position from which execution of the user program or execution module is to be started, and the range of performance analysis by wizard processing (performance analysis processing).

Figure 6-9 Execution Analysis Range Selection Panel



This section explains the following items.

- How to open this panel
- Areas of this panel

How to open this panel

This panel can be opened as follows.

- On the Execution Item Selection Panel,
  - Click the <Next>> button.
- On the Instruction Cache Selection Panel,
  - Click the <<Back> button.

Areas of this panel

This panel consists of the following items.

- Title bar
- Message display area
- [Start Address] combo box
- Performance analysis target function specification area
- Performance analysis target range specification area
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Closes this panel without executing wizard processing (tuning).

(2) Message display area

This area displays messages explaining this panel

(3) [Start Address] combo box

This combo box is used to select the execution start position of the user program and execution module.

Caution 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

(4) Performance analysis target function specification area

This area is used to specify the performance analysis target range for wizard processing (performance analysis). This specification area consists of the following radio button and specification area.

(a) [Set Analyze Function] radio button

This button is used to specify whether the performance analysis target range for wizard processing (performance analysis) is specified by a function name. If this button is checked, the performance analysis target range is specified by a function name.

Caution 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 box is used to select the performance analysis target range (function name) for wizard processing (performance analysis).

Cautions 1. TW 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. 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

(5) Performance analysis target range specification area

This area is used to specify the performance analysis target range for wizard processing (performance analysis). This specification area consists of the following radio button, combo box, and specification area.

(a) [Set Analyze Range] radio button

This button is used to specify whether the performance analysis target range for wizard processing (performance analysis) is specified by a file name, symbol name, or line number.
If this radio button is checked, the performance analysis target range for wizard processing (performance analysis) is specified by a file name, symbol name, or line number.

Caution 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).

Cautions 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. To display a file name as the selection item in this combo box, it is necessary that an original execution module has been generated by using a debug option (-g with the NEC compiler, -g or -G with the GHS compiler, or a debug level of PLAIN/MULTI).

(c) [Line Number] specification area

This area is used to select the performance analysis start position (line number) for wizard processing (performance analysis).

Caution 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

(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).

Cautions 1. This specification area 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. To display a file name as the selection item in this combo box, it is necessary that an original execution module has been generated by using a debug option (-g with the NEC compiler, -g or -G with the GHS compiler, or a debug level of PLAIN/MULTI).

(e) [Line Number] specification area

This area is used to select the performance analysis end position (line number) for wizard processing (performance analysis).

Caution 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

(6) Function buttons

The following buttons are available.

(a) <<Back> button

Closes this panel and opens the Execution Item Selection Panel.

(b) <Next>> button

Closes this panel and then opens the Instruction Cache Selection Panel.

Caution This panel is invalid if the following condition is satisfied.

- If the input contents of the items on this panel are illegal

(c) <Cancel> button

Closes this panel without executing wizard processing (performance analysis).

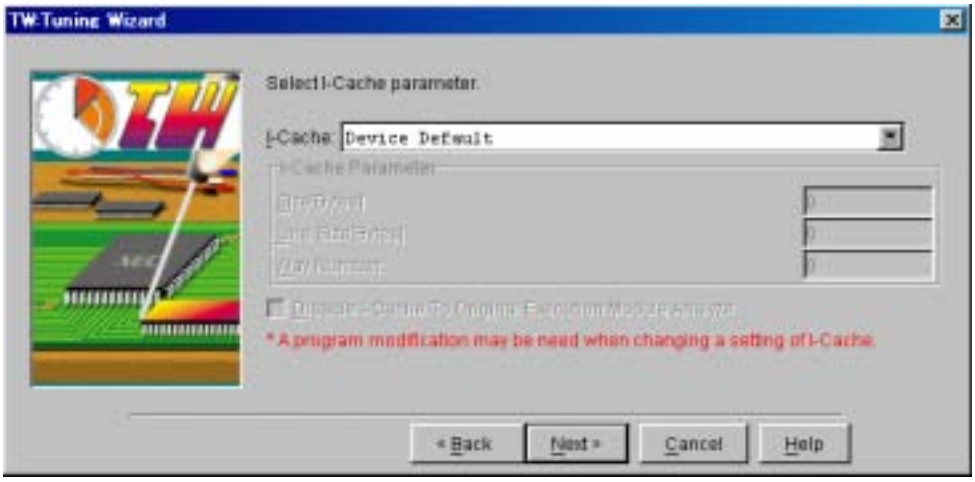
(d) <Help> button

Starts the HTML help.

Instruction Cache Selection Panel

This panel is used to specify the type and parameters of the instruction cache used for wizard processing (performance analysis or tuning).

Figure 6-10 Instruction Cache Selection Panel



This section explains the following items.

- How to open this panel
- Areas of this panel

How to open this panel

This panel can be opened as follows.

- On the Execution Analysis Range Selection Panel,
  - Click the <Next>> button.
- On the Fast Access Memory Area Selection Panel,
  - Click the <<Back> button.
- On the Allocation Destination Selection Panel,
  - Click the <<Back> button.

Areas of this panel

This panel consists of the following items.

- Title bar
- Message display area
- [I-Cache] combo box
- Instruction cache parameter specification area
- [Disable I-Cache To Original Execution Module Analyze] check box

- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Closes this panel without executing wizard processing (performance analysis or tuning).

(2) Message display area

This area displays messages explaining this panel.

(3) [I-Cache] combo box

This combo box is used to select the type of instruction cache used for wizard processing (performance analysis or tuning).

Caution TW can set the instruction cache selected in this combo box as the execution engine, but does not set the instruction cache as described in the Instruction Cache, Data Cache NB85E, NB85ET User's Manual. To use the instruction cache, therefore, processing to set the instruction cache must be added to the startup routine.
With an NEC simulator, the instruction cache does not have to be set.

(4) 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] 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 the number of ways specification area".

Cautions 1. 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

2. 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 simulator.

(b) [Line Size] specification area

This area is used to specify the line size (in bytes) of the instruction cache used for wizard processing (performance analysis and tuning).

Caution 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).

Cautions 1. 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

2. 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 simulator.

(5) [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.
If this check box is checked, the instruction cache is made invalid when the performance of the original execution module is analyzed.

Caution This check box is invalid if either 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

(6) Function buttons

The following buttons are available.

(a) <<Back> button

Closes this panel and then opens the Execution Analysis Range Selection Panel.

(b) <Next>> button

Closes this panel and then opens the Fast Access Memory Area Selection Panel if the [Tune and analyze a tuned execution module.] check box on the Execution Item Selection Panel is checked, and closes this panel and then opens the Allocation Destination Selection Panel if the [Tune and analyze a tuned execution module.] check box on the Execution Item Selection Panel is not checked.

Caution This button is invalid if the following condition is satisfied.

- If data input to the parameters on this panel is illegal.

(c) <Cancel> button

Closes this panel without executing wizard processing (performance analysis or tuning).

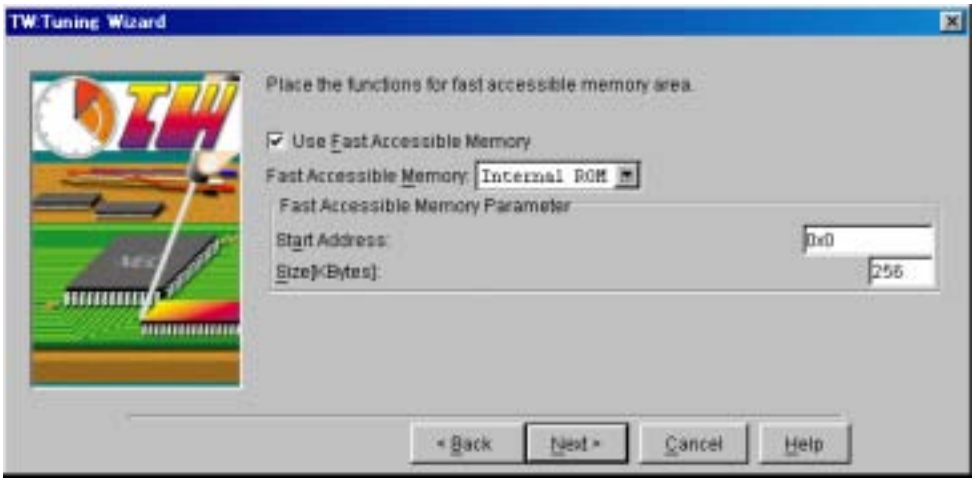
(d) <Help> button

Starts the HTML help.

Fast Access Memory Area Selection Panel

This panel is used to specify the type and parameters of the fast access memory area used for wizard processing (tuning).

Figure 6-11 Fast Access Memory Area Selection Panel



This section explains the following items.

- How to open this panel
- Areas of this panel

How to open this panel

This panel can be opened as follows.

- On the Instruction Cache Selection Panel when the [Tune and analyze a tuned execution module.] check box on the Execution Item Selection Panel is checked,
  - Click the <Next>> button.
- On the Tuning Method Selection Panel,
  - Click the <<Back> button.

Areas of this panel

This panel consists of the following items.

- Title bar
- Message display area
- [Use Fast Accessible Memory] check box
- [Fast Accessible Memory] combo box
- Fast access memory area parameter specification area
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Closes this panel without executing wizard processing (tuning).

(2) Message display area

This area displays messages explaining this panel.

(3) [Use East Accessible Memory] check box

This check box is used to specify whether a function causing the execution performance of the execution module to drop in wizard processing (tuning) is given priority and allocated to the fast access memory area (such as internal ROM).

If this check box is checked, a function causing the execution performance of the execution module to drop in wizard processing (tuning) is given priority and allocated to the fast access memory area (such as internal ROM).

(4) [Fast Accessible Memory] combo box

This combo box is used to specify the type of fast access memory area used for wizard processing (tuning).

Caution This combo box is invalid if the following condition is satisfied.

- If the [Use East Accessible Memory] check box is not checked

(5) 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).

Caution This specification area is invalid if the following condition is satisfied.

- If the [Use East Accessible Memory] check box is not checked

(b) [Size] specification area

This area is used to specify the size (in Kbytes) of the fast access memory area used for wizard processing (tuning).

Caution This specification area is invalid if the following condition is satisfied.

- If the [Use East Accessible Memory] check box is not checked

(6) Function buttons

The following buttons are available.

(a) <<Back> button

Closes this panel and opens the Instruction Cache Selection Panel.

(b) <Next>> button

Closes this panel and then opens the Tuning Method Selection Panel.

Caution This button is invalid if the following condition is satisfied.

- If data input to the parameters on this panel is illegal

(c) <Cancel> button

Closes this panel without executing wizard processing (tuning).

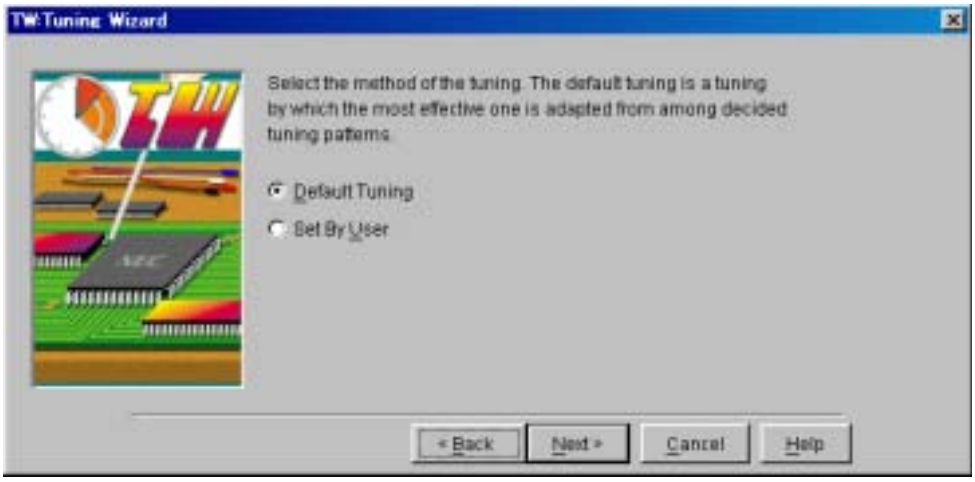
(d) <Help> button

Starts the HTML help.

Tuning Method Selection Panel

This panel is used to specify the method of tuning the user program or execution module (standard pattern or user-specified pattern).

Figure 6-12 Tuning Method Selection Panel



This section explains the following items.

- How to open this panel
- Areas of this panel

How to open this panel

This panel can be opened as follows.

- On the Fast Access Memory Area Selection Panel,
  - Click the <Next>> button.
- On the Fast Access Memory Area Tuning Selection Panel,
  - Click the <<Back> button.
- On the Allocation Destination Selection Panel,
  - Click the <<Back> button.

Areas of this panel

This panel consists of the following items.

- Title bar
- Message display area
- Tuning method specification area
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Closes this panel without executing wizard processing (performance analysis or tuning).

(2) Message display area

This area displays messages explaining this panel.

(3) Tuning method specification area

This area is used to make settings related to the tuning method of the user program and execution module.
This specification area consists of the following radio buttons.

(a) [Default Tuning] radio button

This radio button is used to specify whether the user program or execution module is tuned by the standard pattern (tuning pattern recommended by TW).
If this radio button is checked, tuning is performed by the standard pattern (tuning pattern recommended by TW).

(b) [Set By User] radio button

This radio button is used to specify whether the user program or execution module is tuned by a user-specified pattern. If this radio button is checked, tuning is performed using a user specified pattern.

Caution "Setting of each tuning item" necessary when this radio button is checked is performed during wizard processing (using 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.

(4) Function buttons

The following function buttons are available.

(a) <<Back> button

Closes this panel and opens the Fast Access Memory Area Selection Panel.

(b) <Next>> button

Closes this panel and opens the Allocation Destination Selection Panel when the [Default Tuning] radio button is checked.
Closes this panel and opens the Fast Access Memory Area Tuning Selection Panel when the [Set By User] radio button is checked.

(c) <Cancel> button

Closes this panel without executing wizard processing (performance analysis or tuning).

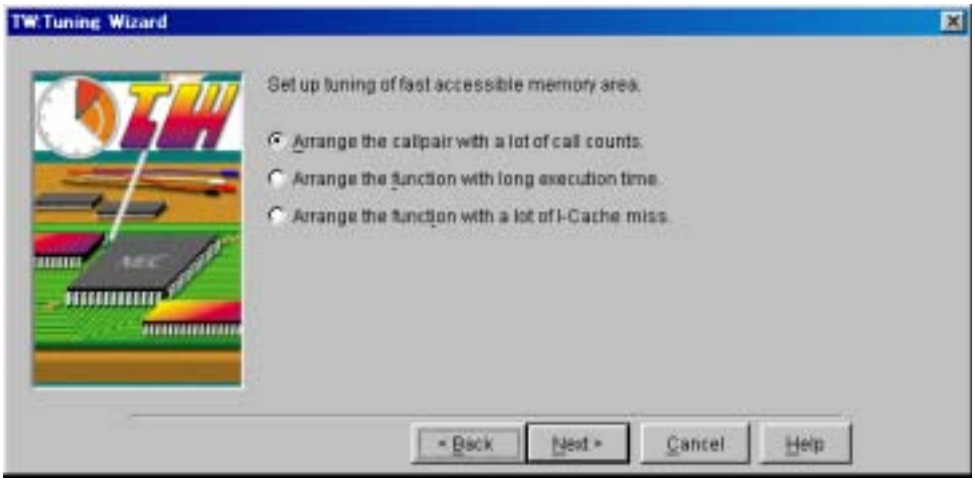
(d) <Help> button

Starts the HTML help.

Fast Access Memory Area Tuning Selection Panel

This panel is used to specify the method of allocating functions in the fast access memory area (such as internal ROM) ,based on the callpair information that was acquired when the performance of the user program or execution module was analyzed.

Figure 6-13 Fast Access Memory Area Tuning Selection Panel



This section explains the following items.

- How to open this panel
- Areas of this panel

How to open this panel

This panel can be opened as follows.

- On the Tuning Method Selection Panel,
  - Check the [Set By User] check box and click the <Next>> button.
- On the Instruction Cache Tuning Selection Panel,
  - Click the <<Back> button.

Areas of this panel

This panel consists of the following items.

- Title bar
- Message display area
- Function allocation method specification area
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Closes this panel without executing wizard processing (performance analysis or tuning).

(2) Message display area

Displays messages explaining this panel.

(3) 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 the internal ROM) based on the information (such as callpair information and execution time information) that was acquired when the performance of the user program or 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 the functions that have been called many times (calling functions and called functions), a function with a 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 the user program or execution module was analyzed.
When this radio button is checked, among the 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.

Caution 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 the user program or execution module was analyzed.
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.

Cautions 1. This radio button is invalid if all the following conditions are 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 Execution Item 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 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. If this radio button is checked, a function with many instruction cache misses is given priority and allocated to the fast access memory area.

Caution 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

(4) Function buttons

The following function buttons are available.

(a) <<Back> button

Closes this panel and opens the Tuning Method Selection Panel.

(b) <Next>> button

Closes this panel and then opens the Instruction Cache Tuning Selection Panel.

(c) <Cancel> button

Closes this panel without executing wizard processing (performance analysis or tuning).

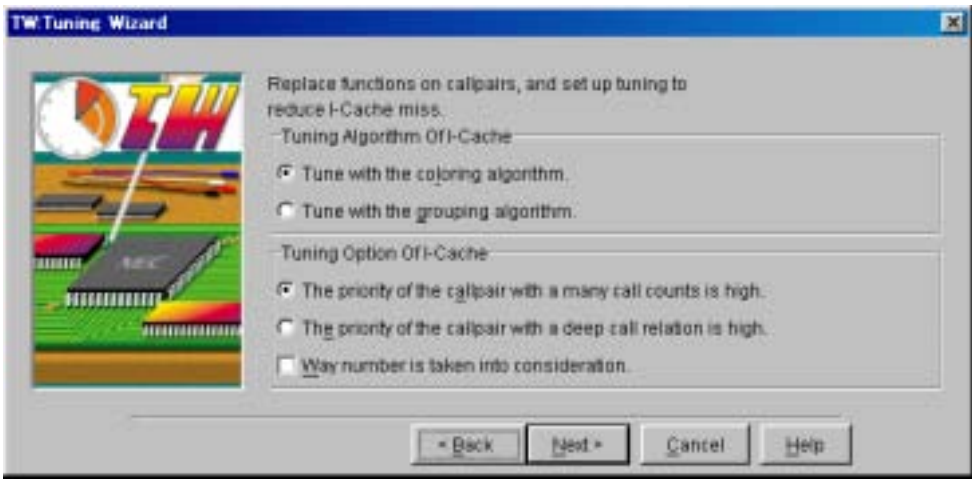
(d) <Help> button

Starts the HTML help.

Instruction Cache Tuning Selection Panel

This panel is used to specify the method of tuning the instruction cache (tuning algorithm and allocation priority correction option), based on the callpair information that was acquired when the performance of the user program or execution module was analyzed.

Figure 6-14 Instruction Cache Tuning Selection Panel



This section explains the following items.

- How to open this panel
- Areas of this panel

How to open this panel

This panel can be opened as follows.

- On the Fast Access Memory Area Tuning Selection Panel,
  - Click the <Next>> button.
- On the Callpair Correction Option Selection Panel,
  - Click the <<Back> button.

Areas of this panel

This panel consists of the following items.

- Title bar
- Message display area
- Tuning algorithm specification area
- Allocation priority correction option specification area
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Closes this panel without executing wizard processing (performance analysis or tuning).

(2) Message display area

This area displays messages explaining this panel.

(3) 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 the user program or execution module was analyzed.
This specification area consists of the following radio buttons.

(a) [Tune with the coloring algotithm.] radio button

This button is used to specify whether functions are allocated to the instruction cache by a coloring algorithm. If this button is checked, functions are allocated to the instruction cache by a coloring algorithm.

Cautions 1. The coloring algorithm allocates functions in the instruction cache based on the callpair information (calling relationship between functions) of each function.

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. If this button is checked, functions are allocated to the instruction cache by a grouping algorithm.

Cautions 1. The grouping algorithm groups functions in a close calling relationship and allocates functions in group units to the instruction cache, based on the callpair information (calling relationship between functions) of each function and assuming the size of the instruction cache to be the upper-limit value of the group size.

2. This radio button is invalid if the following condition is satisfied.

- If the size of the instruction cache is 0

(4) 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 the user program or 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 function that is called many times is given priority and allocated to the instruction cache. If this radio button is checked, a function that is called many times is given priority and allocated to the instruction cache.

Caution 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 function close to the end of a calling relationship is given priority and allocated to the instruction cache. If this radio button is checked, a function close to the end of a calling relationship is given priority and allocated to the instruction cache.

Caution 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.
If this check box is checked, the cache mechanism (the number of ways) is taken into consideration when a function is allocated to the instruction cache.

Caution This radio button is invalid if either of the following conditions is satisfied.

- If the size of the instruction cache is 0
- If the cache mechanism (number of ways) of the instruction cache is less than 2

(5) Function buttons

The following buttons are available.

(a) <<Back> button

Closes this panel and then opens the Fast Access Memory Area Tuning Selection Panel.

(b) <Next>> button

Closes this panel and then opens the Callpair Correction Option Selection Panel.

(c) <Cancel> button

Closes this panel without executing wizard processing (performance analysis or tuning).

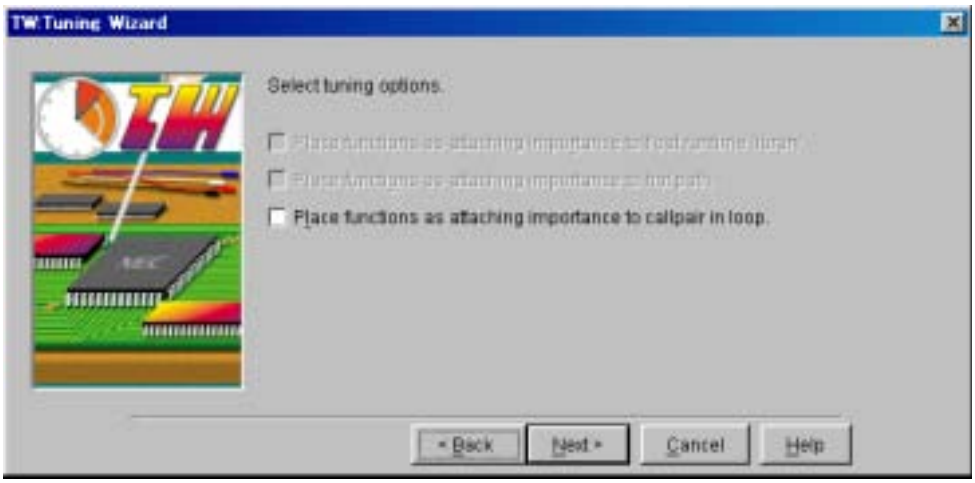
(d) <Help> button

Starts the HTML help.

Callpair Correction Option Selection Panel

This panel is used to specify the method of tuning functions to be reallocated (callpair information correction method), based on the callpair information that was acquired when the performance of the user program or execution module was analyzed.

Figure 6-15 Callpair Correction Option Selection Panel



This section explains the following items.

- How to open this panel
- Areas of this panel

How to open this panel

This panel can be opened as follows.

- On the Instruction Cache Tuning Selection Panel,
  - Click the <Next>> button.
- On the Priority Allocation Function Selection Panel,
  - Click the <<Back> button.

Areas of this panel

This panel consists of the following items.

- Title bar
- Message display area
- Callpair Correction Option Selection Panel
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Closes this panel without executing wizard processing (performance analysis or tuning).

(2) Message display area

This area displays messages explaining this panel.

(3) Callpair correction option specification area

This area is used to make settings related to the tuning method (callpair information correction method) of the functions to be reallocated, based on the callpair information that was acquired when the performance of the user program or 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 functions (such as expf and cosf) included in the float runtime library are given priority and allocated to the instruction cache or fast access memory area.
If this check box is checked, the functions (such as expf and cosf) included in the float runtime library are given priority and allocated to the instruction cache or fast access memory area.

Cautions 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 all the following conditions are satisfied.

- If the user development environment is the GHS tool chain
- If the [Use libtw] check box in the Option Dialog Box is not checked.

3. This check box is invalid if the following condition is satisfied.

- If the functions (such as expf and cosf) included in the float runtime library are not included in the user program or execution module

(b) [Place functions as attaching importance to hot path.] check box

This check box specifies whether a function corresponding to a hot path (callpair function related to the applicable function, based on the execution time information of each function including child functions) is given priority and allocated to the instruction cache or fast access memory area. 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.

Cautions 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.
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.

Caution 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

Caution If two or more check boxes are specified at the same time, TW corrects the callpair information in the following order.

float > hot path > loop processing

(4) Function buttons

The following function buttons are available.

(a) <<Back> button

Closes this panel and then opens the Instruction Cache Tuning Selection Panel.

(b) <Next>> button

Closes this panel and then opens the Priority Allocation Function Selection Panel.

(c) <Cancel> button

Closes this panel without executing wizard processing (performance analysis or tuning).

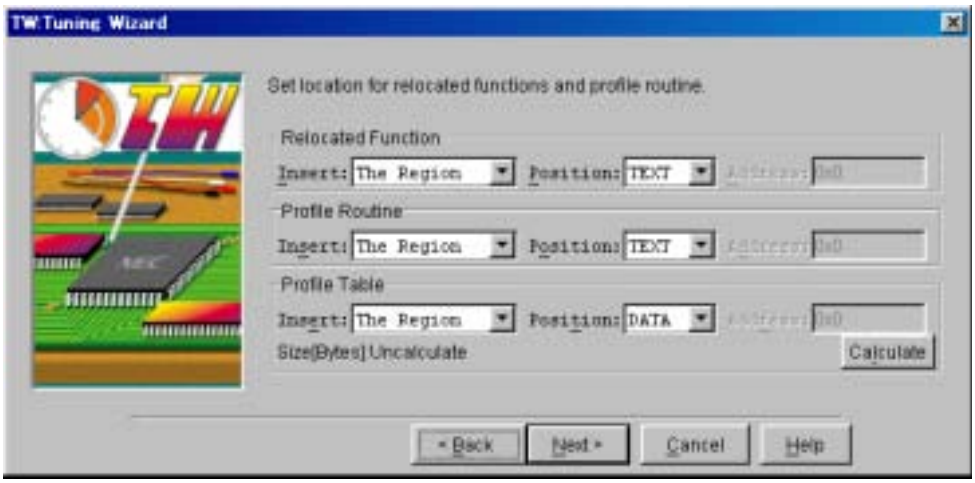
(d) <Help> button

Starts the HTML help.

Allocation Destination Selection Panel

This panel is used to specify the memory area in which to reallocate a function that is subject to reallocation during tuning, and the memory area where the profile routine and profile table output as intermediate files during performance analysis or tuning are to be allocated.

Figure 6-16 Allocation Destination Selection Panel



This section explains the following items.

- How to open this panel
- Areas of this panel

How to open this panel

This panel can be opened as follows.

- On the Instruction Cache Selection Panel when the [Tune and analyze a tuned execution module.] check box on the Execution Item Selection Panel is not checked,
  - Click the <Next>> button.
- On the Priority Allocation Function Selection Panel,
  - Click the <Next>> button.
- On the Pre-execution Confirmation Panel,
  - Click the <<Back> button.

Areas of this panel

This panel consists of the following items.

- Title bar
- Message display area
- Reallocation target function specification area
- Profile routine specification area

- Profile table specification area
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Closes this panel without executing wizard processing (performance analysis or tuning).

(2) Message display area

This area displays messages explaining this panel.

(3) Reallocation target function specification area

This specification area is used to make settings related to the reallocation memory area of the function to be reallocated during tuning.
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 (in specified area, before specified area, after specified area, or at specified address) of the memory area for reallocating the function subject to reallocation during tuning.

Caution The "The Region" menu of this combo box can be selected only when the user's development environment is the NEC tool chain.
Menus other than the "The Region" menu of 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 reallocating the function subject to reallocation during tuning.

Caution This combo box is invalid if the following condition is satisfied.

- 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 reallocating the function subject to reallocation during tuning.

Caution This specification area is invalid if the following condition is satisfied.

- If a menu other than "Set Address" is selected in the [Insert] combo box

(4) Profile routine specification area

This area is used to make settings related to the memory area for reallocating the profile routine output as an intermediate file during performance analysis or tuning.
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 (in specified area, before specified area, after specified area, or at specified address) of the memory area for allocating the profile routine output as an intermediate file during performance analysis or tuning.

Caution The "The Region" menu of this combo box can be selected only when the user's development environment is the NEC tool chain.
Menus other than the "The Region" menu of this combo box are invalid if either of the following conditions is satisfied.

- If the TWROUTINE segment is defined for the target link directive file
- If the [Use probe code for analyzing(for original execution module only).] radio button on the Execution Item Selection Panel is not checked

(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.

Caution This combo box is invalid if the following condition is satisfied.

- 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.

Caution This specification area is invalid if either of the following conditions is satisfied.

- If a menu other than "Set Address" is selected in the [Insert] combo box
- If the [Use probe code for analyzing(for original execution module only).] radio button on the Execution Item Selection Panel is not checked

(5) 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 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 (in specified area, before specified area, after specified area, or at specified address) of the memory area for allocating the profile table output as an intermediate file during performance analysis or tuning.

Caution The "The Region" menu of this combo box can be selected only when the user's development environment is the NEC tool chain.
Menus other than the "The Region" menu of this combo box are invalid if the following condition is satisfied.

- 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.

Caution This combo box is invalid if the following condition is satisfied.

- If the TWTABLE 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 table output as an intermediate file during performance analysis or tuning.

Caution This specification area is invalid if the following condition is satisfied.

- If a menu other than "Set Address" is selected in the [Insert] combo box

(d) Size display area

This area displays the size (in bytes) of the profile table that can be acquired by clicking the <Calculate> button.

(e) <Calculate> button

This button calculates the size (in bytes) of the profile table and displays the result of calculation in the size display area.

(6) Function buttons

The following buttons are available.

(a) <<Back> button

Closes this panel and opens the panel that was open immediately before (Instruction Cache Selection Panel, Tuning Method Selection Panel, or Priority Allocation Function Selection Panel).

(b) <Next>> button

Closes this panel and then opens the Pre-execution Confirmation Panel.

(c) <Cancel> button

Closes this panel without executing wizard processing (performance analysis or tuning).

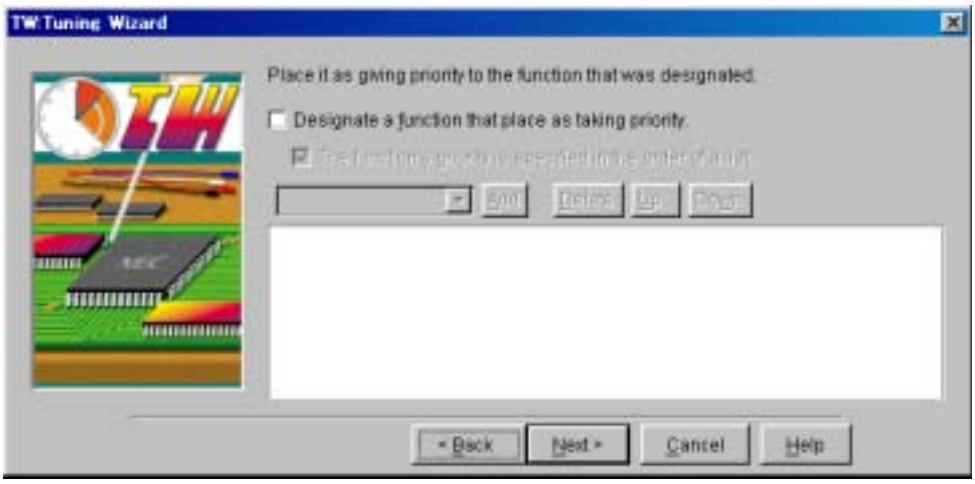
(d) <Help> button

Starts the HTML help.

Priority Allocation Function Selection Panel

This panel is used to specify a function that is given priority and reallocated during wizard processing (tuning).

Figure 6-17 Priority Allocation Function Selection Panel



This section explains the following items.

- How to open this panel
- Areas of this panel

How to open this panel

This panel can be opened as follows.

- On the Callpair Correction Option Selection Panel,
  - Click the <Next>> button.
- On the Allocation Destination Selection Panel,
  - Click the <<Back> button.

Areas of this panel

This panel consists of the following items.

- Title bar
- 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
- Reallocation target function specification area
- Reallocation target function display area
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Closes this panel without executing wizard processing (performance analysis or tuning).

(2) Message display area

This area displays messages explaining this panel.

(3) [Designate a function that place as taking priority.] check box

This check box is used to specify whether a function that is given priority and allocated is specified for wizard processing (tuning).
If this check box is checked, a function that is given priority and allocated is specified.

(4) [The function's priority is specified in the order of a list.] check box

This check box is used to specify whether function reallocation processing is executed during wizard processing (tuning) in the order in which the functions to be reallocated are displayed on this panel (top: highest priority, bottom: lowest priority).
If this check box is checked, function reallocation processing is executed during wizard processing (tuning) in the order in which the functions to be reallocated are displayed on this panel (top: highest priority, bottom: lowest priority).

Caution 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

(5) Reallocation target function specification area

This area is used to specify a function that is given priority and allocated during wizard processing (tuning).
This specification area consists of the following combo box and buttons.

(a) Reallocation target function name specification combo box

This box is used to select the name of the function that is given priority and reallocated during wizard processing (tuning).

Caution 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 registers the function selected in the reallocation target function name specification combo box as a function that is given priority and reallocated.
By clicking this button, the name of the function specified in the reallocation target function name specification combo box is added to the lowest position in the reallocation target function display area.

Caution 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 reallocation target function name specification combo box as a function that is given priority and reallocated.
By clicking this button, the name of the function specified in the reallocation target function name specification combo box is deleted from the reallocation target function display area.

Caution 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 reallocation target function name specification combo box.

By clicking this button, the position of the name of the function selected in the reallocation target function name specification combo box is moved one level up in the reallocation target function display area.

Caution 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 reallocation target function name specification combo box.
By clicking this button, the position of the name of the function selected in the reallocation target function name specification combo box is moved one level down in the reallocation target function display area.

Caution 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

(6) Reallocation target function display area

This area displays a list of the names of the functions to be given priority and reallocated during wizard processing (tuning).

Caution 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

(7) Function buttons

The following buttons are available.

(a) <<Back> button

Closes this panel and opens the Callpair Correction Option Selection Panel.

(b) <Next>> button

Closes this panel and opens the Allocation Destination Selection Panel.

(c) <Cancel> button

Closes this panel without executing wizard processing (tuning).

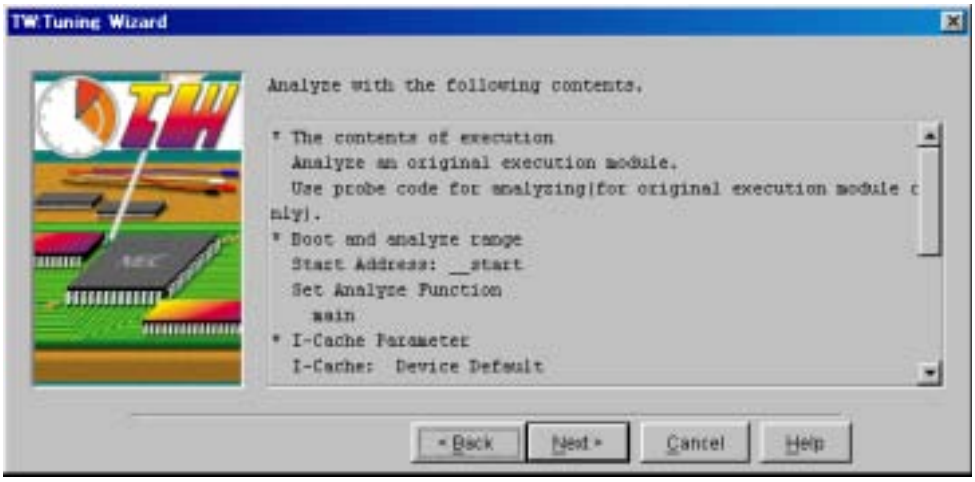
(d) <Help> button

Starts the HTML help.

Pre-execution Confirmation Panel

This panel reports that setting of the performance analysis or tuning items required for TW to execute wizard processing (performance analysis or tuning) has been completed.

Figure 6-18 Pre-execution Confirmation Panel



This section explains the following items.

- How to open this panel
- Areas of this panel

How to open this panel

This panel can be opened as follows.

- On the Allocation Destination Selection Panel,
  - Click the <Next>> button.
- On the Performance Analysis/Tuning Execution Panel,
  - Click the <<Back> button.

Areas of this panel

This panel consists of the following items.

- Title bar
- Message display area
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Closes this panel without executing wizard processing (performance analysis or tuning).

(2) Message display area

This area displays a message reporting that setting of the performance analysis or tuning items required for TW to execute wizard processing (performance analysis or tuning) has been completed.

(3) Function buttons

The following buttons are available.

(a) <<Back> button

Closes this panel and opens the Allocation Destination Selection Panel.

(b) <Next>> button

Closes this panel and opens the Performance Analysis/Tuning Execution Panel.

(c) <Cancel> button

Closes this panel without executing wizard processing (performance analysis or tuning).

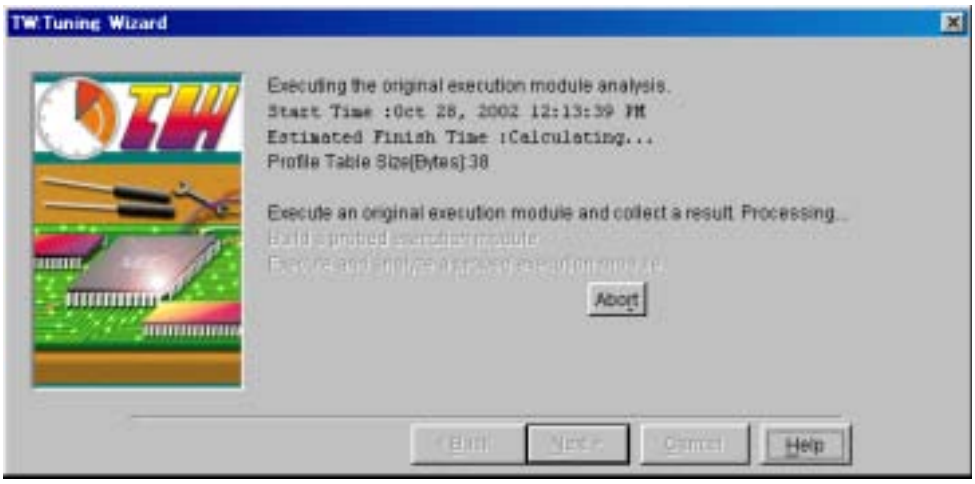
(d) <Help> button

Starts the HTML help.

Performance Analysis/Tuning Execution Panel

This panel reports that TW is executing wizard processing (performance analysis or tuning).

Figure 6-19 Performance Analysis/Tuning Execution Panel



This section explains the following items.

- How to open this panel
- Areas of this panel

How to open this panel

This panel can be opened as follows.

- On the Pre-execution Confirmation Panel,
  - Click the <Next>> button.

Areas of this panel

This panel consists of the following items.

- Title bar
- Message display area
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Closes this panel without executing wizard processing (performance analysis or tuning).

Caution This button is invalid if the following condition is satisfied.

- If TW is executing performance analysis or tuning

(2) Message display area

This area displays a message reporting that TW is executing wizard processing (performance analysis or tuning).

(a) <Abort> button

Stops wizard processing (performance analysis or tuning).

Caution It may take time for the wizard processing to actually stop after this button is clicked.

(b) <Retry> button

Re-executes wizard processing (performance analysis or tuning).

(3) Function buttons

The following buttons are available.

(a) <<Back> button

Closes this panel and opens the Pre-execution Confirmation Panel.

(b) <Next>> button

This button cannot be clicked because it is invalid.

(c) <Cancel> button

Closes this panel without executing wizard processing (performance analysis or tuning).

Caution This button is invalid if the following condition is satisfied.

- If TW is executing performance analysis or tuning

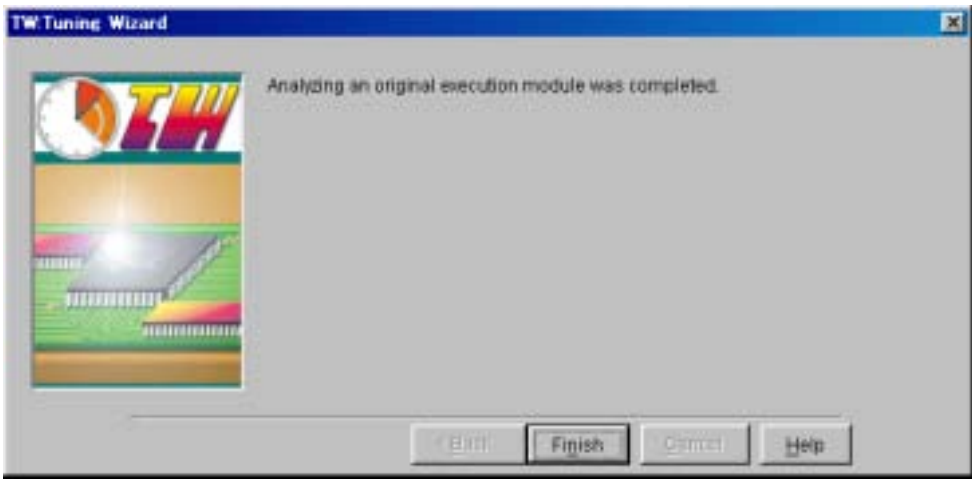
(d) <Help> button

Starts the HTML help.

Execution Completion Confirmation Panel

This panel reports that TW has completed wizard processing (performance analysis or tuning).

Figure 6-20 Execution Completion Confirmation Panel



This section explains the following items.

- How to open this panel
- Areas of this panel

How to open this panel

This panel is automatically opened when TW has correctly completed wizard processing (performance analysis or tuning).

Areas of this panel

This panel consists of the following items.

- Title bar
- Message display area
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Closes this panel.

(2) Message display area

This area displays a message (such as name of folder storing the output file and the output file name) reporting that TW has correctly completed wizard processing (performance analysis or tuning).

(3) Function buttons

The following buttons are available.

(a) <<Back> button

This button cannot be clicked because it is invalid.

(b) <Finish> button

Closes this panel.

(c) <Cancel> button

This button cannot be clicked because it is invalid.

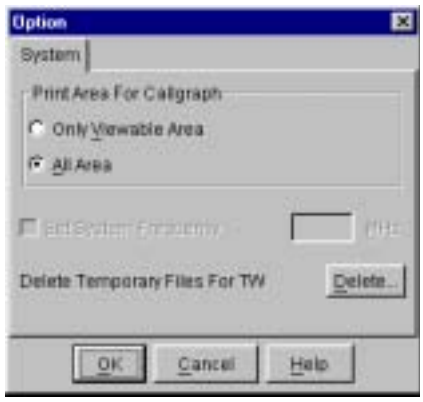
(d) <Help> button

Starts the HTML help.

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 to delete the intermediate files (temporary files) TW output while it was executing performance analysis or tuning processing.

Figure 6-21 Option Dialog Box



This section explains the following items.

- How to open this dialog box
- Areas of this dialog box

How to open this dialog box

This dialog box can be opened as follows.

- On the Main Window,
  - Select [Tool] menu -> [Option...].

Areas of this dialog box

This dialog box consists of the following items.

- Title bar
- System-related tab
- GHS-related tab (when GHS tool chain is used)
- Function buttons

(1) Title bar

This title bar has the following button.

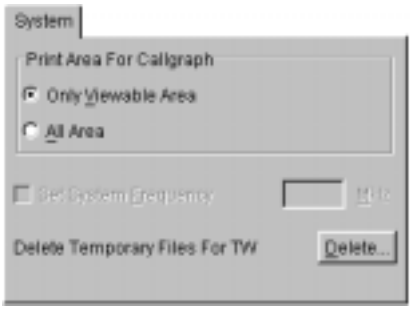
(a) <Close> button

Closes this dialog box without executing processing corresponding to the information specified by the system-related tab and GHS-related tab.

(2) System-related tab

This tab is used to set options related to the overall processing of TW.

Figure 6-22 [System] Tab



This tab consists of the following specification areas and buttons.

(a) Printing method specification area

- [Only Viewable Area] radio button

  Specifies whether to print the Callgraph Window displayed in the internal window display area of the Main Window as a displayed image.
  If this radio button is checked, the Callgraph Window is printed as a displayed image.

- [All Area] radio button

  Specifies whether to print the Callgraph Window displayed in the internal window display area of the Main Window as the entire Callgraph Window.
  If this radio button is checked, the Callgraph Window is printed as the entire Callgraph Window.

(b) [Set System Frequency] check box

Specifies whether information (operating frequency of the execution engine) for the operating frequency specification area can be specified. If this check box is checked, information for the operating frequency specification area (operating frequency of the execution engine) can be specified.

Caution This check box is invalid if the following condition is satisfied.

- TW can automatically detect the operating frequency of the execution engine (debugger or simulator).

(c) Operating frequency specification area

Specifies the operating frequency (in MHz) of the execution engine (such as a debugger or simulator).

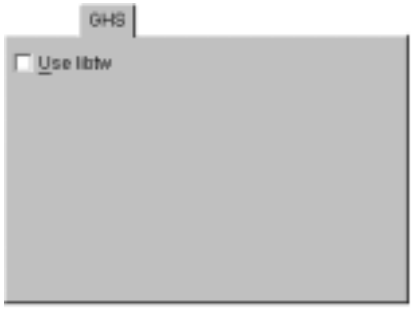
(d) <Delete...> button

Deletes the intermediate (temporary) files output by TW while it was executing performance analysis or tuning processing.

(3) GHS-related tab (when GHS tool chain is used)

This tab is used to set options related to the library file that is used to create a load module. This tab is displayed only when the user's development environment is the GHS tool chain.

Figure 6-23 [GHS] Tab



This tab consists of the following check boxes.

(a) [Use libtw] check box

Specifies whether libtw*.a, a system library for TW, is used when performance analysis or tuning processing is executed.
If this check box is checked, the libtw*.a system library for TW is used.

(4) Function buttons

The following function buttons are available.

(a) <OK> button

Executes processing corresponding to the information specified by the system-related tab and GHS-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 and GHS-related tab.

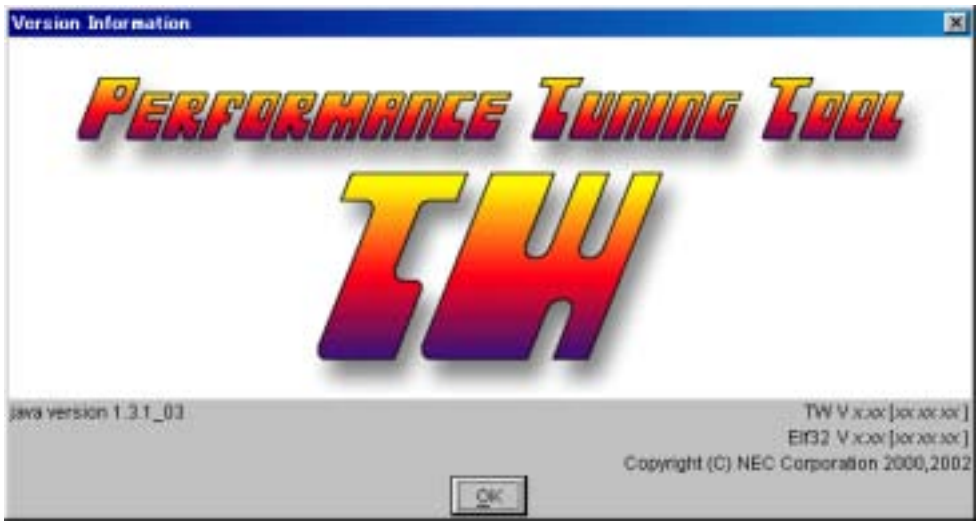
(c) <Help> button

Starts the HTML help.

Version Information Dialog Box

This dialog box displays the version information of TW (such as the version and copyright).

Figure 6-24 Version Information Dialog Box



This section explains the following items.

- How to open this dialog box
- Areas of this dialog box

How to open this dialog box

This dialog box can be opened as follows.

- On the Main Window,
  - Select [Help] menu -> [About...].

Areas of this dialog box

This dialog box consists of the following items.

- Title bar
- Message display area
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Closes this dialog box.

(2) Message display area

Displays the version information (such as the version and copyright) of TW.

(3) Function buttons

The following function button is available.

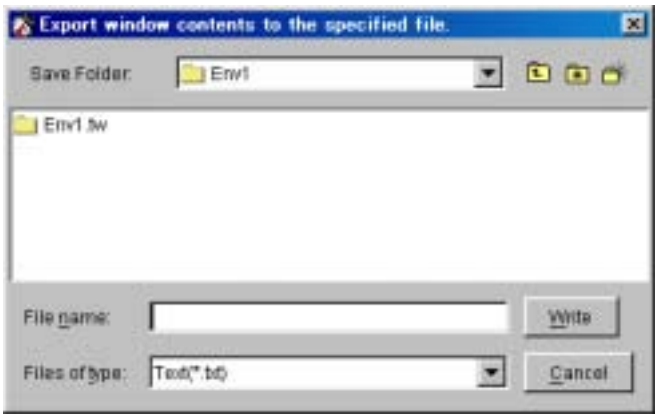
(a) <OK> button

Closes this dialog box.

File Selection Dialog Box

This dialog box is used to specify the files required by TW to execute performance analysis or tuning processing (such as build files and timer control code blocks), and files to save the execution status of TW (such as a project information saving file).

Figure 6-25 File Selection Dialog Box



This section explains the following items.

- How to open this dialog box
- Areas of this dialog box

How to open this dialog box

This dialog box can be opened as follows.

- On the Main Window,
  - Select [File] menu -> [Open MULTI Project...].
- On the Execution Item Selection Panel,
  - Click the <Refer...> button.
- On the Main Window, after moving the focus to the Function Information Table Window, Execution Module Information Window, or Function Detail Information Window,
  - Select [File] menu -> [Save TW Project].
  - Select [File] menu -> [Export...].

Areas of this dialog box

This dialog box consists of the following items.

- Title bar
- Opening location specification combo box
- File name display area
- File name specification area

- File type specification combo box
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

This button closes this dialog box without reporting to TW that the file specified in the file name specification area is a file required by TW to execute performance analysis or tuning processing, and without saving the execution status of TW to the file specified in the file name specification area.

(2) Opening location specification combo box

Selects the name of the folder where the corresponding file (such as a project information saving file or timer control code block) is stored.

(3) File name display area

Lists and displays 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.

Cautions 1. If a filename is clicked, the file is selected and displayed in the file name specification area.

2. If a filename 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 all the files in the specified folder are listed in the file name display area.

(4) File name specification area

In this area, specify a file (such as a project information saving file or timer control code block) required by TW to execute performance analysis or tuning processing.

(5) File type specification combo box

In this combo box, select the type of the file to be displayed in the file name display area.

(6) Function buttons

The following function buttons are available.

(a) <Up One Level> button

Changes the folder to be displayed in the file name display area to a folder one level up.
Note, however, that this button is invalid while the root folder is displayed in the file name display area.

(b) <Home> button

Changes the folder displayed in the file name display area to the home folder.

(c) <Create New Folder> button

Creates a new folder in the folder currently displayed in the file name display area.

(d) <Open>, <Write> button

These buttons close this dialog box after reporting to TW that the file specified in the file name specification area is required by TW to execute performance analysis or tuning processing, or after saving the execution status of TW to the file specified in the file name specification area.

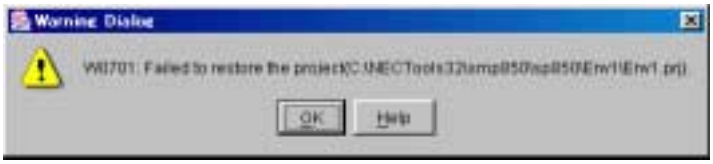
(e) <Cancel> button

Closes this dialog box without reporting to TW that the file specified in the file name specification area is required by TW to execute performance analysis or tuning processing, or without saving the execution status of TW to the file specified in the file name specification area.

Message Dialog Box

This dialog box displays the information to be reported to the user (such as the progress of processing and occurrence of a warning) detected by TW while it was executing performance analysis or tuning processing.

Figure 6-26 Message Dialog Box



This section explains the following items.

- How to open this dialog box
- Areas of this dialog box

How to open this dialog box

This dialog box is automatically opened when TW 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.

Areas of this dialog box

This dialog box consists of the following items.

- Title bar
- Message display area
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Closes this dialog box.

(2) 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) that TW detected while it was executing performance analysis or tuning processing.
For details of the information to be reported to the user (such as the progress of processing and occurrence of a warning), refer to **A.2 Information Messages** and **A.3 Warning Messages**.

(3) Function buttons

The following function buttons are available.

(a) <OK> button

Closes this dialog box.

(b) <Help> button

Starts the HTML help.

Error Dialog Box

This dialog box displays information (such as the occurrence of a fatal error) that has been detected while TW is executing performance analysis processing or tuning processing and that should be reported to the user.

Figure 6-27 Error Dialog Box

This section explains the following items.

- How to open this dialog box
- Areas of this dialog box

How to open this dialog box

This dialog box is automatically opened if TW detects information to be reported to the user (such as the occurrence of a fatal error) while it is executing processing.

Areas of this dialog box

This dialog box consists of the following items.

- Title bar
- Message display area
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Closes this dialog box.

(2) Message display area

This area displays the information (such as the progress of processing and occurrence of an alarm) that is detected while TW is executing performance analysis or tuning processing and that should be reported to the user.
For details of the information to be reported to the user (such as the occurrence of a fatal error), refer to **A.4 Error Messages**.

(3) Function buttons

The following function buttons are available.

(a) <Detail>>> button

Displays details of the information displayed in this dialog box.

(b) <OK> button

Closes this dialog box.

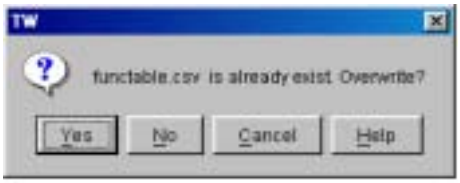
(c) <Help> button

Starts the HTML help.

Verification Dialog Box

This dialog box displays inquiry information (such as a problem that prevents continuation of processing) TW detected while it was executing performance analysis or tuning processing.

Figure 6-28 Verification Dialog Box



This section explains the following items.

- How to open this dialog box
- Areas of this dialog box

How to open this dialog box

This dialog box can be opened as follows.

- On the Main Window,
  - Select [File] menu -> [Exit...].
  - Acquire memory location information.
  - Select [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.
- If the file specified in the File Selection Dialog Box already exists.
- On the [System] tab in the Option Dialog Box,
  - Click the <Delete...> button.

Areas of this dialog box

This dialog box consists of the following items.

- Title bar
- Message display area
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Executes processing corresponding to the contents of the displayed message and closes this dialog box.

(2) Message display area

Displays inquiry information (such as a problem that prevents continuation of processing) TW detected while it was executing performance analysis or tuning processing.

(3) Function buttons

The following function buttons are available.

(a) <Yes> button

Executes processing corresponding to the contents of the displayed message and 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 HTML help.

Input Dialog Box

This dialog box is used to specify information requested by TW (such as the color boundary, display magnification rate, and root function).

Figure 6-29 Input Dialog Box

This section explains the following items.

- How to open this dialog box
- Areas of this dialog box

How to open this dialog box

This dialog box can be opened as follows.

- On the Main Window, after moving the focus to the Callgraph Window,
  - Select [View] menu -> [Callgraph] menu -> [Coloring Threshoud...].
  - Select [View] menu -> [Callgraph] menu -> [Zoom...].
  - Select [View] menu -> [Callgraph] menu -> [Change Root Funsion...].

Areas of this dialog box

This dialog box consists of the following items.

- Title bar
- Message display area
- Request information specification area and request information specification combo box
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

This button closes this dialog box without reporting to TW that the information requested by TW is the information specified in the request information specification area or request information specification combo box.

(2) Message display area

Displays explanation messages related to this dialog box.

(3) Request information specification area and request information specification combo box

In this area and combo box, specify the information requested by TW (such as the color boundary, display magnification rate, and root function).

(4) Function buttons

The following function buttons are available.

(a) <OK> button

Reports to TW that the information requested by TW 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 TW that the information requested by TW is the information specified in the request information specification area or request information specification combo box.

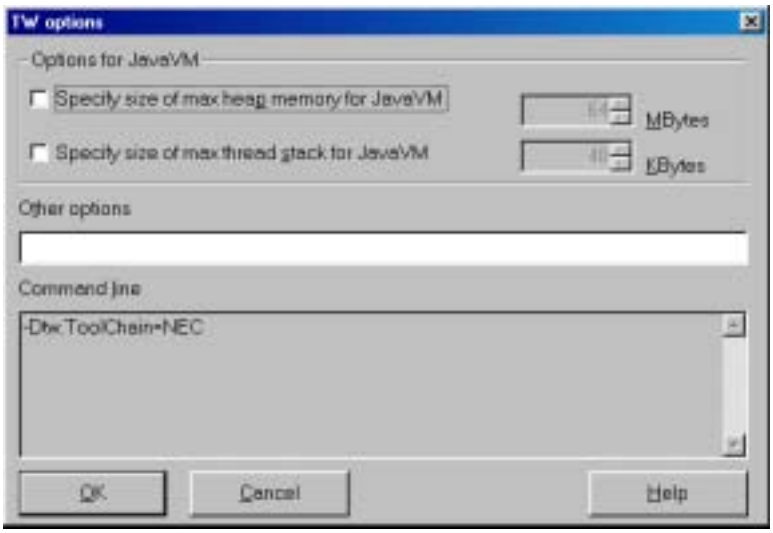
(c) <Help> button

Starts the HTML help.

Start Option Setting Dialog Box

This dialog box is used to specify the start options of TW.

Figure 6-30 Start Option Setting Dialog Box



This section explains the following items.

- How to open this dialog box
- Areas of this dialog box

How to open this dialog box

This dialog box can be opened as follows.

- On the main window of project manager PM plus,
  - Select [Tool] menu -> [TW Options...].

Areas of this dialog box

This dialog box consists of the following items.

- Title bar
- JavaVM option specification area
- Other option specification area
- Start option display area
- Function buttons

(1) Title bar

This title bar has the following button.

(a) <Close> button

Closes this dialog box.

(2) 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

Specifies whether to specify the maximum memory size JavaVM can use as a heap memory area.
If this check box is checked, the maximum size usable as a heap memory area is specified.

(b) Maximum heap memory size specification area

Specifies the maximum size (in Mbytes) JavaVM can use as a heap memory area.

(c) [Specify size of max thread stack for JavaVM] check box

Specifies whether to specify the maximum size JavaVM can use as a thread stack memory area. If this check box is checked, the maximum size usable as a thread stack memory area is specified.

(d) Thread stack size specification area

Specifies the maximum size (in Kbytes) JavaVM can use as a thread stack memory area.

(3) Other option specification area

This area is used to set options related to wizard processing of TW (performance analysis or tuning processing).

Caution For details of the options, refer to **3.2.3 Starting from command line**.

(4) Start option display area

Displays a list of the TW start options (including default options) reported to PM plus, based on the information specified in the JavaVM option specification area and other option specification area.

(5) Function buttons

The following function buttons are available.

(a) <OK> button

Reports the information specified in the JavaVM option specification area and other option specification area to 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 PM plus.

(c) <Help> button

Starts the HTML 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 TW.

A.1 Display Format

A message is generated and displayed in the message display area of the Main Window, Message Dialog Box, or Error Dialog Box if TW detects "information to be reported to the user (such as progress of processing, occurrence of warning, and occurrence of a fatal error)" while it is executing processing.

Figure A-1 Message Display Area in the Main Window

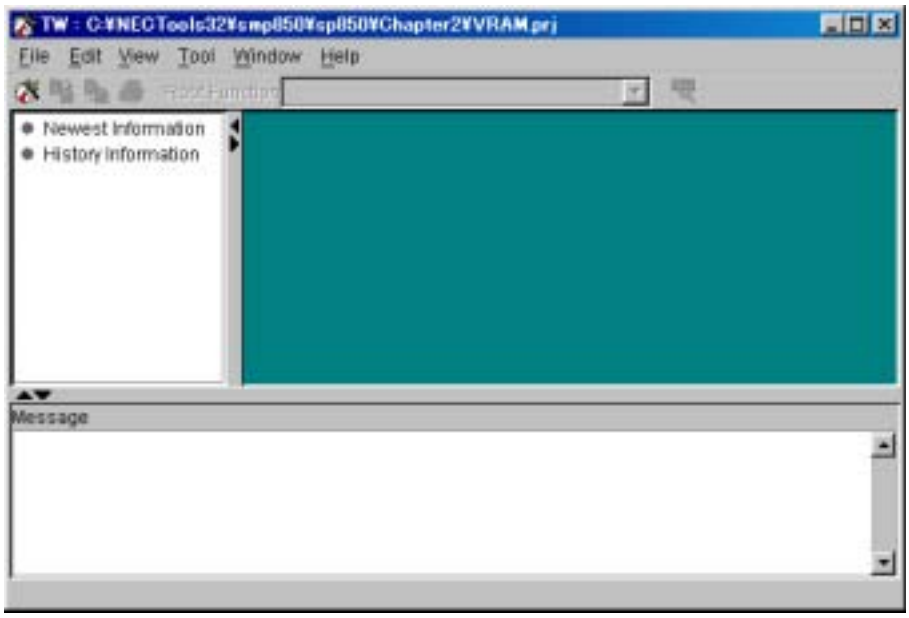


Figure A-2 Message Dialog Box

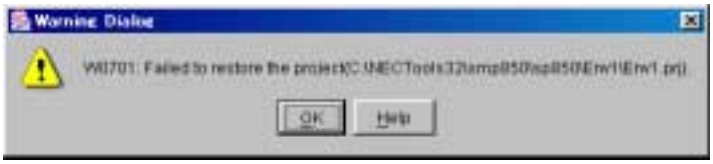


Figure A-3 Error Dialog Box

TW 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, as follows.

I : Information Messages
When TW 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 TW detects information (such as occurrence of 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 TW 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 TW 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.

I0500: Failed to substitute the symbol(%s), but TW ignore it.(%s)

1) Processing to assign symbol %s has been terminated abnormally. As this abnormal termination does not hinder TW continuing processing at present, TW 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 this illegal file declaration does not hinder TW continuing processing at present, TW ignores the illegal 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 declaration %s may have been specified by an assembler option. TW continues processing, assuming that 0 is set as the value of assembler declaration %s.

I0503: Analyzing functions in (%s) source file(%s)...

1) TW reports the 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) TW reports completion of analysis processing on all functions subject to performance analysis, and continues processing.

I0505: Analyzing callpairs in (%s) source file(%s)...

1) TW reports progress %s of callpair information (calling relationship between functions) analysis processing on a function subject to performance analysis in source file %s, and continues processing.

I0506: Process of analyzing callpairs was done.

1) TW reports completion of callpair information (calling relationship between functions) analysis 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 message indicates that execution module information is being analyzed after tuning.

I0511: Process of analyzing the execution module was done.

1) This message indicates completion of analysis of execution module information.

I1400: Could not to set or get the debugger information, but TW ignore it.

1) Cache information output by the debugger could not be detected. As abnormal termination of cache information detection processing does not hinder TW continuing processing, TW ignores the abnormal termination of cache information detection processing, and continues processing.
2) The debugger could not be set in the peripheral mode. As abnormal termination of peripheral mode setting processing does not hinder TW continuing processing, TW ignores the abnormal termination of peripheral mode setting processing, and continues processing.

I1500: Top address of region %s moved from %s to %s, because which was overlapped to existing region.

1) The size of the area (segment or section) %s specified by the link directive file may cause overlapping of the memory area during allocation of the execution module to memory after tuning. TW continues processing, assuming that %s is specified as the first address %s specified by the link directive file.

I1600: %s

1) TW 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 abnormal termination of device information detection processing does not hinder TW continuing processing, TW ignores the abnormal termination of device information detection processing and continues processing.

A.3 Warning Messages

The warning messages that are output if TW detects information (such as occurrence of 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 warning) is detected.

W0100: The version of MULTI is unknown. TW regards as MULTI2000.

1) The version of MULTI may not be supported by TW. TW starts processing assuming that the execution engine is MULTI2000.

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. TW continues processing assuming the current set value.

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 clocks tw_clock_count per timer count" in the timer control code block. TW continues processing assuming that 1 is set as the "number of clocks per timer count".

W0700: TW project file(%s) is ignored.

1) The contents of project file %s may have been updated or destroyed. TW 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. TW does not restore the environment of project file %s and starts processing in the default environment.
2) Project file %s could not be detected. TW does not restore the environment of project file %s and starts processing in the default environment.

W1400: %s

1) Performance analysis processing of TW has been terminated abnormally. TW displays the response log %s of the MULTI command that has been terminated abnormally, and stops processing.

A.4 Error Messages

The error messages that are output when TW 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) The attribute of file %s or the folder storing file %s may limit an access. 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 folder storing the file may be limited. Check the attribute of 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 the file or folder storing the file may be limited. Check the attribute of the file or folder storing the file in accordance with the contents of displayed message %s.

E0003: Could not read the file(%s).

1) Access to the attribute of the file or folder storing the file may be limited. Check the attribute of the file or folder storing the file in accordance with the contents of displayed message %s.

E0004: Could not write the file(%s).

1) Access to the attribute of the file or folder storing the file may be limited. Check the attribute of the file or folder storing the file in accordance with the contents of displayed message %s.
2) The memory area necessary for writing data to file %s may run short. Delete unnecessary files to secure a sufficient memory area.
3) Windows may be unstable and the file may be locked. Restart Windows.

E0005: Could not delete the file(%s).

1) Access to the attribute of the file or folder storing the file may be limited. Check the attribute of the file or folder storing the file in accordance with the contents of displayed message %s.
2) Windows may be unstable and the file 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 v1.3) supported by TW.
3) TW may have a bug. Consult NEC.

E0007: Could not read the project(%s).

1) Read processing of PM project file %s has been terminated abnormally. Click the <Detail>>> button in the Error Dialog Box to check the cause of the occurrence and corrective action.
2) Read processing of MULTI build file %s has been terminated abnormally. Click the <Detail>>> button in the Error Dialog Box to check the cause of the occurrence and corrective action.

E0100: Failed to load DLL.

1) Windows may be unstable and the DLL file may be locked. Restart Windows.
2) TW may not have been correctly installed. Uninstall and then reinstall TW.

E0101: TW is exited because could not connect with other NEC tools.

1) The NEC tool (PM plus, ID850, or SM850) may not have been started correctly. Check that the NEC tool has been correctly started.
2) Windows may be unstable. Restart Windows.
3) The NEC tool (PM plus, ID850, or SM850) may not be operating correctly. Uninstall and then reinstall the NEC tool.

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 v1.3) supported by TW.
3) TW may not have been installed correctly. Uninstall and then reinstall TW.

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 v1.3) supported by TW.
3) TW may not have been installed correctly. Uninstall and then reinstall TW.
4) TW may have a bug. Consult NEC.

E0106: TW is exited, because the free memory is insufficient.

1) The memory area necessary for JavaVM to operate may have run short. Terminate unnecessary applications to secure sufficient memory space.
2) Windows may be unstable. Restart Windows.
3) The heap memory area or thread stack memory area may have run short.
   When the NEC tool chain is used, increase the size of the memory area by using the [TW Options...] menu of PM plus. When the GHS tool chain is used, increase the size of the memory area by referencing **3.2.3 Starting from command line**.

E0107: TW is exited, because the stack memory is insufficient.

1) The thread stack memory area necessary for JavaVM to operate may have run short. Terminate unnecessary applications to secure sufficient memory space.
2) 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 the one supported by TW. Check that the version %s of Java execution environment is the one supported by TW.

E0202: Could not save the TW project file(%s).

1) Access to project file %s for TW or the attribute of the project data folder for TW that stores project file %s for TW may be limited. Check project file %s for TW and the attribute of project data folder for TW that stores project file %s for TW.
2) The memory area necessary for saving project file %s for TW may run have short. Delete unnecessary files to secure sufficient memory area.

E0204: Could not delete the TW temporary file(%s).

1) Access to temporary file %s for TW or the attribute of folder that stores temporary file %s for TW may be limited. Check temporary file %s for TW and the attribute of the folder that stores temporary file %s for TW.

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) TW may not have been installed correctly. Uninstall and then reinstall TW.

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 memory area necessary for JavaVM to operate may have run short. Specify the heap memory size using a TW start option.
2) The memory area necessary for JavaVM to operate may have run short. Terminate unnecessary applications to secure sufficient memory area.
3) Windows may be unstable. Restart Windows.

E0211: The stack memory is insufficient.

1) The thread stack memory area necessary for JavaVM to operate may have run short. Specify the stack memory size using a TW start option.
2) The thread stack memory area necessary for JavaVM to operate may have run short. Terminate unnecessary applications to secure sufficient vacant 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 reserved for Windows (such as AUX, COM1, COM2, COM3, COM4, CON, LPT1, LPT2, LPT3, LPT4, PRN, and NUL).

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 PM plus.
2) Communication processing with PM plus has been terminated abnormally. Click the <Detail>>> button in the Error Dialog Box to check the cause of the occurrence and corrective action.

E0300: Failed to calculate the profile table size.

1) Size calculation processing of the profile table has been terminated abnormally. Click the <Detail>>> button in the Error Dialog Box to check the cause of the occurrence and corrective action.

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 TW may be being used. Check if the version is supported by TW.

E0303: There is no analyzable function with TW.

1) An illegal project may have been specified. TW cannot analyze the performance of or tune a project for which a C source file and assembly source file have not been specified.
2) An illegal link directive file may have been specified. TW cannot analyze the performance of or tune a link directive file for which no function is assigned to the TEXT segment or .text section.
3) 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 TW can support.

E0304: Failed to execute performance analysis.

1) An illegal project may have been specified. Click the <Detail>>> button in the Error Dialog Box to check the cause of the occurrence and corrective action.

E0305: Failed to execute tuning.

1) An illegal project may have been specified. Click the <Detail>>> button in the Error Dialog Box to check the cause of the occurrence and corrective action.

E0306: Failed to open the tuning wizard.

1) An illegal project may have been specified. Click the <Detail>>> button in the Error Dialog Box to check the cause of the occurrence and corrective action.

E0308: Failed to restore the tuned result.

1) Access to the attribute of the project data folder for TW that stores the project file for TW may be limited. Check the attribute of the project data folder for TW that stores the project file for TW.
2) he memory area necessary for restoring the tuning result may have run short. Delete unnecessary files to secure sufficient vacant memory space.

E0309: Failed to save the temporary file to %s.

1) Access to the temporary file or the attribute of folder %s that stores the temporary file may be limited. Check the temporary file and the attribute of the folder that stores the temporary file.
2) The memory area necessary for saving the temporary file may have run short. Delete unnecessary files to secure sufficient vacant memory space.

E0310: Failed to restore the temporary file(%s).

1) Access to temporary file %s or the attribute of folder that stores temporary file %s may be limited. Check temporary file %s and the attribute of the folder that stores temporary file %s.

E0311: Failed to delete the temporary file(%s).

1) Access to temporary file %s or the attribute of folder that stores temporary file %s may be limited. Check temporary file %s and the attribute of the folder that stores temporary file %s.

E0314: The running debugger(%s) was not specified in the project.

1) An undefined debugger %s may have been started by a project (e.g., SM850 if ID850 is specified by the project). TW 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 in the Error Dialog Box to check the cause of the occurrence and corrective action.
2) A C compiler package of a version not supported by TW may be being used. Check that the version is supported by TW.

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 in the Error Dialog Box to check the cause of the occurrence and corrective action.
2) The memory area necessary for inserting a probe code or saving a profile table may have run short. Change the specified link directive file to secure sufficient vacant memory space.
3) A C compiler package of a version not supported by TW may be being used. Check that the version is supported by TW.

E0402: Failed to build the tuned execution module.

1) Generation processing of a tuned execution module has been terminated abnormally. Click the <Detail>>> button in the Error Dialog Box to check the cause of the occurrence and corrective action.
2) The memory area necessary for relocating a function to be tuned may have run short. Change the specified link directive file to secure sufficient vacant memory space.
3) A C compiler package of a version not supported by TW may be being used. Check that the version is supported by TW.

E0403: Failed to generate an assembly source file.

1) Generation processing of an assembly source file has been terminated abnormally. Click the <Detail>>> button in the Error Dialog Box to check the cause of the occurrence and corrective action.
2) A C compiler package of a version not supported by TW may be being used. Check that the version is supported by TW.

E0404: Failed to analyze callpair information.

1) Analysis processing of callpair information has been terminated abnormally. Click the <Detail>>> button in the Error Dialog Box to check the cause of the occurrence and corrective action.

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 in the Error Dialog Box to check the cause of the occurrence and corrective action.

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 in the Error Dialog Box to check the cause of the occurrence and corrective action.
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 supported by TW.

E0407: Failed to execute the tuned execution module.

1) Execution processing of a tuned execution module has been terminated abnormally. Click the <Detail>>> button in the Error Dialog Box to check the cause of the occurrence and corrective action.
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 supported by TW.

E0408: Failed to analyze function information.

1) Analyzing function information has been terminated abnormally. Click the <Detail>>> button in the Error Dialog Box to check the cause of the occurrence and corrective action.

E0409: Failed to tune an execution module.

1) Tuning of an execution module has been terminated abnormally. Click the <Detail>>> button in the Error Dialog Box to check the cause of the occurrence and corrective action.

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 TW.
2) The version of system library information file %s may not be supported by TW. Check that the version of system library information file %s is supported by TW.

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 TW.

E0503: Syntax error or arithmetic error.(%s)

1) The assembly source file may describe an illegal syntax/evaluation expression. Change relevant location %s of the assembly source file to a syntax/evaluation expression supported by TW.

E0504: Too many pseudo instructions to terminate conditional assemble.(%s)

1) The assembly source file may not have a conditional assemble start quasi directive that corresponds to a conditional assemble end quasi directive. Add a conditional assemble start quasi directive to relevant location %s of the assembly source file, or delete the description at the relevant location.

E0505: The system reservation macro name(%s) is used.

1) Macro name %s may match a macro name reserved for the system. Change macro name %s from a system-reserved macro name reserved for TW.

E0506: Could not evaluate because the unsolved symbol is in the expression.(%s)

1) Static analysis processing of the symbol information has been terminated abnormally. Define the relevant symbol included in the evaluation expression of assembly source file %s.

E0507: The execution module is not read.

1) Access to the execution module or the attribute of the folder storing the execution module may be limited. Check the execution module or the attribute of 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 have run short. Increase the size of the segment or section to which the profile memory area is assigned by using the link directive file.
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 TW has been manipulated (moved, deleted, or attribute changed). Restore the file generated by TW to the original status by re-executing performance analysis/tuning processing.

E0509: The symbol(%s) is not found in the execution module.

1) Detection of symbol %s has been terminated abnormally because the execution module generated by TW has been manipulated (moved, deleted, or attribute changed). Restore the module generated by TW to the original status by re-executing performance analysis/tuning processing.

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 label of assembly source file %s.

1) E0511: There are too many end quasi directives or repeat end quasi directives of macro definition (%s).

E0511: Too many pseudo instructions to terminate macro definition or repetition.(%s)

1) A macro definition/repeat start quasi directive corresponding to a macro definition/repeat end quasi directive may not have been described in the assembly source file. Add a corresponding macro definition/repeat start quasi directive at relevant location %s of the assembly source file, or delete the description at the relevant location.

E0512: The number of macro nesting exceeded limited level(%s).(%s)

1) The number of times of nest expansion of the macro may exceed the limit of %s. Change the number of times of nest expansion of the macro in assembly source file %s to the limit of %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.

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 tuned execution module or the attribute of the folder storing the tuned execution module may be limited. Check the tuned execution module or the attribute of 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 tuned execution module or the attribute of the folder storing the tuned execution module may be limited. Check the tuned execution module or the attribute of the folder storing the tuned execution module in accordance with the contents of displayed message %s.

E0704: Failed to get the version of MULTI.

1) Version information of MULTI could not be detected. Check that "search path to MULTI" has been set as an environmental variable of the host machine in an environmental variable setting file (such as autoexec.bat).
2) Version information detection command gversion.exe supplied by MULTI may have been deleted. Uninstall and then reinstall MULTI.

E0802: Found the option(%s) which is not supported.

1) Option %s not supported by TW may have been specified for a project. 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 the folder %s storing the project file may be limited. Check the attribute of the folder %s storing the project file.

E0804: Failed to read the TW project file(%s).

1) Access to project file for TW %s or the attribute of the folder storing project file for TW %s may be limited. Check project file for TW %s or the attribute of the folder storing project file for TW %s.
2) The contents of project file for TW %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 project file for TW %s or the attribute of the folder storing project file for TW %s may be limited. Check project file for TW %s or the attribute of the folder storing project file for TW %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 TW may be limited. Check the attribute of the folder storing the project data folder for TW.
2) The existing folder name may match project data folder name for TW %s. Change the existing folder name to one that differs from project data folder for TW %s.

E0807: Failed to delete the TW temporary file(%s).

1) Access to temporary file for TW %s or the attribute of the folder storing temporary file for TW %s may be limited. Check temporary file for TW %s or the attribute of the folder storing temporary file for TW %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.

E0809: Illegal project type.

1) An illegal project may have been specified. TW cannot analyze the performance of or tune a file of a project type other than Program.

E0810: A link directive file is not specified.

1) An illegal project file may have been specified. TW cannot analyze the performance of or tune a project file without a specified link directive file.

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 TW is set using the [Debugger Settings...] menu of PM plus.
2) The execution engine (such as the debugger or simulator) could not be detected. Check that "remote target debug server remote" is correctly set in resource file *.rc supplied by MULTI.

E0812: Environment variable(%s) is not set.

1) Environmental variable %s may not have been described. Check that %s is set as an environmental variable of the host machine in an environmental variable setting file (such as autoexec.bat).

E0813: Failed to find MULTI resource file(%s).

1) Resource file %s could not be detected. Check that resource file %s is in the folder storing the project file.
2) Access to resource file %s or the attribute of the folder storing resource file %s may be limited. Check resource file %s or the attribute of the folder storing resource file %s.

E0815: Not exist the remote command in the MULTI resource file(%s).

1) The remote command could not be detected. Check that "remote target debug server remote" is correctly set in resource file %s.

E0816: The active project is not supported by TW.
%s

1) An illegal project (active project) may have been specified. TW cannot analyze the performance of or tune a project for which no C source file or assembly source file is specified.
2) The active project may be of a type not supported by TW. Check that the type of the active project is supported by TW.

E0817: There is no information(%s) in the project file(%s).

1) TW cannot detect information necessary for it to execute performance analysis/tuning processing from project file %s. Set information %s necessary for TW to execute performance analysis/tuning processing by using PM plus.

E0818: There is no project information about debugger.

1) Information concerning 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 PM plus.

E0820: Source file is not registered.

1) The information (source file name) necessary for TW to execute performance analysis/tuning processing could not be detected from the project file. Set the source file name necessary for TW to execute performance analysis/ tuning processing by using PM plus.

E0821: Debug target file is not specified.

1) The information (debug target file name) necessary for TW to execute performance analysis/tuning processing could not be detected from the project file. Set the debug target file name necessary for TW to execute performance analysis/tuning processing by using PM plus, and then execute processing to save a work space.

E0822: Unsupported output module format is specified.

1) If high-level option Output dual debug formats" is specified and if "Default", "Plain", or "MULTI" is specified as "Debugging level", TW cannot analyze the performance of or tune a project. Specify options other than the above.

E0900: The execution module(%s) is unsupported format.

1) Execution module %s may have been specified in an illegal format. Check that execution module %s has been generated by a C compiler package supported by TW.

E0901: Could not convert from the line number(%s) to the address.

1) Illegal line number %s may have been specified. TW cannot analyze the performance of or tune a line number that does not exist or a line number outside a function, which cannot be converted into an address.

E0902: Could not convert from the address information(%s) to the address.

1) Illegal symbol %s may have been specified. TW cannot analyze the performance of or tune a symbol that does not exist.

E1001: Illegal device file(%s).

1) Device file %s may have been specified in an illegal format. Check that the device file can correctly generate an execution module.
2) The device file may not have been correctly installed. Uninstall and then reinstall the device file.

E1100: Not exist string constant sign(%s) which makes a pair.

1) An end symbol condition to declare a character string constant may not be described in the assembly source file. Add the corresponding end signal at relevant location %s of the assembly source file, or delete the description at the relevant location.

E1101: Illegal operand which is not convertible to a string.

1) An operand that declares a character string constant may describe an illegal character string constant in the assembly source file. Change the relevant location of the assembly source file to a "declaration of a character string constant" that the C compiler package supported by TW can support.

E1102: Could not to evaluate the expression.

1) An illegal expression may have been described in the assembly source file. Change the relevant location of the assembly source file to an evaluation expression that the C compiler package supported by TW can support.

E1103: The macro definition does not have the declaration of a macro name.

1) A macro name may not have been described in a macro definition in the assembly source file. Change the macro name at the relevant location of the assembly source file to a macro name that the C compiler package supported by TW 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 is different from the number of arguments when the macro is used in the assembly source file. Correct the assembly source file so that the number of arguments is the same.

E1105: The specification of the file symbol is illegal.

1) An illegal file symbol declaration may have been described in the assembly source file. Check that the file symbol declaration in the assembly source file is described correctly.

E1200: Not exist bracket which makes a pair.

1) Parentheses in the evaluation expression may not match in the assembly source file. Make sure that the parentheses included in the evaluation expression of the assembly source file match.

E1201: Not exist an expression to evaluate.

1) An illegal evaluation expression may have been described in the assembly source file. Change the evaluation expression of the assembly source file to one that the C compiler package supported by TW can support.

E1202: Could not evaluate the numerical expression(%s).

1) An evaluation expression including illegal numeric expression %s may have been described in the assembly source file. Change the numeric expression of the assembly source file to one that the C compiler package supported by TW can support.

E1203: Could not evaluate the symbol(%s).

1) An evaluation expression containing illegal symbol %s may have been described in the assembly source file. TW cannot evaluate a symbol whose value is not defined.
2) An evaluation expression containing label %s may have been described in the assembly source file. TW 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) An evaluation expression containing an operator with an illegal number of items may have been described in the assembly source file. Change the evaluation expression in the assembly source file to the one that the C compiler package supported by TW can support.

E1205: Illegal usage of the operator(%s).

1) An evaluation expression containing an illegal operator may have been described in the assembly source file. Change the evaluation expression in the assembly source file to the one that the C compiler package supported by TW can support.

E1206: Divided by zero in the expression.

1) An illegal evaluation expression (division by zero) may have been described in the assembly source file. Change the evaluation expression in the assembly source file to the one that the C compiler package supported by TW 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 the execution module for original analysis is correctly generated.
2) Generation processing of an execution module for profile 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 the execution module for profile is correctly generated.
3) Generation processing of a tuned execution module 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 the tuned execution module is correctly generated.
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 temporary file or the attribute of the folder storing the temporary file may be limited. Check the temporary file or the attribute of the folder storing the temporary file, in accordance with message displayed %s.
2) The temporary file may have been referenced from another application (such as the editor). Check whether the temporary file was referenced by another application, in accordance with message displayed %s.

E1303: Found the unsupported symbol in the execution module.

1) An illegal execution module may have been specified. TW cannot analyze the performance of or tune an execution module that includes a monitor program for system performance analyzer AZ850.

E1400: Failed to execute the execution module(%s).

1) Execution processing of the execution module has been terminated abnormally. Click the <Detail>>> button in the Error Dialog Box to check the cause of the occurrence and corrective action. Also refer to **4.8.3 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 correctly performed.
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 in the Error Dialog Box to check the cause of the occurrence and corrective action.

E1403: Failed to analyze the profile information.

1) Profile information could not be detected. Check the file and the attribute of the project data folder for TW storing the file.
2) Analysis processing of the profile information has been terminated abnormally. Check that "remote target debug server remote" has been correctly set in resource file *.rc supplied by MULTI.
3) TW may have a bug. Consult NEC.

E1404: Failed to analyze the trace data.

1) Trace data could not be detected. Click the <Detail>>> button in the Error Dialog Box to check the cause of the occurrence and corrective action.
2) Analysis processing of trace data has been terminated abnormally. Check that "remote target debug server remote" has been correctly set in resource file *.rc supplied by MULTI.

E1405: Not exist data files for trace analysis.

1) Access to the attribute of the project data folder for TW storing the data file for trace data analysis may be limited. Check the attribute of the project data folder for TW storing the data file for trace data analysis.
2) Generation processing of the data file for trace data analysis has been terminated abnormally.
Change the number of loops of the wait processing specified by the TW start option -Dtw.MultiShellWaitCount to an appropriate value.

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 the execution module for original analysis has been correctly generated.
2) Access to link directive file %s or the attribute of the folder storing the link directive file may be limited. Check link directive file %s or the attribute of the folder storing the link directive file.
3) The link directive file may have been described in a format not supported by TW. Check that the version of the C compiler package is supported by TW. 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 TW storing the link directive file may be limited. Check the attribute of the project data folder for TW storing 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.

E1602: Failed to download the execution module(%s) to the target.

1) Downloading processing of the execution module has been terminated abnormally. Check that the mapping size of the execution module has been correctly specified.
2) Access to execution module %s or the attribute of the folder storing execution module %s may be limited. Check execution module %s or the attribute of 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 may have been specified in an illegal format. Check that execution module %s has been generated by a C compiler package supported by TW.
5) The version of another tool may not be supported by TW. Check that the version of the other tool is supported by TW.
6) Another 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 another tool may not be supported by TW. Check that the version of the other tool is supported by TW.
2) Another 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 another tool may be not supported by TW. Check that the version of the other tool is supported by TW.
2) Another 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 another tool may not be supported by TW. Check that the version of the other tool is supported by TW.
3) Another 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. heck 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 another tool may not be supported by TW. Check that the version of the other tool is supported by TW.
3) Another 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 another tool may not be supported by TW. heck that the version of the other tool is supported by TW.
3) Another 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 another tool may not be supported by TW. heck that the version of the other tool is supported by TW.
2) Another 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 project file %s or the attribute of the folder storing project file %s may be limited. Check project file %s or the attribute of the folder storing project file %s.
2) The version of another tool may not be supported by TW. heck that the version of the other tool is supported by TW.
3) Another 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 another tool may not be supported by TW. heck that the version of the other tool is supported by TW.
2) Another 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) TW may not be correctly installed. Uninstall and then reinstall TW.

E1637: Failed to get the list of registry value.(%s)

1) Windows may be unstable. Restart Windows.
2) TW or another tool may not be installed correctly. Uninstall and then reinstall TW or the other tool.

E1638: Failed to get the string of the registry value.(%s)

1) Windows may be unstable. Restart Windows.
2) TW or another tool may not be installed correctly. Uninstall and then reinstall TW or the other tool.

E1639: Failed to get the numerical value of the registry value.(%s)

1) Windows may be unstable. Restart Windows.
2) TW or another tool may not be installed correctly. Uninstall and then reinstall TW 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 action.

E1642: The wrapper command is not installed normally.

1) The wrapper command supplied by TW may not have been installed correctly. Execute setup command TWWSetup.exe and install the wrapper command.

E1646: Could not start the debugger within the specific time(%s sec).

1) Starting processing of the debugger has been terminated abnormally. Change the timeout value %s specified by the TW start option -Dtw.DebuggerTimeout to an appropriate value.
2) The debugger may not be operating correctly. Individually start the debugger and check that the debugger operates normally.
3) 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 relevant external command %s was correctly started when it was started from command prompt.
2) The external command could not be detected. Check that "search path to 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 have run short. Terminate unnecessary applications to secure sufficient vacant memory area.

E1801: The device information(%s) is not set in registry.

1) The device file may not be correctly installed. Uninstall and then reinstall the device file.

E1802: The device information is not set in environment(%s).

1) Device information could not be detected. Check that "search path to device information" has been set as an environmental variable of the host machine in an environmental variable setting file (such as autoexec.bat).

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) Device file %s could not be detected. Check that DEVICE_FILE and IEPATH have been set as environmental variables of the host machine in an environmental variable setting file (such as autoexec.bat).
3) Access to device file %s or the attribute of the folder storing device file %s may be limited. Check device file %s or the attribute of the folder storing device file %s.

E1900: Can not connect to ToolLinkManager.

1) TW may be unstable. Restart TW.
2) Windows may be unstable. Restart Windows.
3) ToolLinkManager may not be correctly installed. Uninstall and then reinstall ToolLinkManager.

E1901: ToolLinkManager is not running.

1) ToolLinkManager may not be operating correctly. Terminate the NEC tool (PM plus, ID850, SM850, or TW), and restart ToolLinkManager.

E9000: TCould not start TW.

1) The Java execution environment may not be correctly installed. Install the Java execution environment supported by TW (Java2 Runtime Environment Standard Edition v1.3).

2) The Java execution environment could not be detected. Check that "search path to Java execution environment" is 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 project file or the attribute of the folder storing the project file may be limited. Check the project file or the attribute of the folder storing the project file.
3) An option not supported by TW may have been specified for the project. Delete option %s from the project.

E9002: The fatal error occurred. Could not start TW.

1) The memory area necessary for JavaVM to operate may have run short. Terminate unnecessary applications to secure sufficient vacant 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 be correctly installed. nstall the Java execution environment supported by TW (Java2 Runtime Environment Standard Edition v1.3).
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.

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 file %s or the attribute of the folder storing file %s may be limited. Check file %s or the attribute of the folder storing file %s.
3) The memory area necessary for copying the file may have run short. Delete unnecessary files to secure sufficient vacant memory area.

E9101: Failed to check the wrapper command(%s).

1) Access to the attribute of wrapper command AS850.EXE for the assembler supplied by TW may be limited. Check the attribute of wrapper command AS850.EXE for the assembler supplied by TW.
2) Access to wrapper command %s or the attribute of the folder storing wrapper command %s may be limited. Check wrapper command %s or the attribute of the folder storing wrapper command %s.
3) The memory area necessary for wrapper command % to operate may have run short. Terminate unnecessary applications to secure sufficient vacant 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 TW.
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 TW or the contents of the registry information may have been destroyed. Uninstall and then reinstall TW.

INDEX

S

T

V

W

[MEMO]

NEC

Facsimile Message

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

From:

Name

Company

Tel. FAX

Address

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: +1-800-729-9288
+1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Technical Documentation Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-6462-6829

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Korea
NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: +82-2-528-4411

Taiwan
NEC Electronics Taiwan Ltd.
Fax: +886-2-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: +81- 44-435-9608

I would like to report the following error/make the following suggestion:

Document title: _____

Document number: _____ Page number: _____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	❑	❑	❑	❑
Technical Accuracy	❑	❑	❑	❑
Organization	❑	❑	❑	❑