FR/F²MC FAMILY

SOFTUNE C CHECKER MANUAL

for V3

FUJITSU

FR/F²MC FAMILY

S<small>OFTUNE</small> C CHECKER MANUAL

for V3

FUJITSU LIMITED

PREFACE

■ **Objectives**

Thank you for purchasing the C checker, Softune C Checker.

The C checker, Softune C Checker (referred to as the C Checker in the remainder of this manual), is a built-in development support tool that accepts source programs written in C as input and points out coding errors as well as lines where performance can be improved.

The Softune C Checker runs on Windows XP, Windows Me, Windows 2000, Windows 98 and Windows NT 4.0.

This manual is written for those who wish to use this system to check C source programs.

This system can check those C source programs that conform to the "American National Standard for Information System Programming Language C, X3.159-1989" (referred to as the ANSI standard in the remainder of this manual).

Readers of this manual require familiarity with the basic operation of Windows XP, Windows Me, Windows 2000, Windows 98 or Windows NT 4.0 as well as the basic knowledge of the C language specifications.

For an explanation of the C language specifications, see the "Programming Language C JIS X 3010-1993" or any other book on C conforming to the ANSI standard.

■ **Trademarks**

SOFTUNE is a trademark of FUJITSU LIMITED.

Microsoft, Windows, and Windows NT are registered trademarks of Microsoft Corporation in the U.S. and other countries.

The names of products and systems appearing in this manual are trademarks or registered trademarks of their respective companies.

■ **Configuration of this manual**

To get an overview of this system, read Chapter 1, "Overview."

To learn how to use this system to check a source program, read Chapter 2, "Checking Procedures."

To learn how to set the check conditions for the source program to be checked, read Chapter 3, "Setting up Check Options."

To learn how to handle the source program file to be checked, read Chapter 4, "File," Chapter 5, "Edit," and Chapter 6, "Search."

To learn how to display and manipulate windows, read Chapter 7, "View" and Chapter 12, "Window."

To learn how to manage source program projects, read Chapter 8, "Project."

To learn how to display and manipulate diagnostic messages, read Chapter 9, "Message."

To learn how to compile source programs, read Chapter 10, "Compile."

To learn how to set the compiler environment, read Chapter 11, "Setup."

To learn how to manipulate help files, read Chapter 13, "Help."

This manual consists of the following twelve chapters:

Chapter 1 Overview

This chapter provides an overview of using the C Checker to check C source programs.

Chapter 2 Checking Procedures

This chapter explains the procedures required to use the C Checker to check a C source program.

Chapter 3 Setting up Check Options

This chapter explains how to set up check conditions.

Chapter 4 File

This chapter explains how to manipulate files.

Chapter 5 Edit

This chapter explains how to use the Edit menu.

Chapter 6 Search

This chapter explains how to use the Search menu.

Chapter 7 View

This chapter explains the View menu.

Chapter 8 Project

This chapter explains how to create and manipulate projects.

Chapter 9 Message

This chapter explains how to display and manipulate diagnostic messages.

Chapter 10 Compile

This chapter explains how to compile programs.

Chapter 11 Setup

This chapter explains the setup operation.

Chapter 12 Window

This chapter explains how to display and manipulate windows.

Chapter 13 Help

This chapter explains how to display and manipulate help files.

Chapter 14 Sorting of the Diagnostic Message

This chapter explains how to sort and manipulate diagnostic message.

■ Related manual

When using this system, refer to the following manual as required:

- "FR/F²MC Family Softune C Compiler Manual"

Reading This Manual

■ **Product name abbreviation**

In this manual and this product, product names are designated as follows:

SOFTUNE is designated as Softune.

F^{2}MC is designated as FFMC.

The Microsoft® Windows® 98 operating system is abbreviated to Windows 98.

The Microsoft® Windows® Millennium Edition operating system is abbreviated to Windows Me.

The Microsoft® Windows NT® Workstation operating system Version 4.0 and
the Microsoft® Windows NT® Server network operating system Version 4.0 are abbreviated to
Windows NT 4.0.

The Microsoft® Windows® 2000 Professional operating system,
the Microsoft® Windows® 2000 Server operating system,
the Microsoft® Windows® 2000 Advanced Server operating system and
the Microsoft® Windows® 2000 Datacenter Server operating system are abbreviated to
Windows 2000.

The Microsoft® Windows® XP Professional operating system and
the Microsoft® Windows® XP Home Edition operating system are abbreviated to Windows XP.

■ **Note on descriptions**

The figures and screen examples presented in this manual are those that appear when
Windows 98 is used. Depending on the OS, environment, machine type, and resolution, your
screen may look slightly different.

CONTENTS

viii

CHAPTER 1   Overview

This chapter provides an overview of using the C Checker to check C source programs.

1.1 Softune C Checker

The Softune C Checker is a tool that accepts source programs written in C as input to point out the lines where quality and performance can be improved and to display and to print diagnostic messages useful in software development. This tool offers information effective for improving the efficiency of program development and object code. This tool supports the built-in functions of C compilers for FUJITSU microcontrollers.

■ **Effects of using the Softune C Checker**

The C Checker assists in meeting the following needs:

- Removes coding mistakes in early stages.
- Gives new programs a thorough review in a short time.
- Writes high-quality code (even for novices in C).
- Writes high-performance code.
- Allows non-FUJITSU CPU software resources to be used with FUJITSU CPUs.
- Understands the features of FUJITSU C compilers

2

1.2   Function Outlines

The functions provided by the C Checker are largely divided into the following groups. This section briefly explains these groups, in the indicated order.

- **Check**
- **Set check conditions**
- **File**
- **Edit**
- **Search**
- **View**
- **Project management**
- **Diagnostic message**
- **Compile**
- **Setup**
- **Window**
- **Help**

■ **Check**

   Contains functions for pointing out the lines with a problem found in a source program.

■ **Set checking conditions**

   Contains functions for setting check conditions according to the purpose.

■ **File**

   Contains functions for creating, displaying, saving, and printing files.

■ **Edit**

   Contains functions assisting in editing a source program to reflect diagnostic results.

■ **Search**

   Contains functions for searching for a character string through the file being edited.

■ **View**

   Contains functions used to display a variety of windows.

■ **Project management**

   Contains functions for managing the source programs to be checked.

■ **Diagnostic message**

   Contains functions used to display the messages for the lines with a problem found.

■ **Compile**

   Contains functions for compiling a source program in which diagnostic results have been reflected.

3

■ **Setup**

> Contains functions for customizing the internal editor and for setting the environment of the compiler which has already been installed.

■ **Window**

> Contains functions for changing the state of a window.

■ **Help**

> Contains functions for displaying, searching for, and printing function outlines and operating instructions.

1.2.1   Check

The check functions are for outputting messages calling the user's attention to the problem lines in a source program. It contains the functions for selecting the file to be checked and for setting up check options. These functions enable the user to easily view the problem lines in a program, according to the purpose.

■ **Files that can be checked**

> The C Checker can check C source files.

■ **Selecting the file to be checked**

> Select the source file to be checked by the C Checker. A source file can be selected as described below. A file can also be selected by using the corresponding buttons located in the check tool bar.

> ❍ File being edited

> A check can be performed on the active window of the file opened by the editor.

> ❍ Selected file in a project

> A check can be performed on the file selected in the project management window.

> In the project management window, select a file as described below:

> • To select a single file: Click it with the left mouse button.

> • To select multiple files: Press and hold down the Control key and click them with the left mouse button; or press and hold down the Shift key and click them with the left mouse button.

> ❍ All files in a project

> A check can be performed on all files in a project.

■ **Setting up check options**

The following conditions can be set:

❍ Check mode

- Target compiler

- Chip classification

- Target MCU

❍ Language

❍ Quality

- Portability

- Extended specification

- Possibility of coding mistake

- Maintainability

- Porting mode

❍ Performance

- Architecture independent

- Architecture dependent

❍ Other options

- Include path

- Macro definition

- Suppressing messages

1.2.2   Set check conditions

The functions in this group are for setting check conditions. Choosing the [Set up check options] command from the [Check] menu displays the [Set up check options] dialog box, in which check conditions can be selected. These functions enable the user to specify check conditions appropriate to the purpose.

■ **Check mode**

Select the target compiler and target MCU to be used by the C Checker.

Items that can be set are as follows:

- Target compiler (fcc911s, fcc907s, fcc896s, ANSI)
- Chip classification
- Target MCU

■ **Language**

The language by which the diagnostic message is displayed is shown.

The language which can be selected is only English:

■ **Check classification**

Select the desired check items. The following items can be selected:

- Quality (portability, extended specification, possibility of coding mistake, maintainability, and conversion mode)
- Performance (architecture independent, architecture dependent)

■ **Check option information file**

The options set up in the [Set up check options] dialog box can be saved in a file by using the [Save information file] button.

An existing check option information file can be opened by using the [Open information file] button.

This is very useful because it enables multiple check option patterns to be stored.

The file identifier of a check option information file is ".cif."

7

■ **Include paths**

Set up the include path of the header file used with the source file to be checked by the C Checker.

To register an include path, enter the include path name directly in the column. Alternatively, click the [Browse] button to display the folder browse dialog box and enter the include path.

Click the [Add] button to register the entered include path.

Registered include paths are used, starting from the uppermost one, to search for the files to be included.

It is possible to change the order in which paths are registered and to delete registered paths.

■ **Macro definitions**

Set up the macro definitions used in the source file to be checked by the C Checker. To register a macro definition, enter the macro name directly in the column. Click the [Add] button to register the entered macro name. Registered macros are defined starting from the uppermost one. It is possible to change the order in which the macro names are registered and to delete registered names. If macro definitions with the same name are registered, the macro definition registered last takes precedence.

■ **Suppressing messages**

Set the numbers of the messages that the C Checker is to suppress.

To suppress the message number, the check box of the number which wants to be suppressed id checked. If the icon + in front of the check box is clicked, the suppress of a more detailed message becomes possible.

8

1.2.3   Files

The functions in this group are for manipulating source program files. These functions enable the user to save and print corrected source program files.

■ **New**

Creates a new file.

A file with no title is opened on the editor.

■ **Open**

Opens an existing file.

Choosing the [Open] command from the [File] menu displays the dialog box for opening a file. From this dialog box, select the file to be opened and click the [Open] button, and the file is opened.

Either the "*.c" or "*.*" file type can be selected so that files of that file type can be displayed.

By default, files with the ".c" file identifier are displayed.

■ **Close**

Closes an open file.

If the open file has been modified, a confirmation message dialog box appears asking whether the contents of the file should be saved.

■ **Overwrite**

Saves an open file, overwriting its previous contents. If the open file is a newly created one, the Save as dialog box appears.

■ **Save as**

Saves an open file with a name.

In the Save as dialog box, the locations in which the file can be saved are displayed as well as the directories and files saved in them.

Either the "*.c" or "*.*" file type can be selected so that files of that file type can be displayed.

By default, files with the ".c" file identifier are displayed.

■ **Print**

Prints an open file.

Selecting the [Print] command from the [File] menu displays the [Print] dialog box.

In the dialog box, specify the printer to be used, printing range, number of copies, and other detailed printing information. Then click the [OK] button, to begin printing.

■ **Print preview**

Displays what is to be printed in a window.

Printing can be started by selecting the [Print] dialog box from this window.

■ **Recent File**

Stores the names of up to five previously opened files.

If more than five files are opened, the excess names are removed from the list, beginning with the oldest one.

■ **Exit SOFTUNE C checker**

Exits from the C Checker.

If any open file has not been saved, a conformation message dialog box appears asking whether the contents of the file should be saved.

1.2.4 Edit

The functions in this group are for editing source programs. The cutting and pasting operations can be performed on the file being edited. These functions enable the user to efficiently correct the lines in the source program that require attention as indicated by the check function.

■ **Undo**

Undoes the immediately preceding editing operation.

■ **Redo**

Redoes the immediately preceding undo operation.

■ **Cut**

Cuts a selected portion and copies it to the clipboard.

■ **Copy**

Copies a selected portion to the clipboard.

■ **Paste**

Pastes the contents of the clipboard to the cursor position.

■ **Delete**

Deletes a selected portion.

■ **Select all**

Selects all files.

11

1.2.5 Search

The functions in this group are for searching for a character string through the source program opened on the editor. They include the functions for specifying search conditions, setting tags during search, and jumping to tags. These functions enable the user to efficiently search for the desired character string on the editor.

■ **Search**

Searches for a character string through the file being edited.

If Search is chosen, the search dialog box appears.

■ **Replace**

Replaces a character string in the file being edited.

If Replace is chosen, the replace dialog box appears.

■ **Find in Files**

Search a string on some file in the target directory.

For a search from a file, the [Find in Files] dialog box is displayed.

■ **Tag**

The following tag operations can be performed in the file being edited:

❍ Jump Tag Next (Down)

Jumps to the next tag.

❍ Jump Tag Next (Up)

Jumps to the previous tag.

❍ Add Tag (or Delete Tag)

Adds tags to or deletes tags from the line on which the cursor is positioned.

❍ Delete Tag All

Deletes all tags.

❍ List Tag

Displays a list of tags.

Note: To delete diagnostic message jump tags, use [Delete all tags].

■ **Jump**

Allows the user to select the line to jump to, from among a specified line, last line, and first line.

1.2.6   View

The view functions are for turning tool bars and windows on or off and for customizing a variety of windows and the editor. These functions enable the user to change screen settings freely.

■ **Tool bars**

Turns the following tool bars on or off.

- Standard tool bar
- Check tool bar
- Window state changing tool bar
- Find tool bar

■ **Status bar**

Turns the status bar on or off.

■ **Window**

Turns the following windows on or off:

- Project management window
- Diagnostic message window
- Check log display window

■ **Customize**

Allows customization of font type and size for the following windows. It also provides a function which allows all windows to be reset to their standard settings.

- Project management window
- Diagnostic message window
- Check log display window
- Standard built-in editor

In addition, the editor can be customized as follows:

- Show the New Line
- Show the EOF
- Show the Ruler
- Show the Line Number
- Automatic Indent
- Show double-byte blank
- Show the Tab and Number of Tabs
- Highlighting the keywords

13

1.2.7   Project management

The project management functions are for managing the source program to be checked. They include the following functions:
- **Create a new project**
- **Use an existing project**
- **Create a project from Softune Workbench project file**

These functions enable the user to manage source programs for all projects.

■ **Project files**

> The file identifier for project files is ".cpj".

■ **Create a project**

> Creates a new project or Creates a project from Softune Workbench project file.

> In the dialog box, specify the desired project name and project file folder name.

> The folder browse dialog box can be displayed to specify the desired project file folder.

> The dialog box for allocating networked drives can also be displayed.

> In the dialog box, the name of the drive to which a networked drive is to be allocated can be specified and an actual path can be selected.

> In addition, reconnection at logon can be set.

> In a project file folder, the "Summary" folder, the "Output" folder and the option files (extension is .opt) are created. The "Summary" folder will contain summary information for each file. The "Output" folder will contain output files that make by C compiler. The option files are used by the project.

■ **Select a project**

> To select an existing project, choose the [Open] command from the [Project] menu. The [Open] dialog box appears. From this box, select the desired project file.

■ **Save a project**

> Saves the contents of a project.

> If the project contains a file that has been edited, a confirmation message dialog box appears asking whether the contents of the file should be saved.

> The project can be renamed before being saved.

■ **Load a file into a project**

> Loads a file into an open project to register the file.

> To load a file, specify the file name in the dialog box. The file names with identifiers "." and ".h" can be specified.
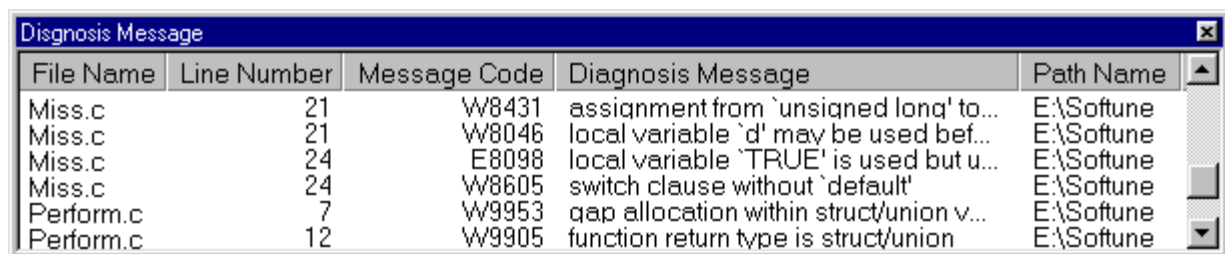
14

1.2.8   Diagnostic message

The diagnostic message functions are used to display information about the lines that have been checked. Display information includes file names, line numbers with a problem found, message numbers, simple messages, and file path names. Double-clicking the file name at the beginning of a diagnostic message or another item causes a jump to the line with a problem found in the source program. These functions enable the user to find problem lines in the source program quickly. By choosing Detailed message from the popup menu, which is displayed by clicking the right mouse button, a detailed message displayed and diagnostic message can be sorted.

■ **Diagnostic messages**

The check results are displayed in the diagnostic message window.

The messages displayed in the diagnostic message window are in the following format.

The diagnostic message format is as follows:

File Name	Line Number	Message Code	Diagnosis Message	Path Name
Miss.c	21	W8431	assignment from `unsigned long' to...	E:\Softune
Miss.c	21	W8046	local variable `d' may be used bef...	E:\Softune
Miss.c	24	E8098	local variable `TRUE' is used but u...	E:\Softune
Miss.c	24	W8605	switch clause without `default'	E:\Softune
Perform.c	7	W9953	gap allocation within struct/union v...	E:\Softune
Perform.c	12	W9905	function return type is struct/union	E:\Softune

Figure 1.2-a The diagnostic message format

The diagnostic message functions are as follows:

❍ Open

Reads a diagnostic message file to display the diagnostic messages in the diagnostic message window.

❍ Close

Closes the displayed diagnostic message file.

❍ Save

Saves the displayed diagnostic messages into a file.

❍ Save as

Saves the displayed diagnostic messages with another name.

❍ Print

Prints the displayed diagnostic messages.

The three commands, [Open], [Save], [Save as], and [Print] are effective only when diagnostic messages are displayed.

15

■ **Format for detailed messages**

Detailed messages are displayed in help format. These messages contain the following items.

A "program example," "program correction example," and "hint" can be displayed in a popup window by clicking their respective buttons with the left mouse button.

○ Detailed explanation of what problem was found in the line

Explains what problem was found in the line and how to correct it.

○ Sample program

Shows a program example that could be output messages for a problem found.

○ Program correction plan

Proposes an example of correcting the line with a problem found in the checked program.

○ Key advice

Shows a hint etc. for coding.

○ Related items

Displays a list of similar items. Other detailed messages can be retrieved and displayed from the list.

16

1.2.9   Compile

The compile functions are for compiling a source program. The following compilers can be selected:
- **fcc911s (compiler for the FR Family)**
- **fcc907s (compiler for the F^{2}MC-16 Family)**
- **fcc896s (compiler for the F^{2}MC-8L Family)**
- **ANCI (ANSI compiler)**

These functions enable the user to translate and verify a source program in which the lines with a problem found by a check have been corrected. To use the Compile functions, either compiler must be installed.

■ **File to be compiled**

Select the file to be compiled. The following files can be selected.

A file can also be selected using the Compile button in the check tool bar.

- File being edited

- File selected in a project

- All files in a project

■ **Select the compiler**

Select the compiler used with the C Checker.

The following compilers can be selected:

- fcc911s (compiler for the FR Family)

- fcc907s (compiler for the F^{2}MC-16 Family)

- fcc896s (compiler for the F^{2}MC-8L Family)

- ANSI (ANSI compiler)

17

■ **Compile options**

This function allows the user to set up the options for the selected compiler. The following options can be set up. The options that have been set up can be verified.

❍ Target compiler

Select the target C compiler.

❍ Language

Select the language of message which the compiler outputs.

❍ Chip classification

Select the chip classification.

❍ Optimization level

Set the optimization level for the compiler.

❍ Warning level

Set the warning level for the compiler.

❍ Translation control

Select the range in which translation is to be executed.

❍ Memory model

Specify the memory model of target CPU. (fcc907s only)

❍ Target MCU

Specify the target MCU.

❍ Output the debug information

Specify whether to output the debug information.

❍ Suppress reading of default option file

Specify whether to suppress the reading of the default option file.

❍ Output the used stack information

Specify whether to output used stack information.

❍ Position the const variables on RAM

Specify whether to position the const variables on RAM. (fcc907s only)

❍ Details of options

• Preprocessor: Set up the detailed options for the preprocessor.

• Language specification: Set up the detailed options for the language specifications.

• Optimization: Set up the detailed options for optimization.

• Output object: Set up the detailed options for output.

❍ Other options

Sets up the other options specifiable for the compiler directly.

18

1.2.10 Setup

The setup functions are for setting the internal editor and an installation environment for the compilers.

■ **Editor Customize**

The Editor Customize function permits the user to specify the editor (internal editor or your own editor) to be used for editing.

■ **Compilation environment**

Set the path in which the environment is installed for the compiler used.

If the compiler is not installed, a dialog box opens to indicate the compiler is not installed.

19

1.2.11 Window

The main window contains the following:
- **Menu bar**
- **Tool bars**
- **Status bar**
- **Project management window**
- **Diagnostic message management window**
- **Check log display window**
- **Detailed message window**
- **Editor**

These components enable the user to obtain information required for a check.

■ **Menu bar**

From the menu bar, the following can be selected.

- File
- Edit
- Search
- View
- Project
- Check
- Message
- Compile
- Window
- Setup
- Help

■ **Tool bars**

The following tool bars can be displayed:

Instead of the buttons in the tool bars, the corresponding menu commands can also be used.

Positioning the mouse pointer on a button displays the tips for the tool.

- Standard tool bar
- Check tool bar
- Window state changing tool bar
- Find tool bar

■ **Status bar**

The status bar displays an explanation of a command, the keyboard status, and other information.

20

■ **Project management window**

> The project management window displays the files contained in a project and summary information for every file checked.

■ **Diagnostic message display window**

> The diagnostic message display window displays the check results.
>
> The following items are displayed.
>
> Double-clicking the name of a checked file in this window with the left mouse button displays the edit screen for the checked file.

- Name of the checked file
- Number of the lines with a problem found
- Message number
- Simple message
- File path name

■ **Check log display window**

> The check log display window displays the execution log information collected during a check.
>
> In this window, the checked files, the check status, and other information can be referenced.

■ **Detailed message window**

> The detailed message window displays the detailed information related to a diagnostic message, in help format.
>
> Display information includes an explanation of diagnostic, sample program, program correction plan, key advice, and related items.
>
> When this window is displayed, the user can correct the lines with a problem found in the source program by looking at program examples and correction examples.
>
> Technical terms are explained in a popup window.

■ **Editor**

> On the edit screen, the user can edit the source program by using the correction examples displayed in the detailed message window for reference.

■ **Operations on windows**

> The Window functions can perform the following operations on displayed windows:

- Overlap a window.
- Arrange windows horizontally.
- Arrange windows vertically.
- Arrange multiple icons on the display when individual windows are minimized.

1.2.12 Help

The help functions include the function used to display the help files containing the function outlines and operating instructions for the C Checker. They also include the functions for printing help files and searching for keywords. These functions enable the user to view the operating instructions for this tool and the information about the check items.

■ **Tip of the day**

       Displays the convenient uses of the C Checker and other information.

       Such information can also be displayed when the C Checker is started.

■ **Help topics**

       Displays the contents of the help files of the C Checker as well as keywords.

■ **About SOFTUNE C Checker**

       Displays C Checker version information.

CHAPTER 2   Checking Procedures

This chapter explains the procedures required to use the C Checker to check a C source program.

2.1 Starting the C Checker

To start the C Checker, click the C Checker icon registered in the system. (See Figure 2.1-a.)
When this tool is started, the starting window of the C Checker is displayed. (See Figure 2.1-b.)

■ **Starting the C Checker**

To start the C Checker, click the C Checker icon registered in the system. (See Figure 2.1-a.)
Alternatively, choose [Program]-[Softune C Checker]-[Softune C Checker] from the Start menu.
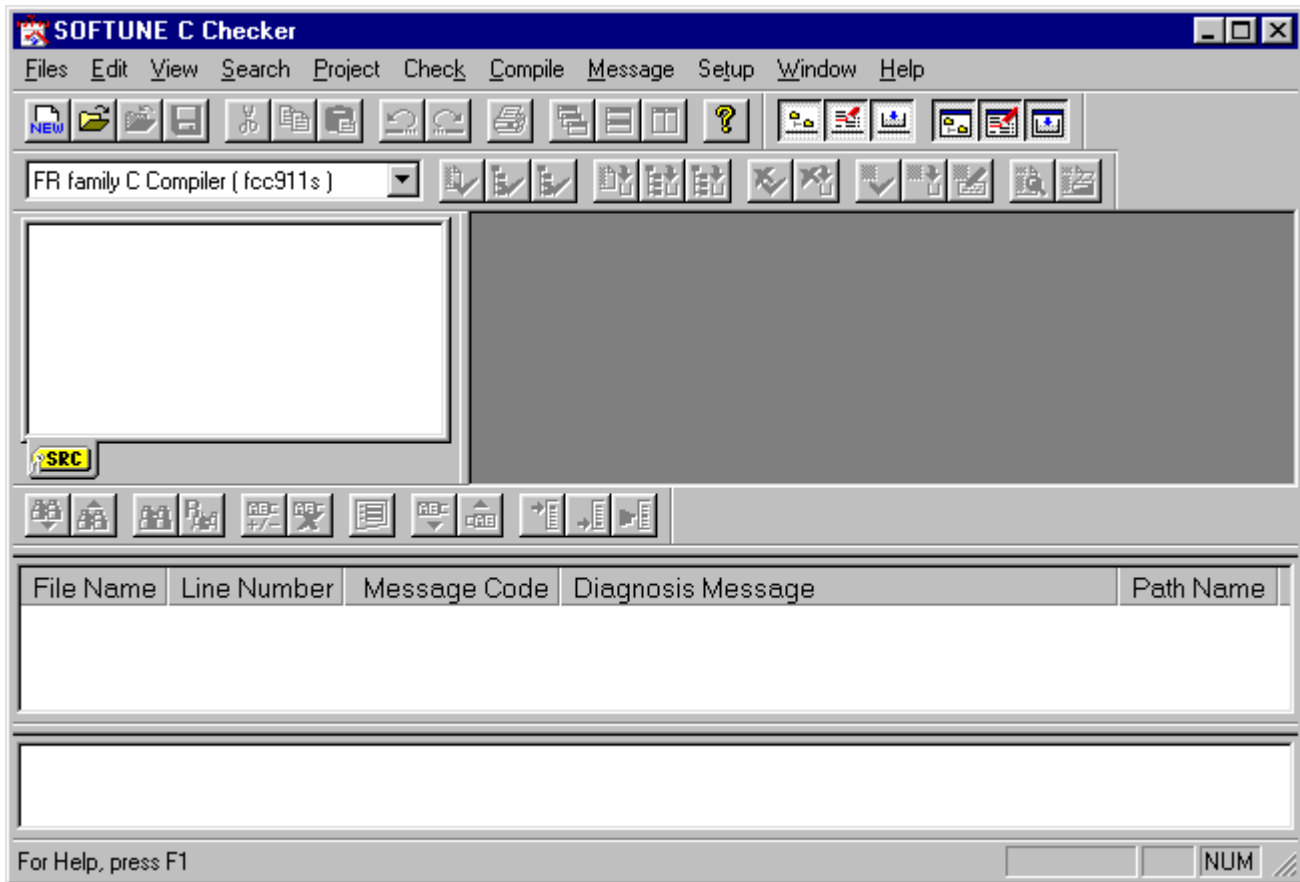
Figure 2.1-a C Checker icon

Figure 2.1-b Starting window of the C Checker

24

2.2   Selection of a Source File

To check a source program using the C Checker, select the source program file to be checked.

- **[Open] command from the [Files] menu**
- **[New] command from the [Project] menu**
- **[Open] command from the [Project] menu**

Execute one in the above-mentioned command.

■ **[Open File]**

Choose the [Open] command from the [Files] menu, and the [Open] dialog box, shown in Figure 2.2-a, appears. Select a file with the ".c" file identifier, and the file is opened on the built-in editor. (See Figure 2.2-b.)
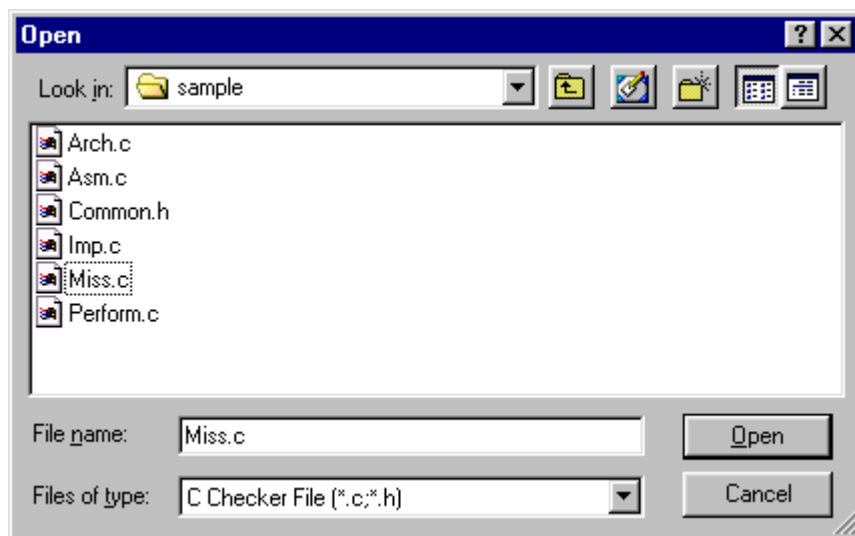

Figure 2.2-a [Open] dialog box

25
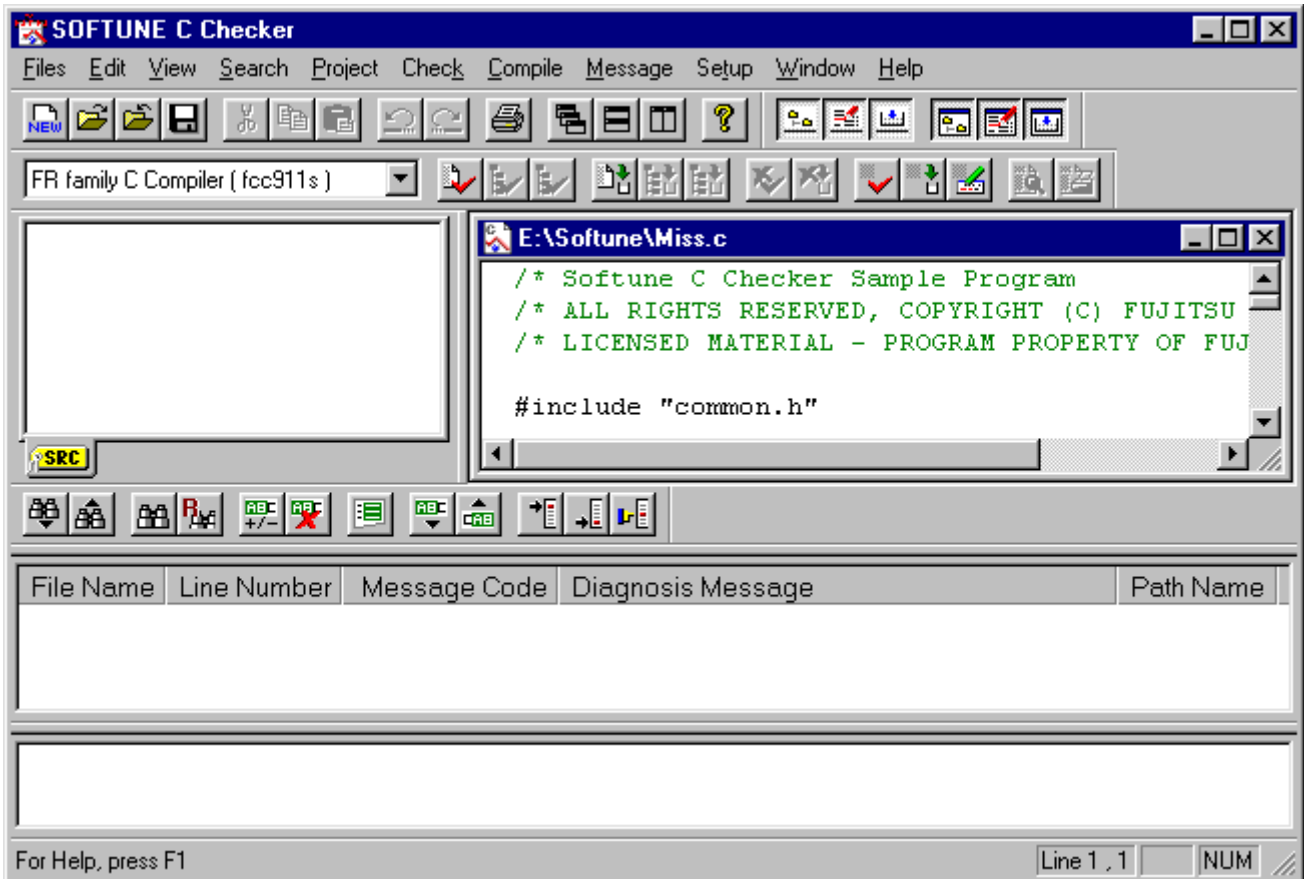
Figure 2.2-b Window of the file opened on the built-in editor

■ **[Create Project]**

Choose the [New] command from the [Project] menu, and the [Create Project] dialog box, shown in Figure 2.2-c, appears. Create a new project file.
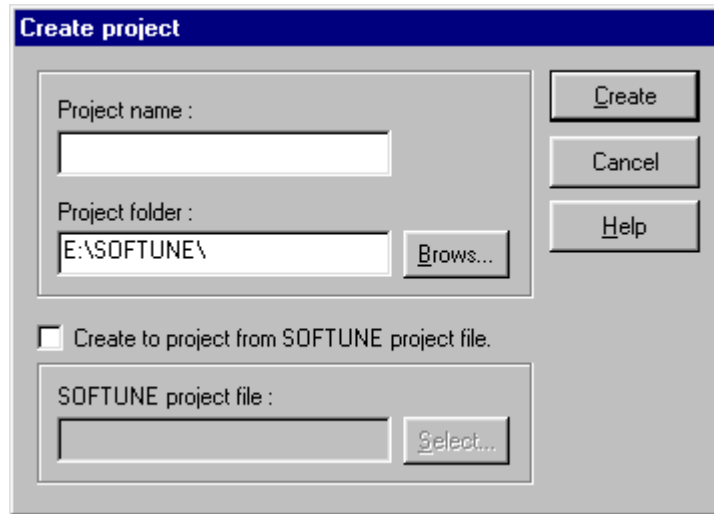
Figure 2.2-c [Create Project] dialog box

Choose the [Import File...] command from the [Project] menu, and the [Import] dialog box, shown in Figure 2.2-d, appears. Select a file with the ".c" file identifier, and the selected files are registered in the project. (See Figure 2.2-e)
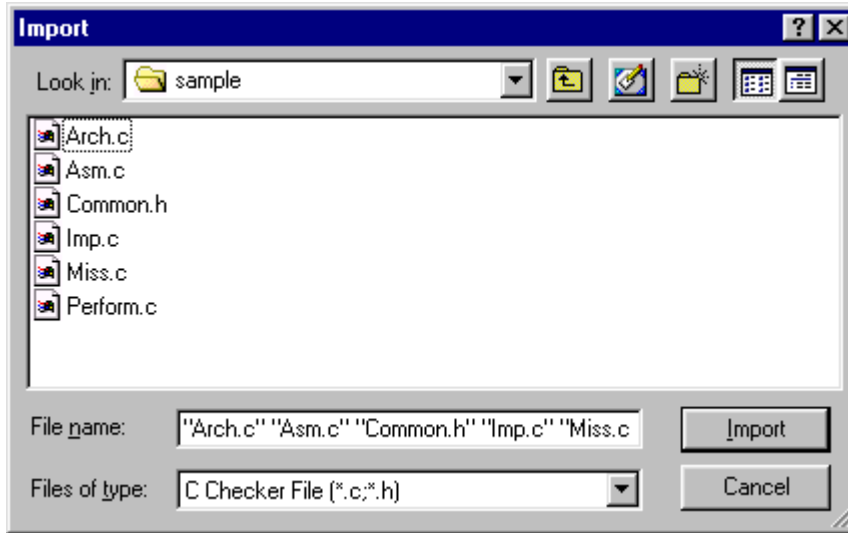
Figure 2.2-d [Import] dialog box
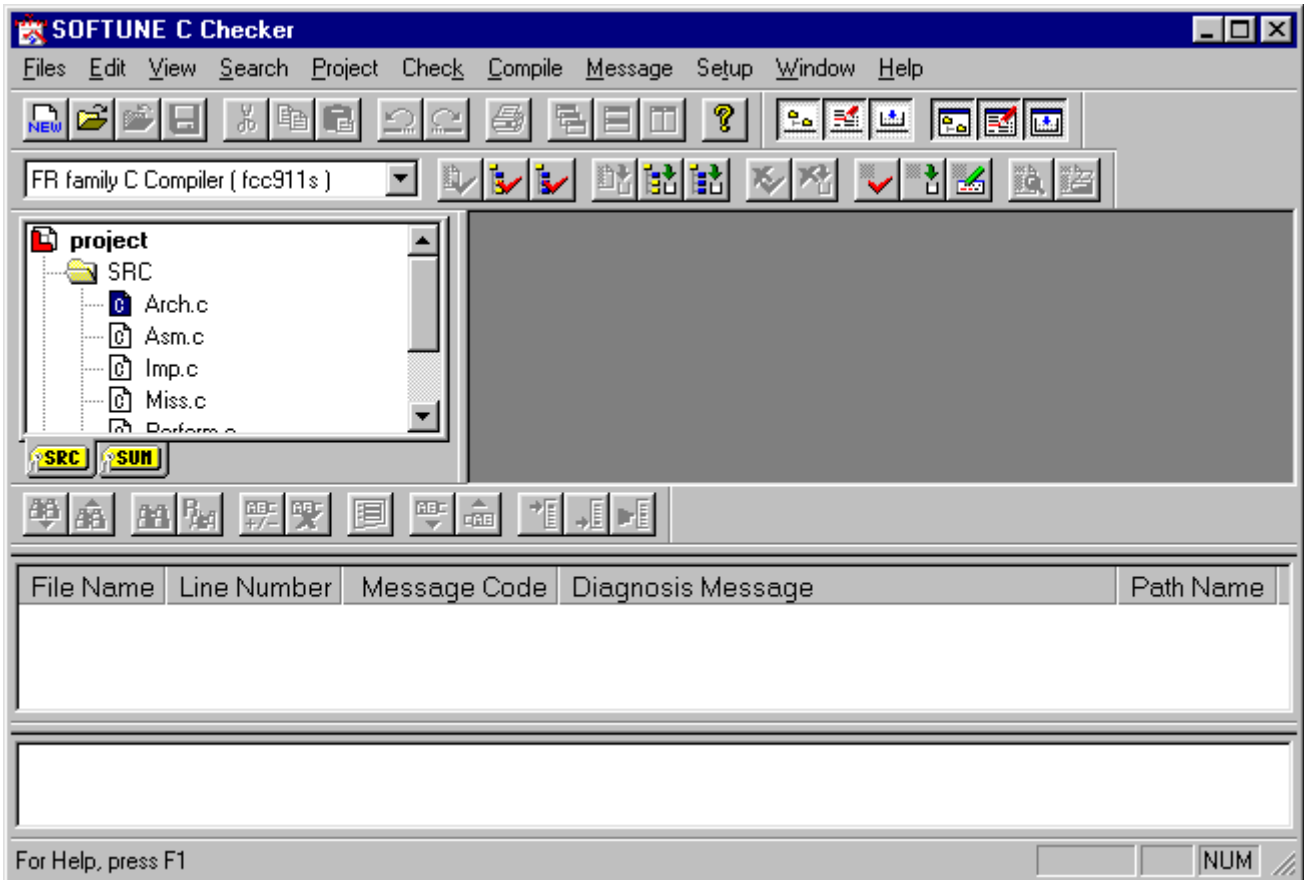
Figure 2.2-e Window of the file registered in project

28

■ **[Open Project]**

Choose the [Open] command from the [Project] menu, and the [Open] dialog box, shown in Figure 2.2-f, appears. Select a file with the ".cpj" file identifier, and the project file made before is opened.
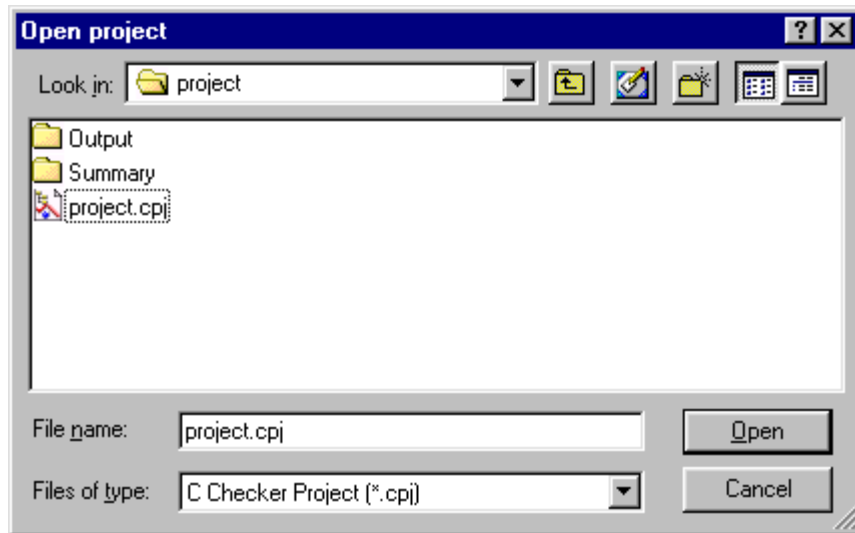
Figure 2.2-f [Open Project] dialog box

2.3   Setting up Check Options

Before using the C Checker to check a source program, set up the required options.

■ **Setting up check options**

Choose the [Check Option] command from the [Check] menu, and the [Check Option] dialog box, shown in Figure 2.3-a, appears.

Figure 2.3-a shows the initial values that the check options assume when the C Checker is started.

Select the check item according to the purpose.
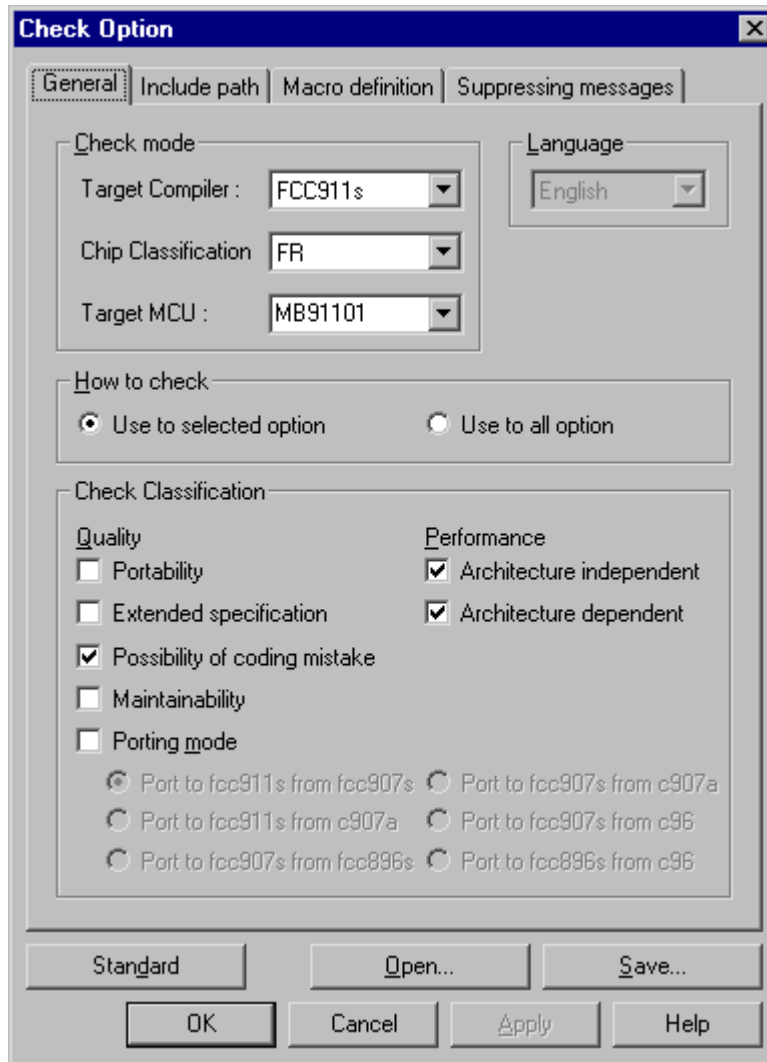
Figure 2.3-a Initial values of the check options

2.4   Starting a Check

To start a check using the C Checker, choose the [Active Document] command from the [Check] menu. Alternatively, click the [Active Document] button in the check tool bar.

■ **Starting a check**

To start a check, choose the one command from the [Check] menu. ([Active Document] command or [Selected Files] command or [All Files] command.) (See Figure 2.4-a.)

The check is begun immediately after the selection.

It is necessary to open project file to select [Selected Files] command and [All Files] command.

Figure 2.4-a [Check] menu

31
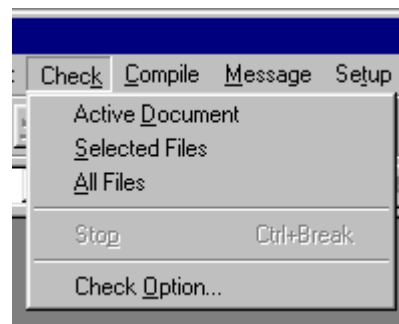
■ **Diagnostic messages**

Upon completion of a check, diagnostic messages are displayed on the screen, as shown in Figure 2.4-b.
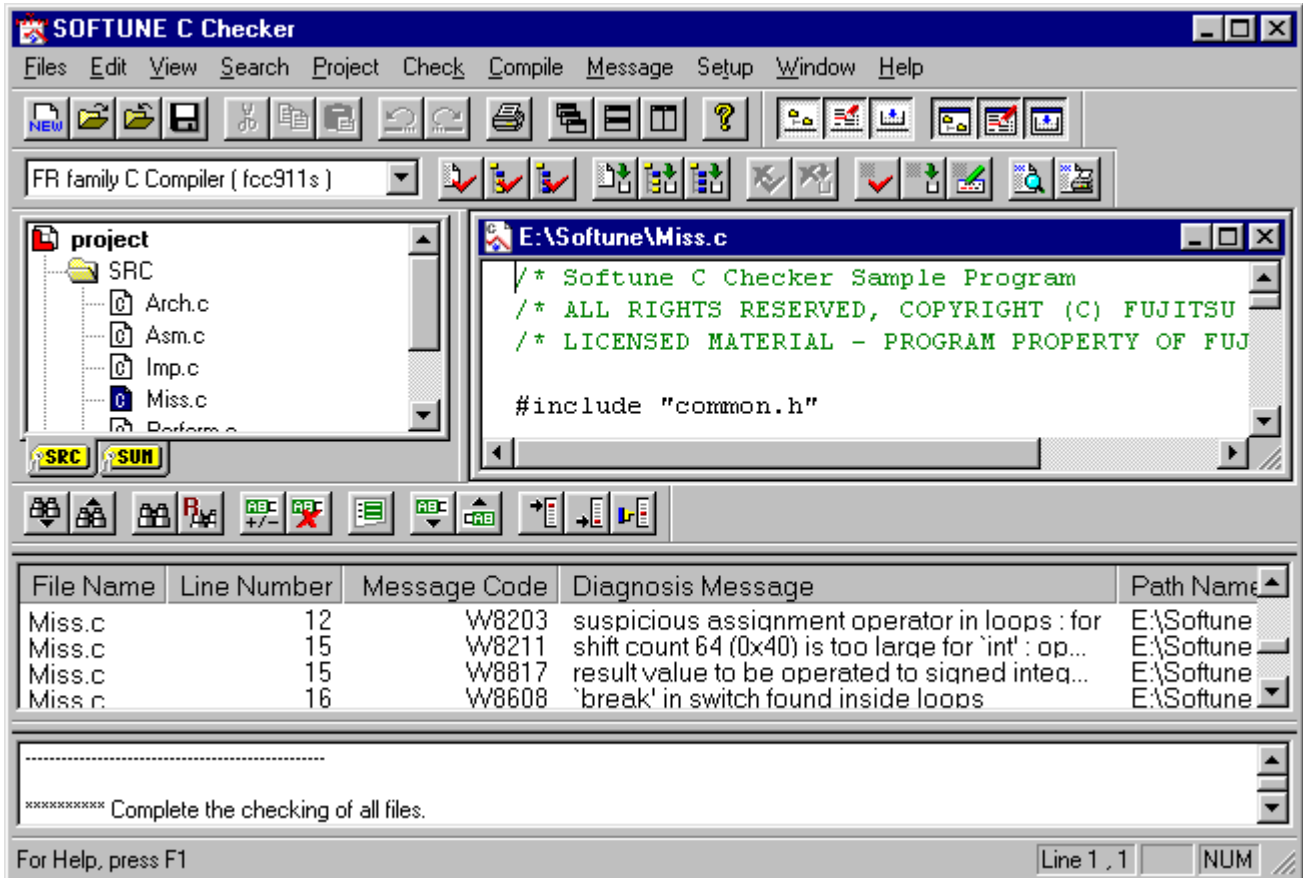
Figure 2.4-b Diagnostic messages

■ **Jumping to the lines with a problem found**

In the diagnostic message window, double-click a warning line with the left mouse button, and the cursor jumps to the corresponding line on the edit screen. (See Figure 2.4-c.)
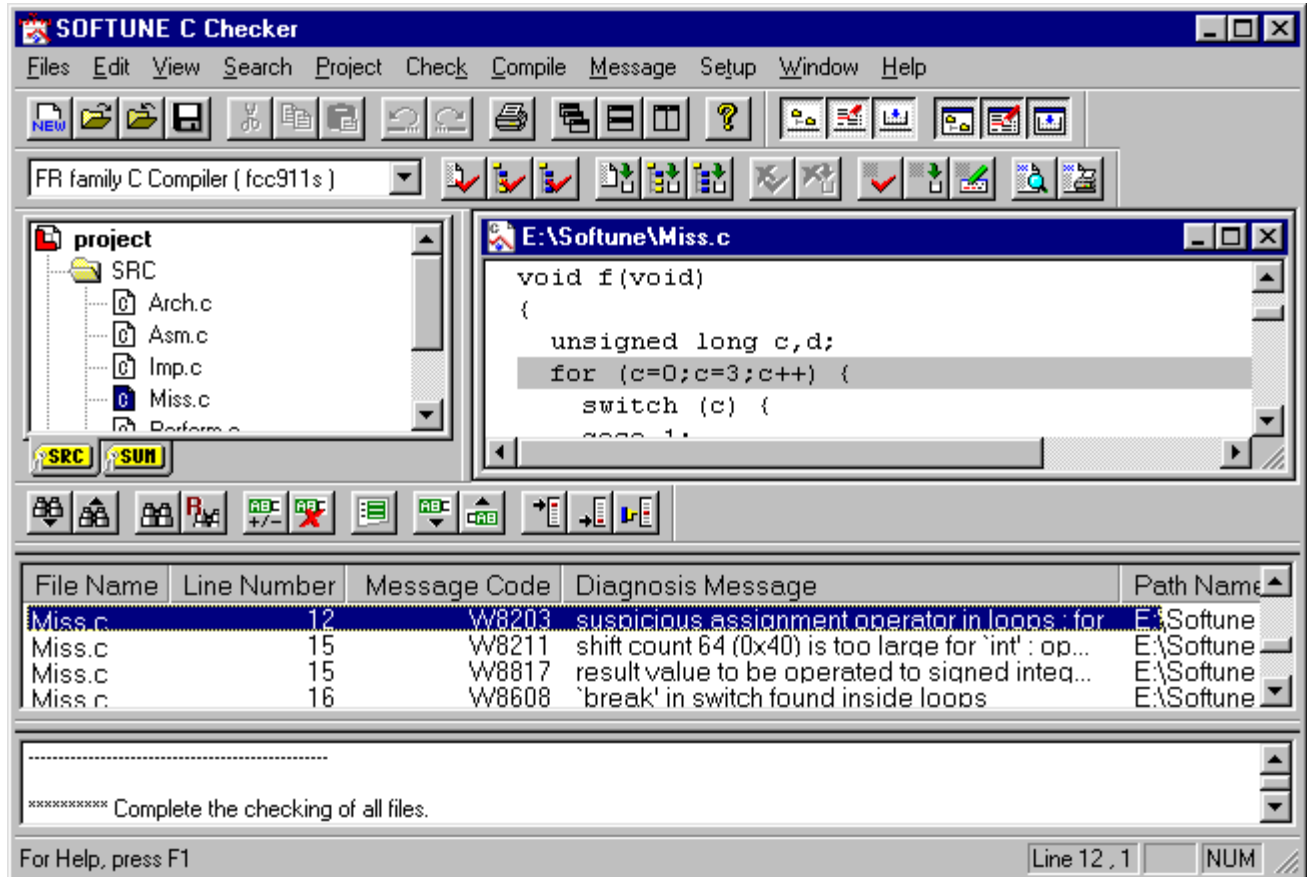
Figure 2.4-c Jumping to the line with a problem found

■ **Sorting of the diagnostic message**

The diagnostic message can be sorted by selecting [All Message], [Hide Message] command from pop-up menu displayed when mouse's right button is clicked on diagnostic message window.

Figure 2.4-d Sorting of the diagnostic message

■ **Displaying detailed messages**

Choose the [Detailed message] command from the [Diagnostic message] menu, or [Detail Message] command in the pop-up menu displayed when mouse's right button is clicked, the detailed explanation of the diagnostic message is displayed. (See Figure 2.4-d.)

```
SOFTUNE C Checker Help                                    _ □ ✕

File   Edit   Bookmark   Options   Help

Contents │ Index │ Back │ Print │  <<  │  >>  │

W8203  Remark on the use (For statement) of assignment
operator
                                                      EXPRESSION

  Sample program      Program correction plan      Key advice

The assignment operator (=) is used in the FOR statement, but it can
be an equivalence operator error (==).

The conditional expression in the FOR statement is judged to be true when
the result is other than 0 and is judged to be false when the result is 0.
In other words, because the value in the conditional expression is used, the
condition is judged from the result of the assignment, not from the result of
the comparison, when the expression is an assignment expression.
Hence, the conditional expression can be an infinite loop when the result of
the assignment is other than 0.
Because it could be a typographical error, review the logic well, then make
the correction.
When you want to judge the condition using the result of the assignment
expression, the logic can be clear if the comparison is performed using the
equivalence operator after using the assignment operator.


[Sample program]

    void func(void)
    {
        unsigned int a=0;
        while(a=1){                    /* Remarked point    */
            a=sub(a);
        }
    }

[Program correction plan]

  (1) In the case of a typographical error
      Correct the assignment operator to the equivalence operator.
```
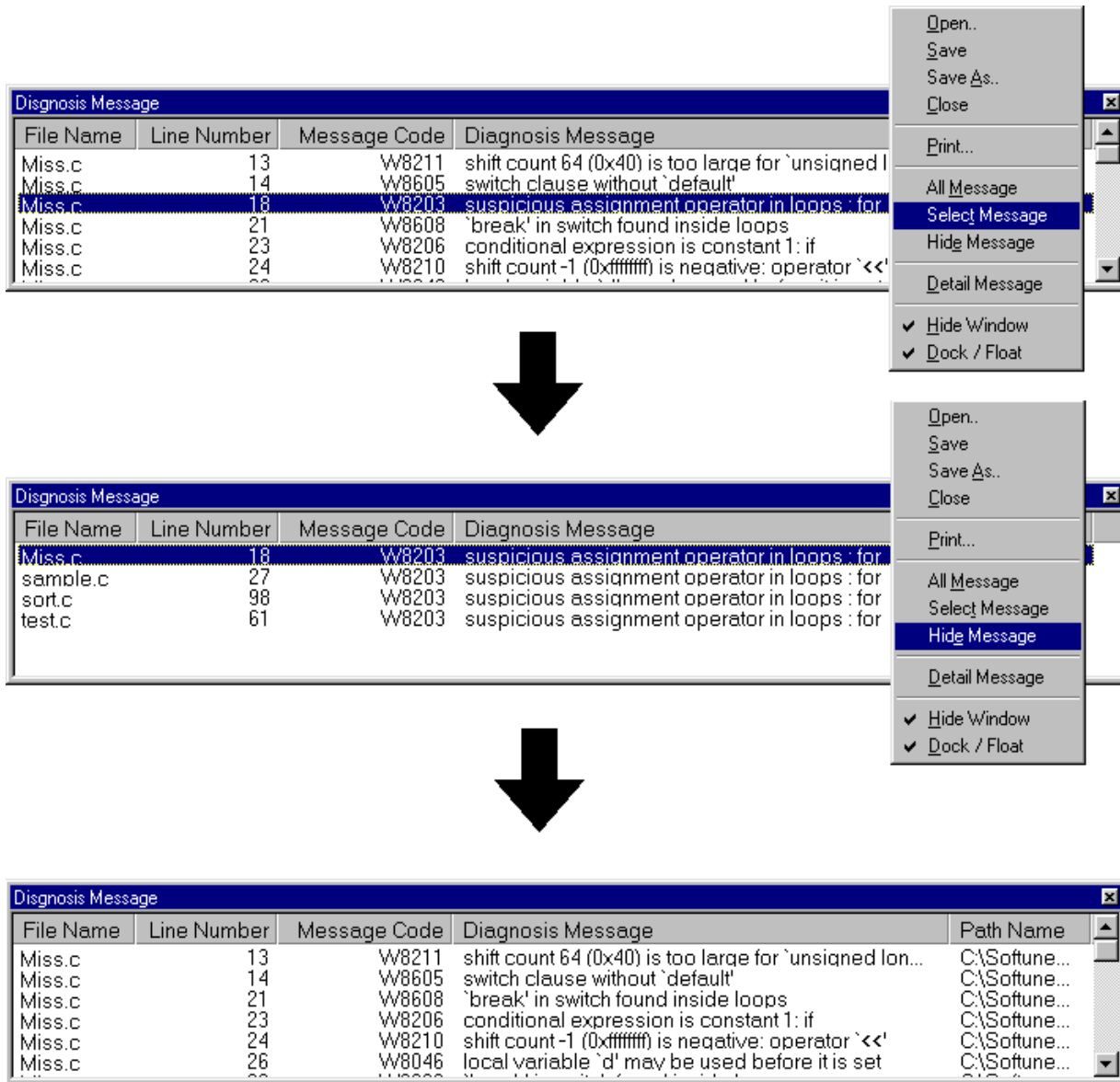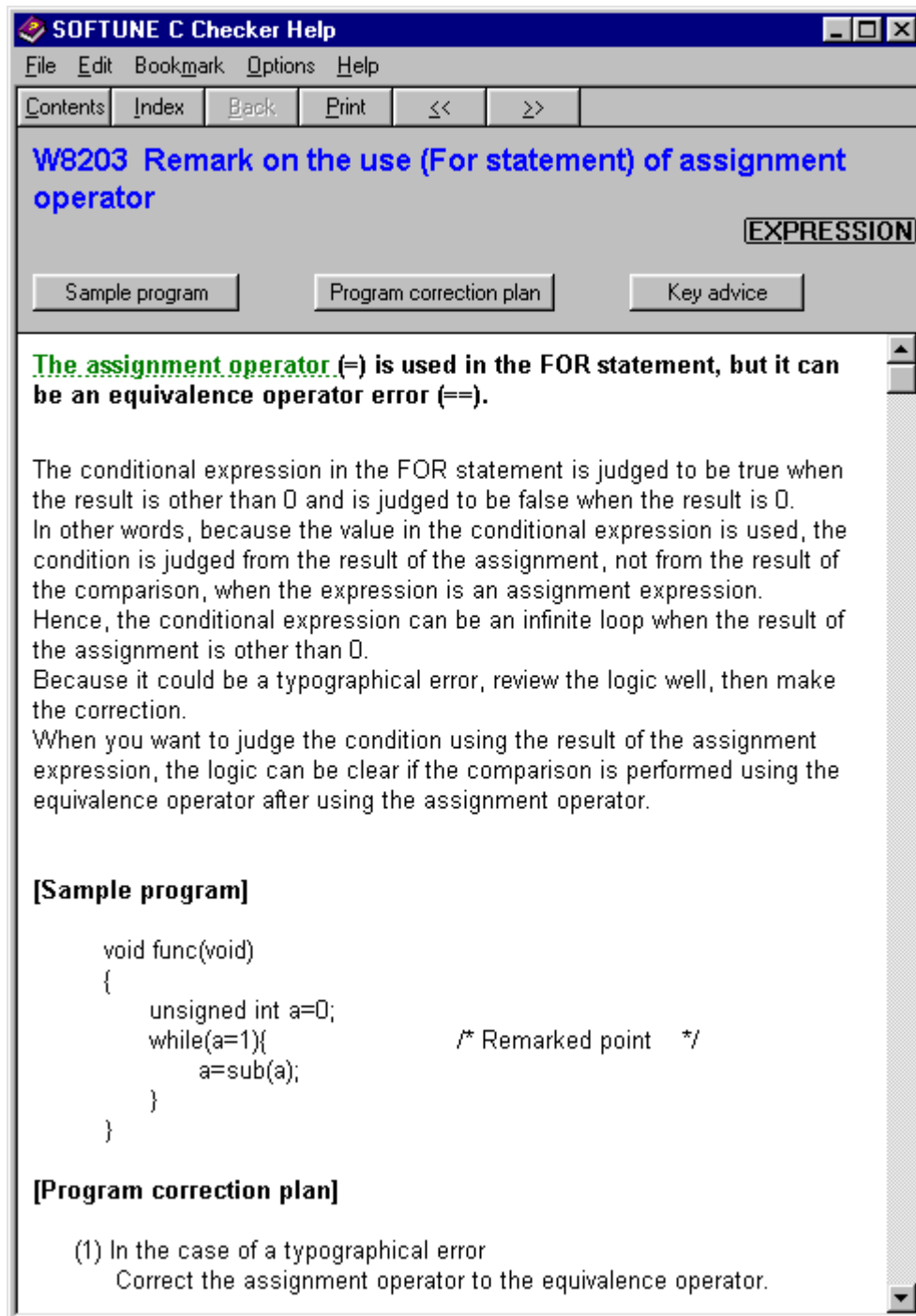
Figure 2.4-e Detailed message

35

2.5   Aborting a Check

To abort a check before it terminates, choose the [Stop] command from the [Check] menu. Alternatively, click the [Stop] button in the check tool bar.

■ **Aborting a check**

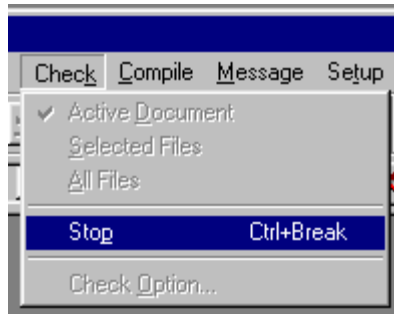Choose the [Stop] command from the [Check] menu, and the check aborts.

Figure 2.5-a [Stop] menu

36

2.6   Exiting from the C Checker

To exit from the C Checker, click the [Exit] command from the [Files] menu. (See Figure 2.6-a.) This causes the system to terminate normally.

■ **Exiting from the C Checker**

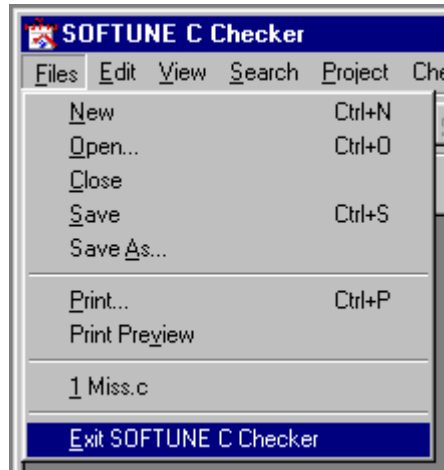Choose the [EXIT SOFTUNE C Checker] command from the [Files] menu, and the C Checker is ended.

Figure 2.6-a [Files] menu

37

CHAPTER 3   Setting up Check Options

This chapter explains how to set up check conditions.

3.1   General

Set up the general options to be passed to the C Checker.

■ **Dialog box for setting up check options**

Choose [Check Option] command from the [Check] menu or [Check Option] command from the pop-up menu in the project management window (See Figure 3.1-a), the [Check Option] dialog box is displayed. Figure 3.2-a shows the dialog box for setting up check options when select [General] tab which exists in the [Check Option] dialog box.
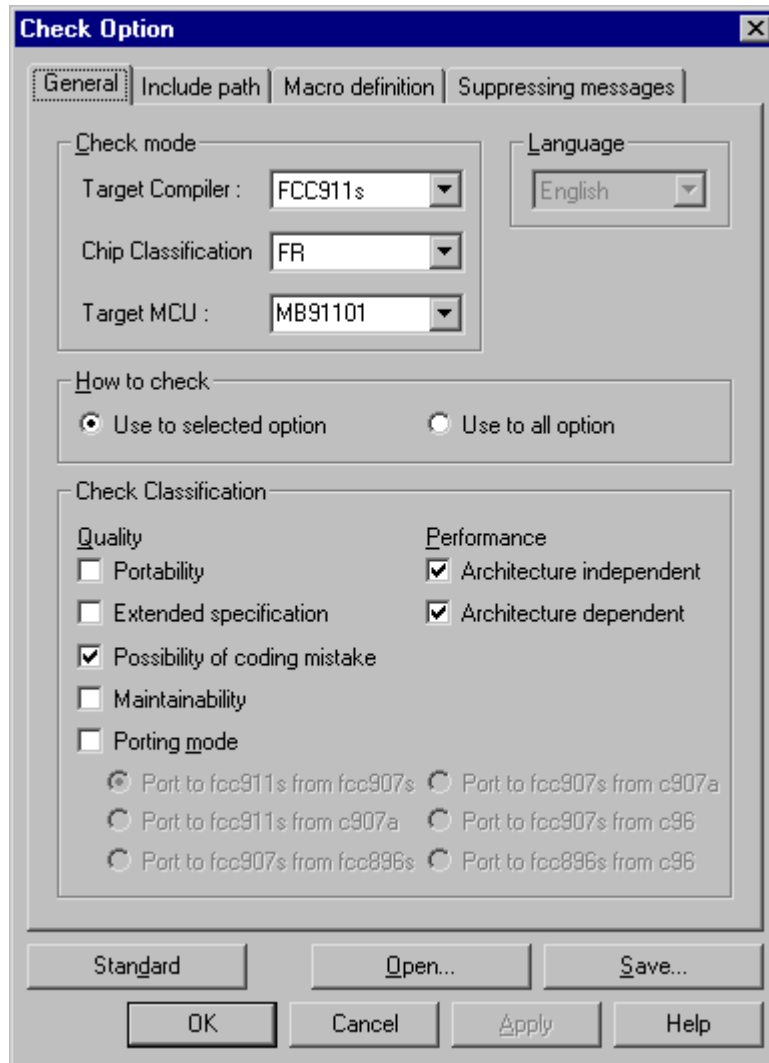

Figure 3.1-a Dialog box for setting up check options

40

■ **Check mode**

Set the operation mode for checking.

❍ Target compiler

Select the target compiler used for a check.

Target compiler	fcc911s compiler
	fcc907s compiler
	fcc896s compiler
	ANSI C compiler

❍ Chip classification

Select the chip type of the target to be checked.

The chip type that can be selected depends on the compiler type.

❍ Target MCU

Specify the target MCU to be checked.

■ **Language**

The language of the message displayed in the diagnostic message window is shown.

Language	English

■ **How to check**

Specify the check mode.

Select one of the following check modes.

- Use to selected items (Check only the selected check items.)

- Use to all items (Check all check items.)

41

■ **Check classification**

Check classification is divided into quality and performance items, so that a detailed check can be performed for both types of items.

Quality	Portability	
	Extended specification	
	Possibility of coding mistake	
	Maintainability	
	Porting mode	Port to fcc911s from fcc907s[1]
		Port to fcc911s from c907a[1]
		Port to fcc907s from fcc896s[1]
		Port to fcc907s from c907a[1]
		Port to fcc907s from c96[1]
		Port to fcc896s from c96[1]
Performance	Architecture independent	
	Architecture dependent	

*1: These options become effective only when porting mode is effective.

■ **[Standard] button**

Recovers all the settings to default value.

■ **[Open] button**

Reads the settings of the options from an information file.

■ **[Save] button**

Saves all the settings into a file.

If the settings of any check options have been changed, a confirmation dialog box appears asking whether the changes should be saved before the options are saved into the information file.

To cancel the changes, click the [Yes] button in the dialog box.

To save the changes, click the [No] button in the dialog box. Then, click the [Apply] button and click the [Save] button again.

42

3.2   Include Paths

Set up the include paths to be passed to the C Checker.

■ **Dialog box for setting up include paths**

Figure 3.2-a shows the dialog box for setting up include paths when select [Include Path] tab which exists in the [Check Option] dialog box.
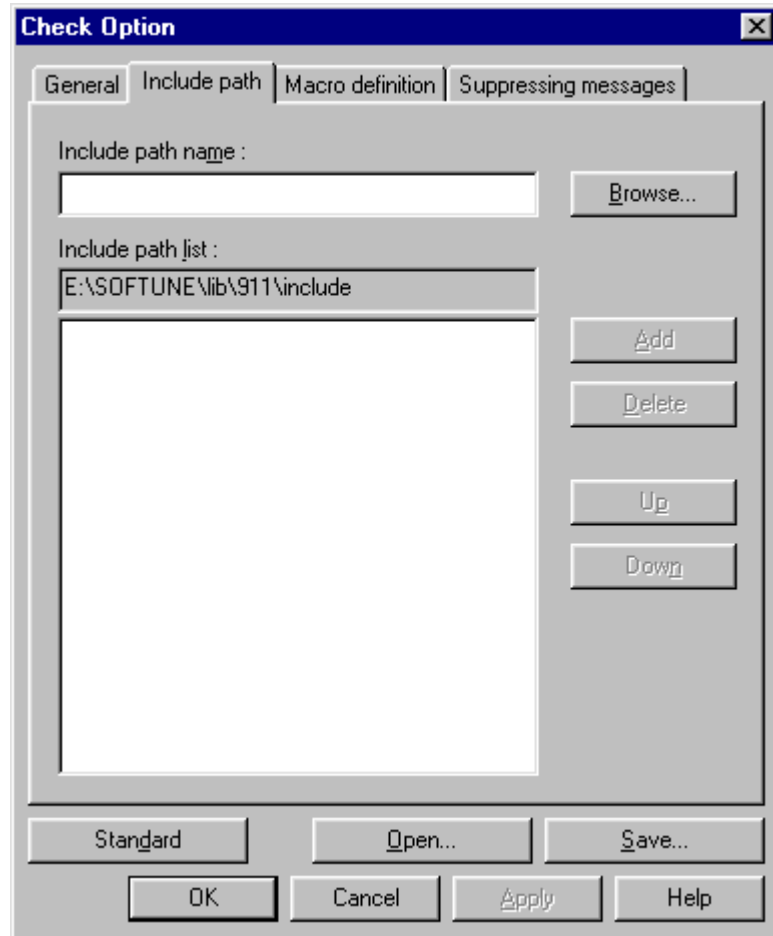

Figure 3.2-a Dialog box for setting up include paths

■ **Explanation**

The dialog box for setting up include paths contains the following:

❍ Include path name field

In this field, enter the include path to be registered.

❍ Include path list

The registered include paths are displayed.

❍ [Browse] button

This button displays the dialog box for browsing include paths.

From this dialog box, an include path can be selected and entered in the include path entry field.

❍ [Add] button

This button adds the include path entered in the include path entry field to the include path register list.

❍ [Delete] button

This button deletes an include path from the include path register list.

To delete an include path, click the include path in the register list with the left mouse button and click this button.

❍ [Up] button

This button moves an include path registered in the include path register list in the upward direction.

The C Checker searches for the files to be included through the registered include paths, starting from the uppermost path.

❍ [Down] button

This button moves an include path registered in the include path register list in the downward direction.

❍ [Standard] button

Recovers all the settings to default value.

❍ [Open] button

Reads the settings of the options from an information file.

❍ [Save] button

Saves all the settings into a file.

If the settings of any check options have been changed, a confirmation dialog box appears asking whether the changes should be saved before the options are saved into the information file.

To cancel the changes, click the [Yes] button in the dialog box.

To save the changes, click the [No] button in the dialog box. Then, click the [Apply] button and click the [Save] button again.

44

❍ [Help] button

This button displays a help file.

■ **Entry procedure**

Set up an include path with the following procedure:

1) In the include path name field, enter an include path. (An include path can also be entered using the [Browse] button.)

2) If it is correct, add the path to the register list shown below the field, using the [Add] button.

3) To change the position of the include path in the register list, use the [Up] and [Down] buttons as appropriate.

4) Delete the unnecessary include paths using the [Delete] button, if any.

3.3   Macro Definitions

Set up the macro definitions to be passed to the C Checker.

■ **Dialog box for setting up macro definitions**

Figure 3.3-a shows the dialog box for setting up macro definitions when select [Macro definition] tab which exists in the [Check Option] dialog box.
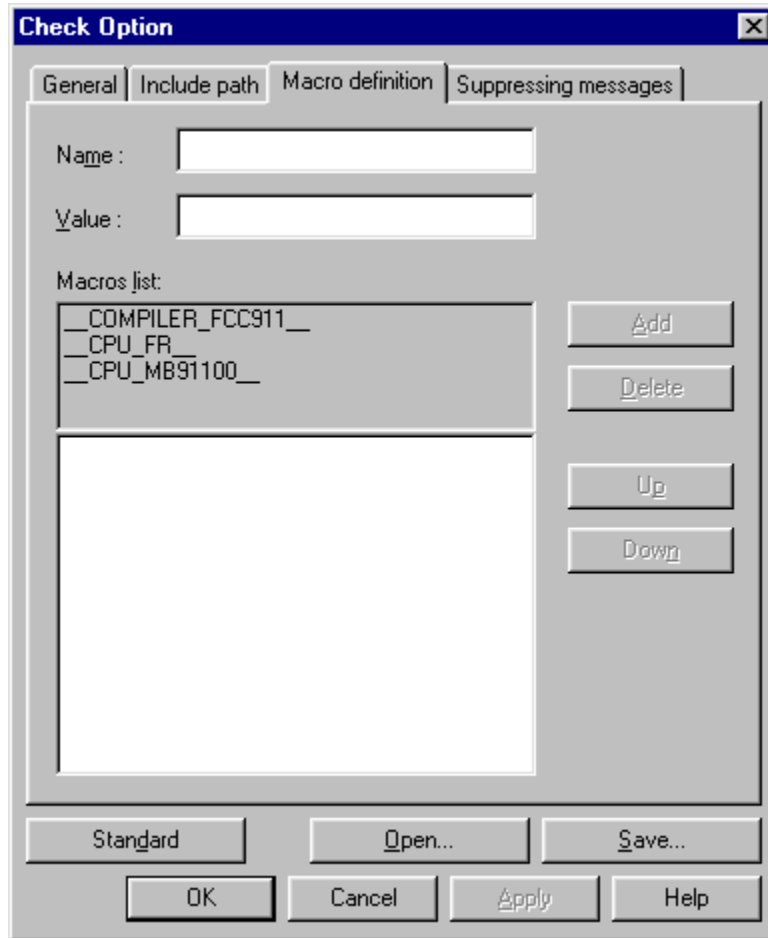

Figure 3.3-a Dialog box for setting up macro definitions

46

■ **Explanation**

The dialog box for setting up macro definitions contains the following:

❍ Macro name field

In this field, enter the macro to be registered.

❍ Macro value field

In this field, enter the value of macro.

❍ Macros list

The registered macros are displayed.

❍ [Add] button

This button adds the macro entered in the macro entry field to the macro register list.

❍ [Delete] button

This button deletes a macro from the macro register list.

To delete a macro, click the macro in the register list with the left mouse button and click this button.

❍ [Up] button

This button moves a macro registered in the macro register list in the upward direction.

The C Checker defines the macros in the order in which they are registered.

❍ [Down] button

This button moves a macro registered in the macro register down one place in the macro register list.

❍ [Standard] button

Recovers all the settings to default value.

❍ [Open] button

Reads the settings of the options from an information file.

❍ [Save] button

Saves all the settings into a file.

If the settings of any check options have been changed, a confirmation dialog box appears asking whether the changes should be saved before the options are saved into the information file.

To cancel the changes, click the [Yes] button in the dialog box.

To save the changes, click the [No] button in the dialog box. Then, click the [Apply] button and click the [Save] button again.

❍ [Help] button

This button displays a help file.

47

■ **Entry procedure**

Set up a macro with the following procedure:

1) In the macro name field, enter a macro.

2) If it is correct, add the macro to the register list shown below the field, using the [Add] button.

3) To change the position of the macro in the register list, use the [Up] and [Down] buttons as appropriate.

4) Delete the unnecessary macros using the [Delete] button, if any.

3.4   Suppressing Messages

Diagnostic messages can be suppressed.

■ **Dialog box for suppressing messages**

Figure 3.4-a shows the dialog box for suppressing messages when select [Suppressing messages] tab which exists in the [Check Option] dialog box.
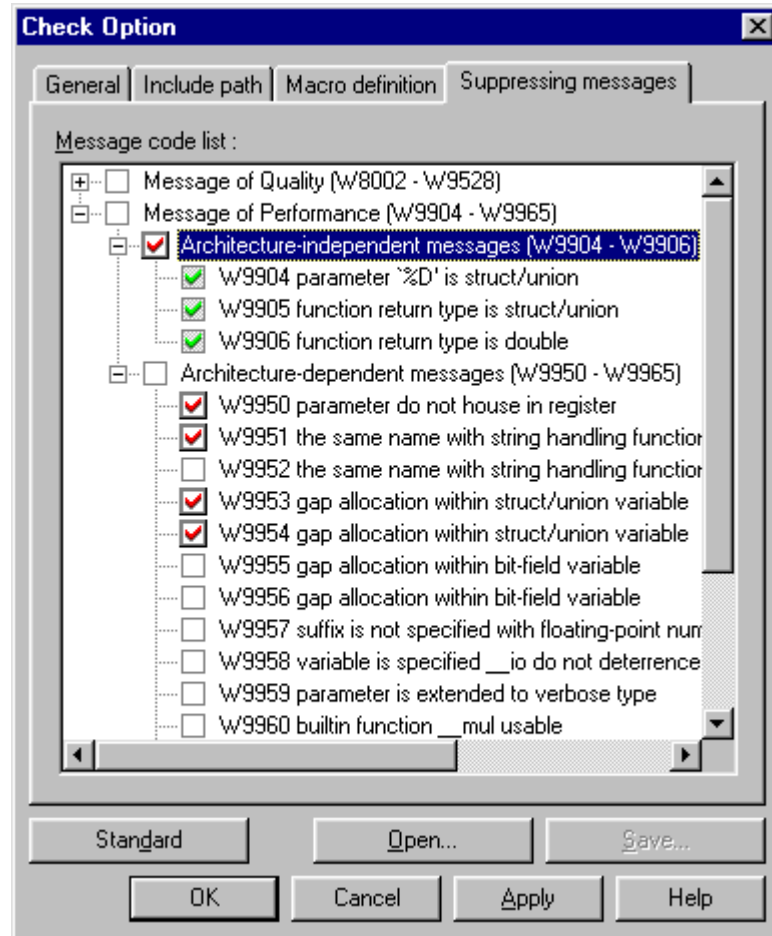

Figure 3.4-a Dialog box for suppressing messages

■ **Explanation**

The dialog box for suppressing messages contains the following:

❍ Check box of suppressing category and number

In this check box, check the category or number of the message to be suppressed.

❍ [Standard] button

Recovers all the settings to default value.

❍ [Open] button

Reads the settings of the options from an information file.

❍ [Save] button

Saves all the settings into a file.

If the settings of any check options have been changed, a confirmation dialog box appears asking whether the changes should be saved before the options are saved into the information file.

To cancel the changes, click the [Yes] button in the dialog box.

To save the changes, click the [No] button in the dialog box. Then, click the [Apply] button and click the [Save] button again.

■ **Entry procedure**

Set the number of the message to be suppressed, with the following procedure:

1) When the category is suppressed, the check box of the category is checked.

2) When each number is suppressed, the check box of each number is displayed clicking the icon of + in front of the check box of the category.

CHAPTER 4   File

This chapter explains how to manipulate files.

4.1 File Menu

The commands on the File menu can be used to create a new file and to open, close, save, and print a file.

The File menu provides the following commands:

- **[New] command**
- **[Open] command**
- **[Close] command**
- **[Save] command**
- **[Save as] command**
- **[Print] command**
- **[Print preview] command**
- **[Recent file] command**
- **[Exit SOFTUNE C Checker] command**

■ **File menu**

❍ [New] command

Creates a new file. Choosing this command opens a new file with no title on the edit screen.

To open an existing file, use the [Open] command.

❍ [Open] command

Opens an existing file.

Choosing this command displays the [Open] dialog box. In this dialog box, files of all types can be displayed. By default, however, only C source files "*.c" are displayed.

Multiple files can be opened at one time. To switch from one file to another, use the window list in the [Window] menu.

To create a new file, use the [New] command.

❍ [Close] command

Closes a currently open window. If the file has been modified, a confirmation dialog box appears asking whether the file should be saved before it is closed. If the file is closed without being saved, all the modifications made after the file was last saved are lost. If this command is chosen for a file without a name, the [Save as] dialog box appears.

In the file name box in the dialog box, enter the appropriate file name for the file. A file can also be closed by double-clicking the control box in the window of the file.

❍ [Save] command

Saves the active file without changing the file name or the location it is saved. If a newly created file is to be saved for the first time, the [Save as] dialog box appears, allowing the user to save the active file with an appropriate file name. To save a file by changing the file name or the location it is saved, use the [Save as] command.

❍ [Save as] command

Saves the active file by assigning a file name for it. Choosing this command displays the [Save as] dialog box, allowing the user to save the file by assigning an appropriate file name for it.

To save the active file without changing the file name or the location it is saved, use the [Overwrite] command.

❍ [Print] command

Prints the contents of the active file. Choosing this command displays the [Print] dialog box. In this box, set the desired printing attributes to print the file.

❍ [Print preview] command

Displays what is to be printed in the [Preview] window.

The [Preview] window contains the following functions:

- [Print] button]: Displays the [Print] dialog box.

- [Set up printing] button: Sets the printing format. Choosing this button displays the [Set up printing] dialog box, allowing the user to set the detailed printing format.

- [Next page] button: Displays a preview of the next page.

- [Previous page] button: Displays a preview of the previous page.

- [Zoom in] button: Displays an enlarged preview.

- [Zoom out] button: Displays a reduced preview.

- [Close] button: Exits from the preview function.

❍ [Recent files] command

Stores the names of up to five previously opened files.

If more than five files are opened, the excess names are removed from the list, starting from the oldest one.

❍ [Exit from SOFTUNE C Checker] command

Exits from the C Checker.

53

4.2   Dialog Box of the [Open] Command

The [Open] dialog box allows the user to set the following:
- **File name**
- **Files of type**
- **Look in**

■ **Dialog box of the [Open] command**

Choose the [Open] command from the [Files] menu, and the [Open] dialog box, shown in Figure 4.2-a, appears.
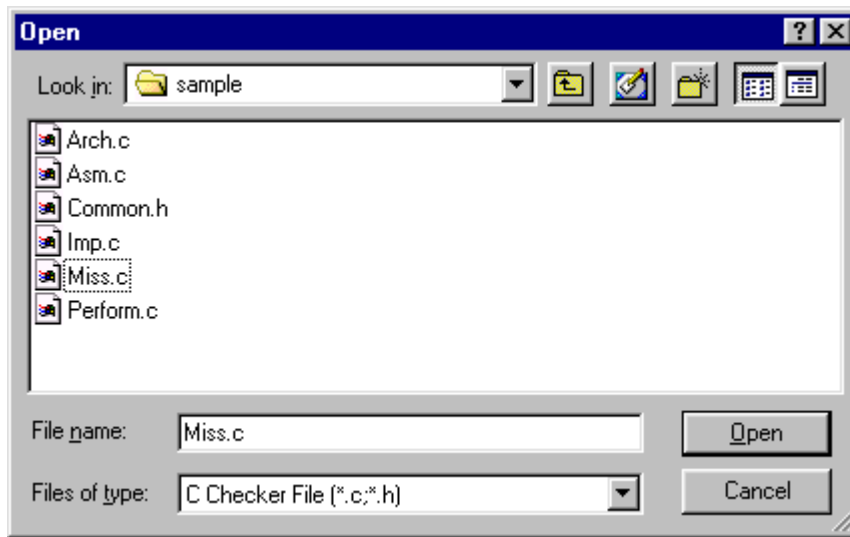

Figure 4.2-a [Open] dialog box

❍ File name

Enter or select the name of the file to be opened. In the list box, the files with the identifier corresponding to the file type selected from the [File type] box are listed.

❍ Files of type

Select the file type of the files to be displayed.

In the list box, files of all types can be displayed. By default, however, only C source files "*.c" and header files "*.h" are displayed.

❍ Look in

Select the place with the opened file.

54

4.3   Dialog Box of the [Save as] Command

The [Save as] dialog box allows the user to set the following, such as a file name and the location where a file is to be saved:

- **File name**
- **Save as type**
- **Save in**

■ **Dialog box of the [Save as] command**

Choose the [Save as] command from the [Files] menu, and the [Save as] dialog box, shown in Figure 4.3-a, appears.
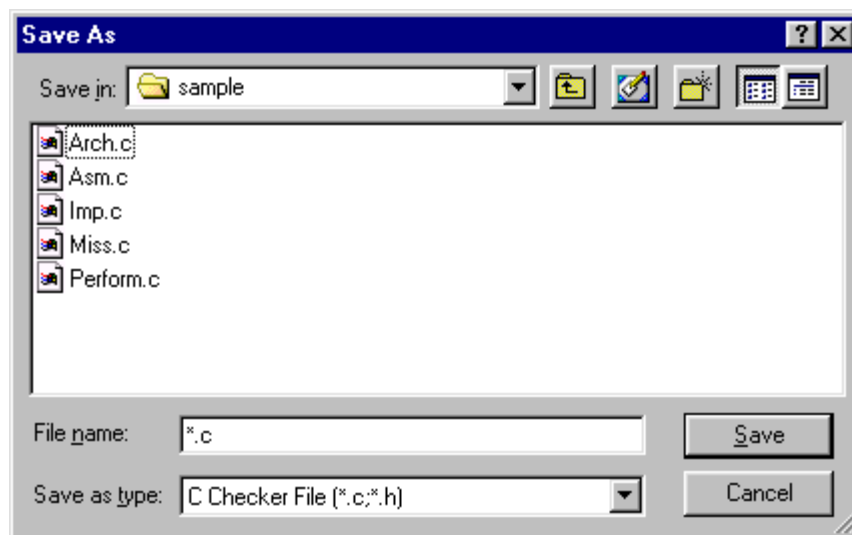
Figure 4.3-a [Save as] dialog box

❍ File name

Enter a file name different from the old one.

❍ Save as type

Select the file type of the files to be displayed.

In the list box, files of all types can be displayed.

By default, however, C source files "*.c" and header files "*.h" are displayed.

❍ Save in

Select the place where the file is saved.

55

4.4   Dialog Box of the [Print Set] Command

The [Print Setup] dialog box allows the user to set the details of the printing format:
- **Columns**
- **Header printing**
- **Footer printing**
- **Line number printing**
- **Keyword highlighting**
- **Top margin**
- **Left margin**

■ **Dialog box of the [Print Set] command**

Choose the [Print PreView] command from the [File] menu, and the PreView window is displayed.

Choose the [Print Set] command in the PreView window, and the [Print Setup] dialog box, shown in Figure 4.4-a, appears.
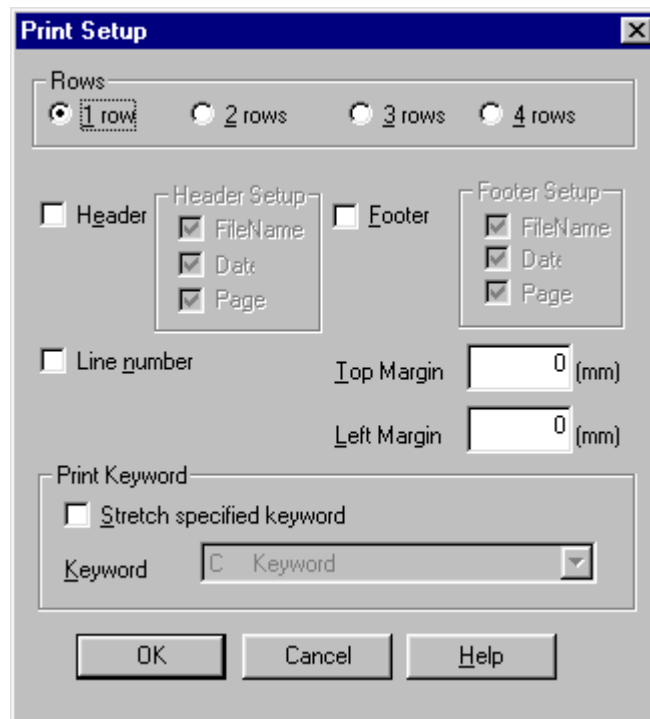

Figure 4.4-a [Print Setup] dialog box

❍ Rows

Set the number of columns. One of one, two, three, and four columns can be selected.

The default is one column.

❍ Header

Specify whether a header is to be printed. As a header, a file name, page number, and date and time can be set.

The default is no header.

❍ Footer

Specify whether a footer is to be printed. As a footer, a file name, page number, and date and time can be set.

The default is no footer.

❍ Line number

Specify whether to print line numbers.

The default is no line numbers.

❍ Print keyword

This option highlights the keyword.

Make "Stretch specified keyword" effective, and select the keyword which wants to highlight.

The default is no highlighting.

The keyword which can be selected is as follows.

• C keyword

• ASM keyword

• Extension keyword

• User-defined keyword

❍ Top margin

Set the top margin.

❍ Left margin

Set the left margin.

CHAPTER 5   Edit

This chapter explains how to use the Edit menu.

5.1   Edit Menu

5.1   Edit Menu

The Edit menu provides the following:
- **[Undo] command**
- **[Redo] command**
- **[Cut] command**
- **[Copy] command**
- **[Paste] command**
- **[Delete] command**
- **[Select all] command**

■ **Edit menu**

❍ [Undo] command

Cancels the immediately preceding editing operation and restores the file to the previous state.

❍ [Redo] command

Cancels the immediately preceding undo operation and restores the file to the previous state.

❍ [Cut] command

Cuts the currently selected data from the file and saves it into the clipboard. If no data has been selected, this command cannot be used.

When data is cut and saved into the clipboard, it replaces the data previously saved in the clipboard.

❍ [Copy] command

Copies the selected data in the file to the clipboard. If no data has been selected, this command cannot be used.

When data is copied to the clipboard, it replaces the data previously saved in the clipboard.

❍ [Paste] command

Inserts the contents of the clipboard into the cursor position.

If nothing is saved in the clipboard, this command cannot be used.

❍ [Delete] command

Deletes the currently selected data from the file. The deleted data is not saved into the clipboard.

❍ [Select all] command

Selects all data in the file.

CHAPTER 6   Search

This chapter explains how to use the Search menu.

6.1   Search Menu

To search for a character string, choose the [Search] command from the [Search] menu. Choosing the [Search] command displays the [Current file search] dialog box. In the dialog box, enter the desired character string in the [Character string to search for] box, specify the search direction and the search start position, and click the [Next] button. Search starts in the specified direction.

The search menu provides the following commands:
- **[Find] command**
- **[Find Next (Down)] command**
- **[Find Next (Up)] command**
- **[Replace] command**
- **[Find in Files] command**
- **[Jump Tag Next (Down)] command**
- **[Jump Tag Next (Up)] command**
- **[Add Tag (or Delete Tag)] command**
- **[List Tag] command**
- **[Jump to Line] command**
- **[Jump to Bottom] command**
- **[Jump to Top] command**

■ **Search menu**

❍ [Find] command

Searches for a character string through the edit screen.

Choosing this command displays the [Find in Current File] dialog box. In this box, enter the character string to search for.

❍ [Find Next (Down)] command

Searches for a character string through the edit screen in the forward direction, starting from the cursor position.

The command searches for the character string entered in the [Find in Current File] dialog box. Once this command is selected, the search can be repeated without opening the [Find in Current File] dialog box.

❍ [Find Next (Up)] command

Searches for a character string through the edit screen in the backward direction, starting from the cursor position.

The command searches for the character string entered in the [Find in Current File] dialog box. Once this command is selected, the search can be repeated without opening the [Find in Current File] dialog box.

❍ [Replace] command

Replaces a character string on the edit screen with another string.

Choosing this command displays the [Replace] dialog box. In this box, enter the character string to be replaced.

❍ [Find in Files] command

Search a string in the files within the target directory.

If this command is selected, [Find in Files] dialog box appears. Then, enter the string to be searched and the target directory.

❍ [Jump Tag Next (Down)] command

Jumps from the current cursor position on the edit screen to the next tag. Tags can be added using the [Add Tag (or Delete Tag)] command.

❍ [Jump Tag Next (Up)] command

Jumps from the current cursor position on the edit screen to the previous tag. Tags can be added using the [Add Tag (or Delete Tag)] command.

❍ [Add Tag (Delete Tag)] command

Adds a tag to the current cursor position on the edit screen.

If a tag has already been set at the cursor position, this command deletes the tag.

❍ [Delete Tag All] command

Deletes all tags from the edit screen.

❍ [List Tag] command

Lists the tags existing on the edit screen.

Choosing this command displays the [Tag list] dialog box, allowing the user to reference and delete the existing tags.

❍ [Jump to Line] command

Jumps to a specified line on the edit screen.

Choosing this command displays the [Jump to Target Line] dialog box. In the box, enter the line to jump to.

❍ [Jump to Bottom] command

Jumps to the last line on the edit screen.

❍ [Jump to Top] command

Jumps to the first line on the edit screen.

6.2   Dialog Box of the [Find in Current File] Command

**The [Current file search] dialog box allows the user to search for a character string
through the edit screen.**

■ **Dialog box of the [Find in Current File] command**

Choose the [Find] command from the [Search] menu, and the [Find in Current File] dialog box,
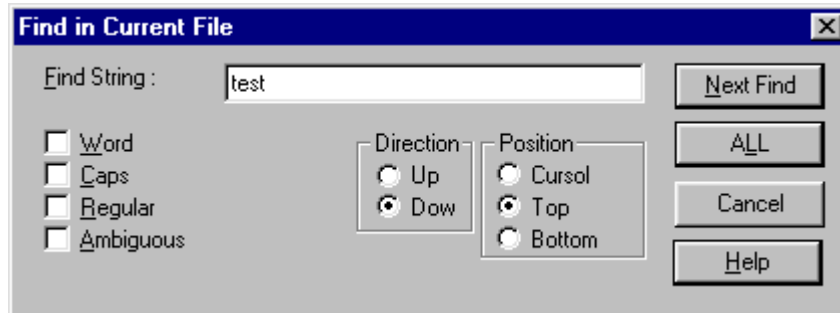shown in Figure 6.2-a, appears.

Figure 6.2-a [Find in Current File] dialog box

In the [Find in Current File] dialog box, the following can be set:

❍ Find string

❍ Search for whole word

❍ Case sensitivity

❍ Normal representation

❍ Search for uncertain string

❍ Direction

❍ Position

Note: Upon completion of search, a message box appears indicating the number of
occurrences of the character string that were found.

6.3   Dialog Box of the [Replace] Command

The [Replace] dialog box allows the user to replace a character string on the edit screen with another string.

■ **Dialog box of the [Replace] command**

Choose the [Replace] command from the [Search] menu, and the [Replace] dialog box, shown in Figure 6.3-a, appears.

Figure 6.3-a [Replace] dialog box

In the [Replace] dialog box, the following can be set:

○ Find string

○ Exchange string

○ Search for whole word

○ Case sensitivity

○ Normal representation

○ Search for uncertain string

○ Direction

○ Position

65

6.4   Dialog Box of the [Find in Files] Command

Use the [Find in Files] dialog box to search for a string from the target file in the target directory.

■ **Dialog box of the [Find in Files] command**

Choose the [Find in Files] command from the [Search] menu, and the [Find in Files] dialog box, shown in Figure 6.4-a, appears.
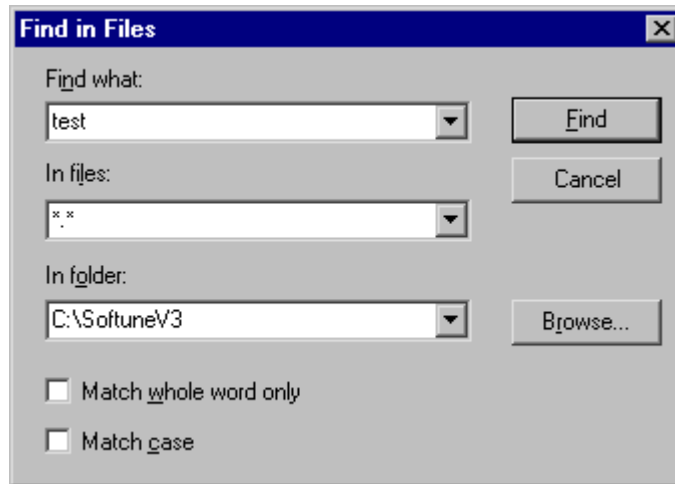
Figure 6.4-a [Find in Files] dialog box

In the [Find in Files] dialog box, the following can be set:

❍ Find string

❍ Search target file

❍ Search target directory

❍ Search in words

❍ Case sensitive

Note: When the search operation is completed, the search result is displayed on the log window.

66

6.5   Dialog Box of the [List Tag] Command

The [Tag list] dialog box lists the tags existing on the edit screen.

■ **Dialog box of the [List Tag] command**

Choose the [List Tag] command from the [Search] menu, and the [Tag list] dialog box, shown in Figure 6.5-a, appears.
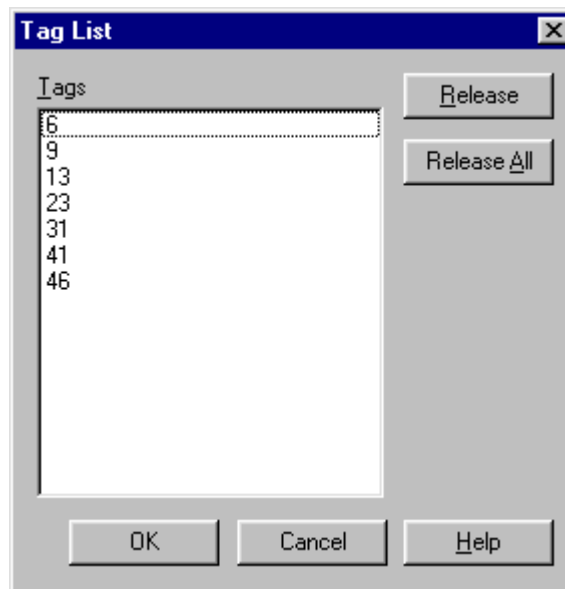
Figure 6.5-a [Tag List] dialog box

The [Tag List] dialog box contains the following:

❍ [Tags] list box

Displays the tags existing on the edit screen, together with the numbers of the lines where they are located.

❍ [Release] button

Deletes a tag from the [Tags] list box. Click the tag to be deleted with the left mouse button, then select this button.

❍ [Release all] button

Deletes all tags from the edit screen.

CHAPTER 7   View

This chapter explains the View menu.

7.1   View Menu

The commands on the View menu turn on or off the tool bars, status bar, project management, diagnostic messages, and others.
The View menu provides the following commands:

- **[Tool bars] command**
- **[Status bar] command**
- **[Project management] command**
- **[Diagnostic message] command**
- **[Check log] command**
- **[Customize…] command**

■ **View menu**

  ❍ [Tool bars] command

  Turns the tool bars on or off.

  The C Checker provides the following four tool bars:

- Standard tool bar: Contains function buttons which operate as the most frequently used commands such as [Open…].

- Check tool bar: Contains the buttons having the functions corresponding to the commands on the [Check] and other menus.

- Window tool bar: Docks subwindows and turns them on or off.

- Find tool bar: Contains the buttons having the functions corresponding to the commands on the [Find] and other menus.

  When each tool bar is displayed, a check mark appears next to the name of this command.

  See Section 7.2, "Tool Bars" for details.

  ❍ [Status bar] command

  Turns the status bar on or off. The status bar displays a brief explanation of a menu command or tool bar button when that command or button is chosen, the ON/OFF states of the special keys on the keyboard, and other information. When the status bar is displayed, a check mark appears next to the name of this command.

  See Section 7.3, "Status Bar" for details.

  ❍ [Project management] command

  Turns the project management window on or off.

  When the project management window is displayed, a check mark appears next to the name of this command.

  See Section 7.4, "Project Management" for details.

70

❍ [Diagnostic message] command

Turns the diagnostic message display window on or off.

When the diagnostic message display window is displayed, a check mark appears next to the name of this command.

See Section 7.5, "Diagnostic Messages" for details.

❍ [Check log] command

Turns the check log display window on or off.

When the check log display window is displayed, a check mark appears next to the name of this command.

See Section 7.7, "Check Log" for details.

❍ [Customize…] command

Displays the customize dialog box for setting the display attributes of a window.

Choosing this command displays the [Customize] dialog box. In the box, enter the desired settings.

See Section 7.8, "Customize" for details.

7.2   Tool Bars

A tool bar is displayed at an upper position in the application window, directly below the menu bar. It contains the tools frequently used in the C Checker, which can be operated by clicking them with the left mouse button.

■ **Tool bar**

Figure 7.2-a shows a tool bar example.

Figure 7.2-a Tool bar example

7.3　Status Bar

The status bar is displayed at the bottom of the C Checker window.
To turn the status bar on or off, choose the [Status bar] command from the [View]
menu.

■　**Status bar**

The figure below shows a status bar example.

| For Help, press F1 | Line 1 , 1 | CAP | NUM |

Figure 7.3-a Status bar example

In the left portion of the status bar, a brief explanation of a menu command is displayed when that command is chosen. Similarly, a brief explanation of a tool bar button is displayed when that button is clicked. To cancel the tool bar button after reading the explanation, move the mouse pointer to a position other than that of the tool bar button and release the mouse button.

In the right portion of the status bar, the ON/OFF states of the keys given in Table 7.3-a are displayed.

Table 7.3-a　ON/OFF states of keys that are displayed in the status bar

Indication	Description
Line1,1	Cursor position
CAP	The [Caps Lock] key is ON.
NUM	The [Num Lock] key is ON.

73

7.4   Project Management

The project management window manage the source and include files loaded into projects.
It also manages summary files in which the check status and other information are recorded.

■ **Project management window**

Figure 7.4-a shows an example of the project management window displaying source files.
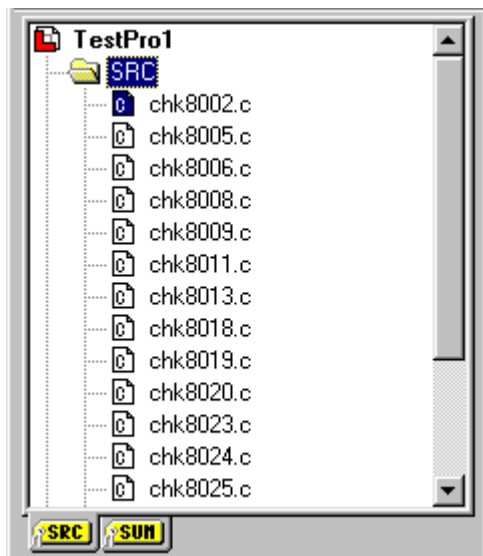Figure 7.4-b shows an example of the project management window displaying summary files.
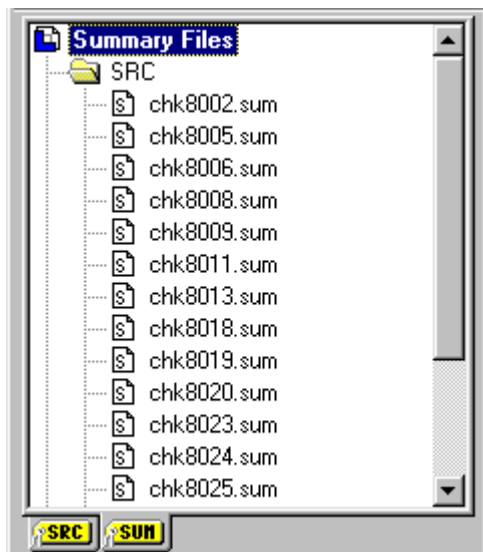

Figure 7.4-a Project management window (source files)

Figure 7.4-b Project management window (summary files)

74

As shown by Figures 7.4-a and 7.4-b, the project management window is divided into two pages:

❍ Source file management page

Used to manage source and include files.

❍ Summary file management page

Used to manage summary files.

Summary file is displayed by double clicking the left mouse button on the file name in the window of by selecting the Summary command (Figure 7.4-c)
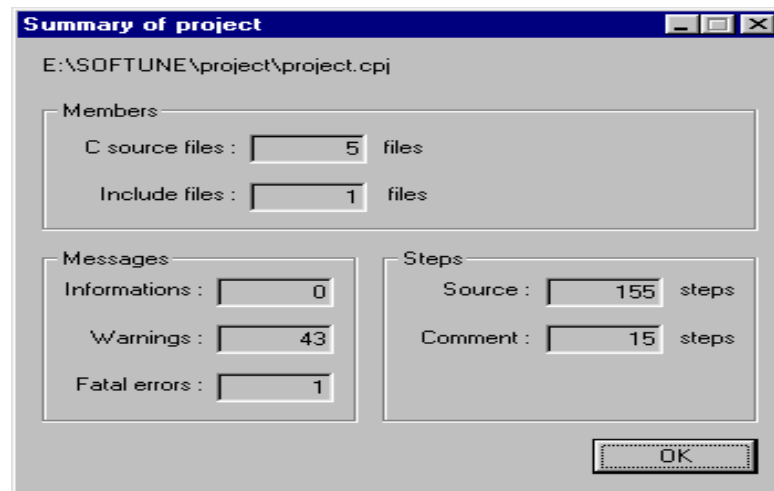
Figure 7.4-c Summary dialog box

This window also provides the following:

❍ Popup menu

The popup menu is displayed by clicking the right mouse button within the window.

The popup menu allows the following operations to be performed quickly:

• Dock: Connects a subwindow to the bottom of its parent window or disconnects it.

• Hide: Turns the window off.

❍ Popup menu on file

The popup menu on file is displayed by double clicking the right mouse button on the file name is the window.

The popup menu allows the following operations to be performed quickly:

• Open: The selected file is opened in the editor.

• Check: The selected file is checked.

• Compile: The selected file is compiled.

• Delete: The selected file is deleted from the project.

• Property: The file information on the selected file is displayed.

The editor can be started by double clicking the left mouse button on the file name in the window.

75

7.5   Diagnostic Messages

The diagnostic message window displays simple diagnostic messages.

■ **Diagnostic message window**

After completion of a check, the diagnostic message window displays simple diagnostic messages. Each diagnostic message contains a file name, line number, message code, simple message, and the directory containing the file, in this order. Figure 7.5-a shows the diagnostic message window.

File Name	Line Number	Message Code	Diagnosis Message	Path Name
chk8046x.c	13	W8046	`a' may be used before it is set	E:\FETOOL\TestSwt
chk8046x.c	17	W8934	different object size	E:\FETOOL\TestSwt
chk8046x.c	24	W8046	`i' may be used before it is set	E:\FETOOL\TestSwt
chk8046x.c	21	W8089	static function `sub' unused	E:\FETOOL\TestSwt
chk8046x.c	7	W8085	symbol `lnum' unused	E:\FETOOL\TestSwt

Figure 7.5-a Diagnostic message window

The diagnostic message window not only displays simple messages but provides the following functions:

❍ Jump to the line with a problem found

Jumps to the line with a problem found.

[Procedure]

If the file containing the line is not opened, the file is opened to jump to the line.

Double-click a file name with the left mouse button.

❍ Display a detailed message

Displays the detailed message window for displaying a detailed diagnostic message.

[Procedures]

Procedure No. 1: Click a diagnostic message with the right mouse button and select the [Detailed message] menu.

Procedure No. 2: Double-click a diagnostic message with the left mouse button and then

choose from the check tool bar.

76

❍ Popup menu

Displays the popup menu to allow the following operations to be performed quickly:
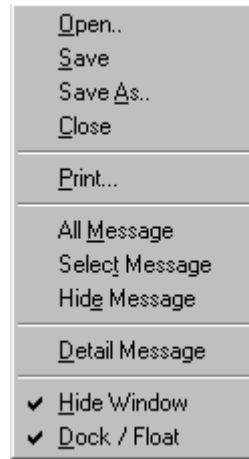

Figure 7.5-b Pop-up menu of the diagnostic message window

• Open: Open the diagnostic message file.

• Save: Save the diagnostic message into file.

• Save as: Save the diagnostic message with another name.

• Close: Close the diagnostic message.

• Print: Print the diagnostic message.

• All Message: All diagnostic message are displayed.

• Select Message: Only the message of the same message code as the code of the
selected diagnostic message is displayed.

• Hide Message: All of the message which has been [Hide Message] and all of the
message of the same message code as the code of the selected
diagnostic message are made non-display.

• Detail message: Displayed a detailed message.

• Dock / Float: Connects a subwindow to the bottom of its parent window or
disconnects it.

• Hide: Turns the window off.

[Procedure]

Click the right mouse button within the window.

7.6 Detailed Messages

The detailed message window displays detailed diagnostic messages.

■ **Detailed message window**

The detailed message window displays the detailed message corresponding to a message displayed in the diagnostic message window. Figure 7.6-a shows the detailed message window.
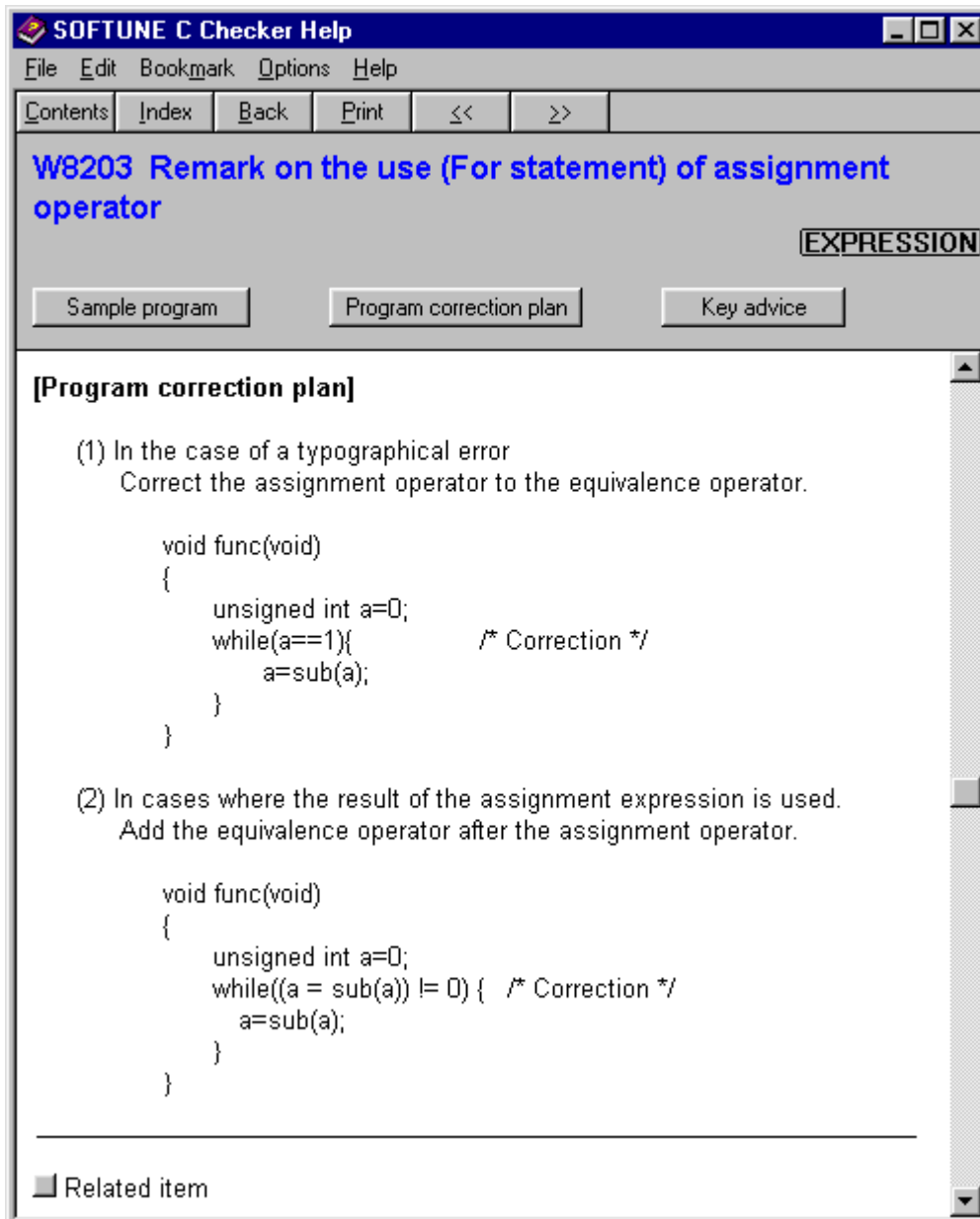
```
SOFTUNE C Checker Help                                    _ □ ✕

File  Edit  Bookmark  Options  Help

Contents │ Index │ Back │ Print │   <<   │   >>   │

W8203  Remark on the use (For statement) of assignment
operator
                                                    EXPRESSION

  Sample program          Program correction plan         Key advice

[Program correction plan]

    (1) In the case of a typographical error
        Correct the assignment operator to the equivalence operator.

            void func(void)
            {
                unsigned int a=0;
                while(a==1){              /* Correction */
                    a=sub(a);
                }
            }

    (2) In cases where the result of the assignment expression is used.
        Add the equivalence operator after the assignment operator.

            void func(void)
            {
                unsigned int a=0;
                while((a = sub(a)) != 0) {   /* Correction */
                  a=sub(a);
                }
            }

    ▄ Related item
```

Figure 7.6-a Detailed message window

78

The detailed message window uses the help window.

Besides the help functions, the detailed message window provides the following:

❍ [Sample program] button

Introduces an incorrect program example that could be output messages for a problem found.

❍ [Program correction plan] button

Introduces an example of correcting the incorrect program example.

Note: The example is merely an example. There may be other ways to correct the program example.

❍ [Key advice] button

This button is provided to introduce the information related to the diagnostic message.

Not all detailed messages have this button.

❍ [Related item] button (related to …)
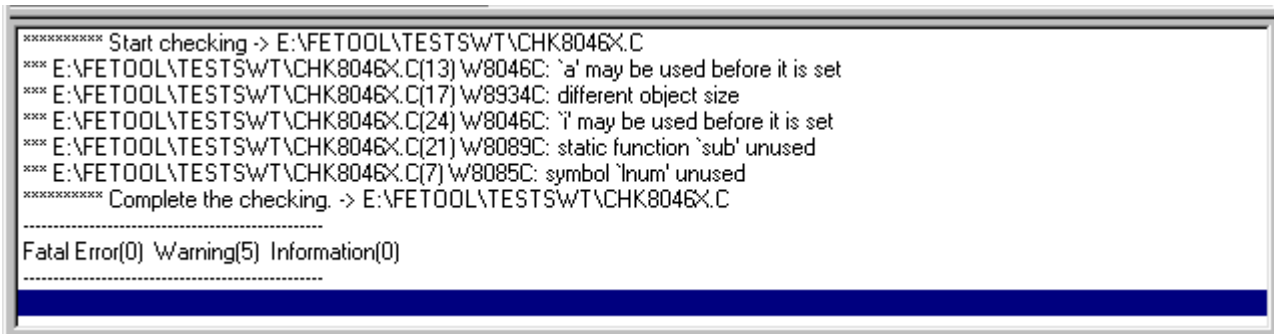
Introduces the other items related to the diagnostic message.

79

7.7   Check Log

The check log window displays the log information collected during a check.

■ **Check log window**

The check log window displays the status of the current check or compilation. Figure 7.7-a shows the check log window.

```
********** Start checking -> E:\FETOOL\TESTSWT\CHK8046X.C
*** E:\FETOOL\TESTSWT\CHK8046X.C(13) W8046C: `a' may be used before it is set
*** E:\FETOOL\TESTSWT\CHK8046X.C(17) W8934C: different object size
*** E:\FETOOL\TESTSWT\CHK8046X.C(24) W8046C: `i' may be used before it is set
*** E:\FETOOL\TESTSWT\CHK8046X.C(21) W8089C: static function `sub' unused
*** E:\FETOOL\TESTSWT\CHK8046X.C(7) W8085C: symbol `lnum' unused
********** Complete the checking. -> E:\FETOOL\TESTSWT\CHK8046X.C
------------------------------------------------
Fatal Error(0)  Warning(5)  Information(0)
------------------------------------------------
```

Figure 7.7-a Check log window

The check log window provides the following:

❍ Popup menu for performing operations quickly

Hide (turns the window off)

Dock / Float (connects a subwindow to the bottom of its parent window or disconnects it)

Save (Save a check log into a file)

Clear (Delete a check log)

[Procedure]

Click the right mouse button within the window.

7.8   Customize

The [Customize] dialog box allows the user to set the following:
- **Screen front and font size**
- **Display attribute of the editor**

■ **[Customize] dialog box**

❍ Screen font and font size

Figure 7.8-a shows the font tag of the [Customize] dialog box. This dialog box provides the following:

- Category: Select the window for which the font is to be changed.

- Font name: Select a font name.

- Size: Change the font size.

- Reset All: Resets all windows to the standard font and size.

  - The standard font and size are MS Gothic and 9 points in Japanese.

  - The standard font and size are MS Sans Serif and 8 points in English.

- Sample: Gives a preview of the selected font.
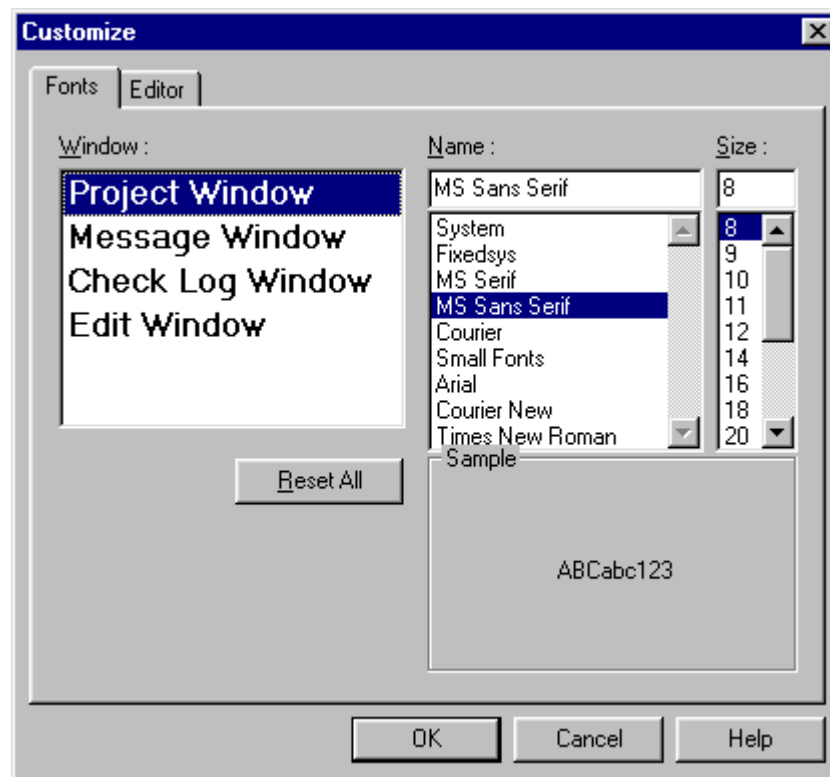
Figure 7.8-a [Customize] dialog box (font tag)

81

❍ Display attributes of the editor

Figure 7.8-b shows the editor tag of the [Customize] dialog box. This dialog box provides the following:

- Highlight the Comment: Turns the comment highlighting on or off on the edit screen.

- Show the New Line: Turns the line feed characters on or off on the edit screen.

- Show the EOF: Turns the EOF symbol on or off on the edit screen.

- Show the Ruler: Turns the ruler on or off on the edit screen.

- Show the Line Number: Turns the line numbers on or off on the edit screen.

- Use of the auto indent: Turns automatic indent on or off on the edit screen.

- Show the Tab: Turns the tabs on or off on the edit screen.

- Number of Tabs: Set the number of tabs on the edit screen.

  - The default is 4. One of 2, 4, and 8 tabs can be set.

- Key word decoration: Set the keyword highlighting.

  - In default, all are not high lighted.

  - The file can be specified about the user-defined function.
    (The keyword preserved in the specified file is only a user-defined function keyword.)
    If the file of the keyword of the REALOS is specified, the system call function can be set in the keyword.

- Colors: Set the color of keywords.
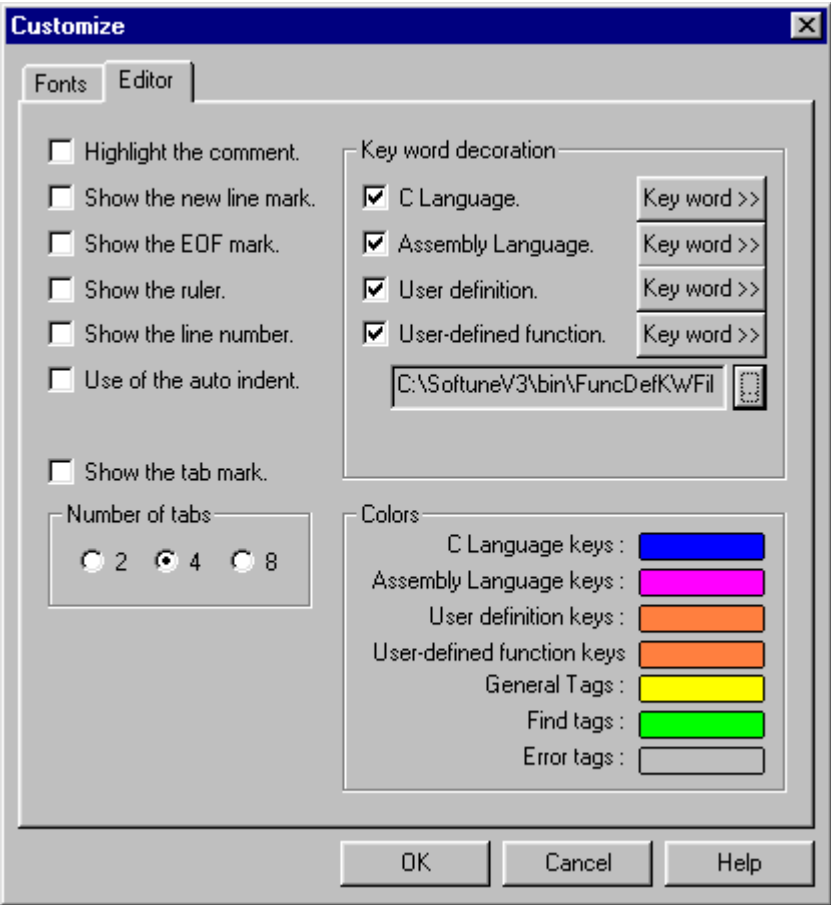
Figure 7.8-b [Customize] dialog box (editor tag)

83

❍ Setting up Key word

Choose the [Key word] command from the [Customize] dialog box (Figure 7.8-b), and the [Key words] dialog box, shown in Figure 7.8-c, appears.

In the [Key words] dialog box, the following can be set:

• Add: A new key word is registered.

• Modify: The selected key word is changed to the input key word.

• Delete: The selected key word is deleted.
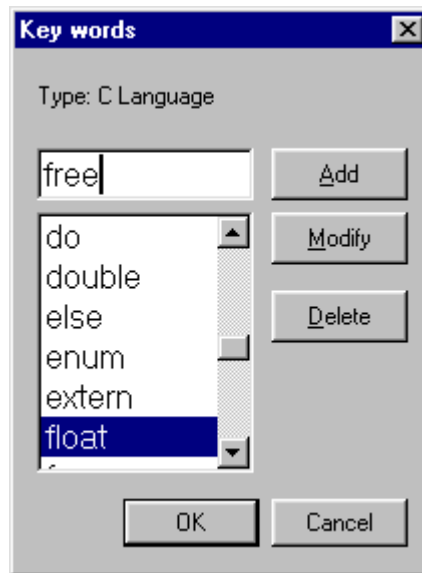
Figure 7.8-c [Key word] dialog box

CHAPTER 8   Project

This chapter explains how to create and manipulate projects.

8.1   Project Menu

The Project menu allows the user to create a new project, manipulate and save files, and perform other operations.
The Project menu provides the following commands:

- **[New] command**
- **[Open] command**
- **[Close] command**
- **[Save] command**
- **[Rename] command**
- **[Import File] command**

■ **Project menu**

❍ [New] command

Creates a new project.

Choosing this command displays the [New Project] dialog box. In the box, enter the desired project name and the name of the directory where the new project file is to be saved.

After "Create to project from SOFTUNE project file." check box is checked when Softune C Checker's project file is made from SOFTUNE Workbench's project file, SOFTUNE Workbench's project file is specified for the box under that.

❍ [Open] command

Opens an existing project.

Choosing this command displays the [Open] dialog box. In the box, specify a project file that has been created using the [New] command to open the project file.

Project files have the ".cpj" identifier.

❍ [Close] command

Closes the currently opened project file.

If the project contains a file that has been edited, choosing this command opens a confirmation dialog box asking whether the file should be saved before the project file is closed.

❍ [Save] command

Saves the contents of the currently opened project file, overwriting the previous contents.

Choosing this command does not close the project file.

To close the project file, choose the [Close] button.

86

❍ [Rename] command

Renames the currently opened project file.

Choosing this command displays the [Rename project] dialog box. In the box, enter a new project name.

When the project is renamed, the project file is also renamed.

❍ [Import file] command

Loads a file into a project.

C source and header files can be loaded.

Choosing this command displays the [Load file] dialog box. In this box, specify the name of the file to be loaded.

Multiple files can be loaded.

8.2   Creating a New Project

To create a new project, choose the [New] command from the Project menu.

■ **Creating a new project**

Choose the [New] command from the Project menu, and the [New Project] dialog box appears. In this box, enter the desired project name and the name of the directory the directory where the new project file is to be saved.

When a new project is created, a file "project-name.cpj" is created in the specified directory.

To open the same project again, specify the created project file, using the [Open] command.

■ **[New Project] dialog box**

This dialog box is used to create a new project. (See Figure 8.2-a.) It provides the following:

- Project name: Enter a new project name.

- Project folder: Specify the folder to contain the project file.

- Browse button: Displays the [Folder browse] dialog box.

- Create to project from SOFTUNE project file.: Check the create to project from SOFTUNE project file.

- SOFTUNE project file.: Specify the SOFTUNE project file.
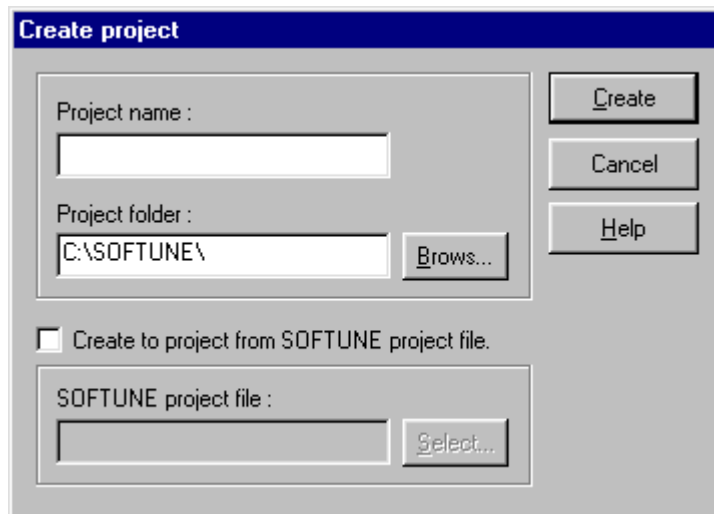
Figure 8.2-a New Project dialog box

Note: If the project name is null or contains characters that cannot be used in a project or file name, a message box appears indicating this error.

■ **[Folder browse] dialog box**

Choosing the [Browse] button in the [New Project] dialog box displays the dialog box for referencing folders. This dialog box provides the following:

- Folder: Displays the folder tree. By referencing the tree, the folder to contain a project can be specified.

- Drive: Allows a drive to be set.

- [Network] button: Allows the user to set the folder to be shared with another machine by allocating a networked drive.

■ **[Allocate networked drive] dialog box**

Choosing the [Network] button from the [Folder browse] dialog box displays the [Allocate networked drive] dialog box. This dialog box provides the following:

- Drive: Specify the drive to be allocated to a network.

- Path: Displays already shared path names.

- Reconnect at logon: Automatically allocates a specified networked drive when the system is started.

  - The default is off.

Note: When a project is created on a network, Reconnect at logon must be turned on.

8.3 Renaming Projects

The [Rename Project] dialog box allows the user to rename the currently edited project.

■ **Renaming a project**

Figure 8.3-a shows the dialog box for renaming a project. This dialog box provides the following:

- Old name: Displays the name of the currently opened project file.
- New name: Enter a new project name.
- [Rename] button: Renames the project name.
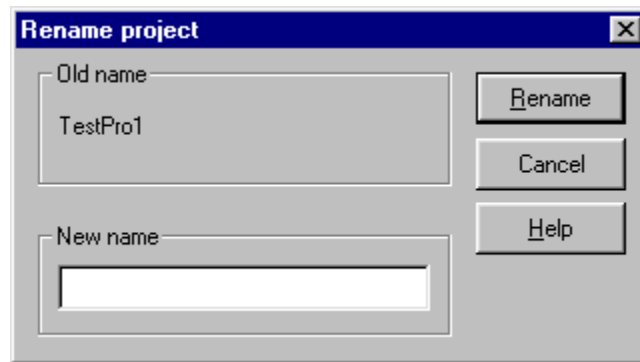
Figure 8.3-a Dialog box for renaming projects

90

8.4   Loads a File into Project

The [Import File] dialog box allows the user to Loads a file into project.

■ **Import file**

Loads a file into a project.

Choosing the [Import File] command displays the [Import] dialog box (Figure 8.4-a). In this box, specify the name of the file to be loaded.

Multiple files can be loaded.

This dialog box provides the following:

- Look in: The folder name is displayed.

- File name: The selected file name is displayed.

- Files of type: Select the displayed file type.

- Import: The selected file is loaded into the project.
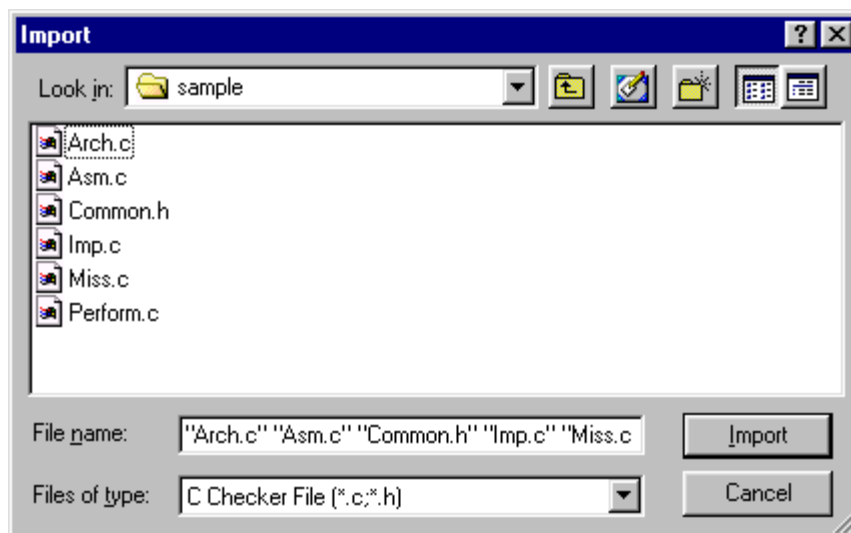
Figure 8.4-a [Import] dialog box

CHAPTER 9   Message

This chapter explains how to view and manipulate diagnostic messages.

9.1   Message Menu

The Message menu allows the user to manipulate and save diagnostic message files and perform other operations.
The Message menu provides the following commands:

- **[Open] command**
- **[Save] command**
- **[Save as] command**
- **[Close] command**
- **[Print] command**
- **[Detail message] command**

■ **Message menu**

❍ [Open] command

Opens an existing diagnostic message file.

Choosing this command displays the dialog box for opening a diagnostic message file. In the box, specify the diagnostic message file to be opened.

A diagnostic message file is a file into which the messages displayed in the diagnostic message window after the completion of a diagnostic check are saved using the [Save] or [Save as] command, and which has the ".cmg" identifier.

❍ [Save] command

Saves the contents of the currently open diagnostic message file, overwriting the previous contents.

❍ [Save as] command

Saves the currently open diagnostic messages with another file name.

Choosing this command displays the Save as dialog box. In this box, specify the name of the file used to save the diagnostic message file.

The file used to save the diagnostic message file must have the ".cmg" identifier.

❍ [Close] command

Close the contents of the currently open diagnostic message file, overwriting the previous contents.

❍ [Print] command

Prints the contents of the currently open diagnostic message file, overwriting the previous contents.

❍ [Detail message] command

Displays the detailed message corresponding to the diagnostic message selected from the diagnostic message window.

Click the name of a file displayed in the diagnostic message window with the left mouse button to select it and then choose the [Detailed message] command or choose the [Display detailed message] button from the tool box.

9.2   Detailed Messages

The detailed message window displays the detailed message corresponding to a diagnostic message displayed in the diagnosis message window.

■ **Detailed messages**

To view the detailed message window, use the procedure described below:

1) Select the desired file and view the contents.

2) In the diagnosis message window, click the file name portion or subitem in the desired message with the left mouse button.

3) Execute any of three of the following, and the detailed message is displayed.

- Choose the [Detail message] command from the [Message] menu. (See Figure 9.2-a)

- Choose the 🔍 button from the check too bar.

- Choose the [Detail message] command of the pop-up menu in the diagnosis message window. (See Figure 9.2-b)
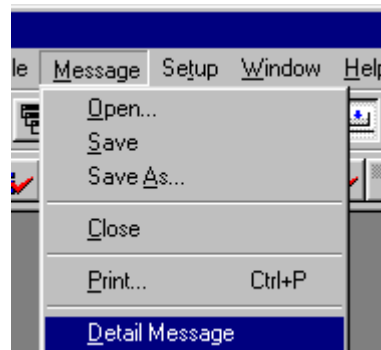
Figure 9.2-a [Message] menu
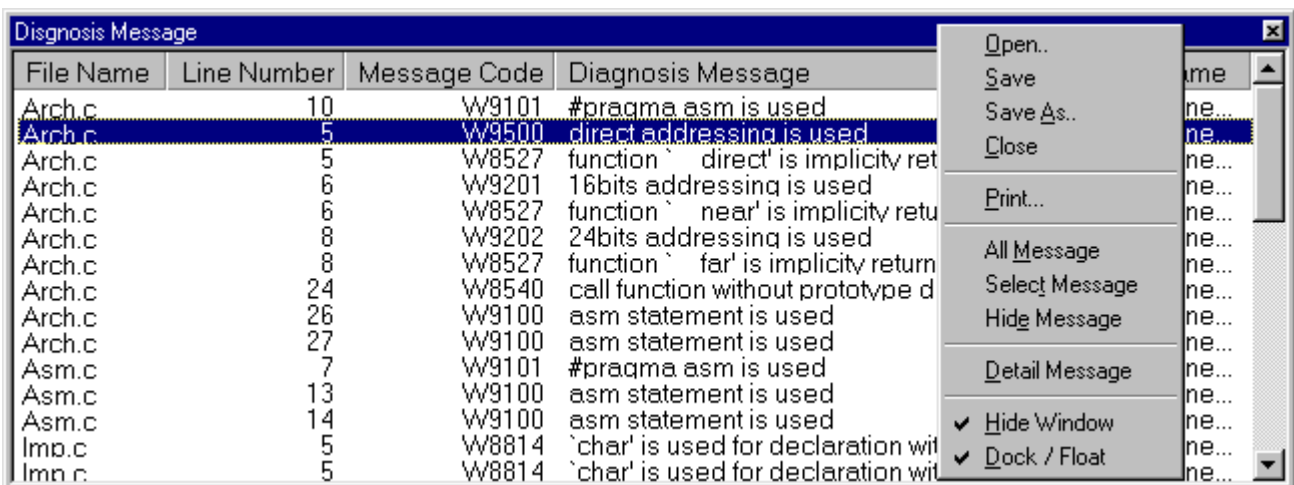
Figure 9.2-b Diagnosis message window pop-up menu

95

CHAPTER 9 Message

CHAPTER 10   Compile

This chapter explains how to compile programs.

10.1 Compile Menu

The Compile menu allows the user to compile the file being edited or the file selected from a project.
The Compile menu provides the following commands:
- **[Active Document] command**
- **[Selected Files] command**
- **[All Files] command**
- **[Stop] command**
- **[Compile Option] command**

■ **Compile menu**

Figure 10.1-a shows the Compile menu.
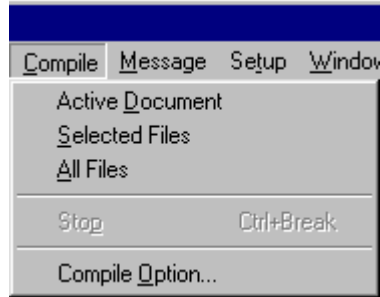
Figure 10.1-a Compile menu

❍ [Active Document] command

Compiles the active file, which is currently being edited.

❍ [Selected Files] command

Compiles the files selected from the project management window.

From the project management window, select a file as described below:

- To select a single file: Click it with the left mouse button.

- To select multiple files: Press and hold down the Control key and click them with the left mouse button; or press and hold down the Shift key and click them with the left mouse button.

❍ [All Files] command

Compiles all the files in the project management window.

❍ [Stop] command

Stops the compilation in progress.

❍ [Compiler Option] command

Set the compiler options.

10.2 Setting up fcc911s Compiler Options

This section explains how to set up the fcc911s compiler options.

■ **Dialog box for setting up the fcc911s compiler options**

Choose [Compile Option] command from the [Compile] menu or [Compile Option] command from the pop-up menu in the project management window, the [Compile Option] dialog box is displayed. (See Figure 10.2-a)
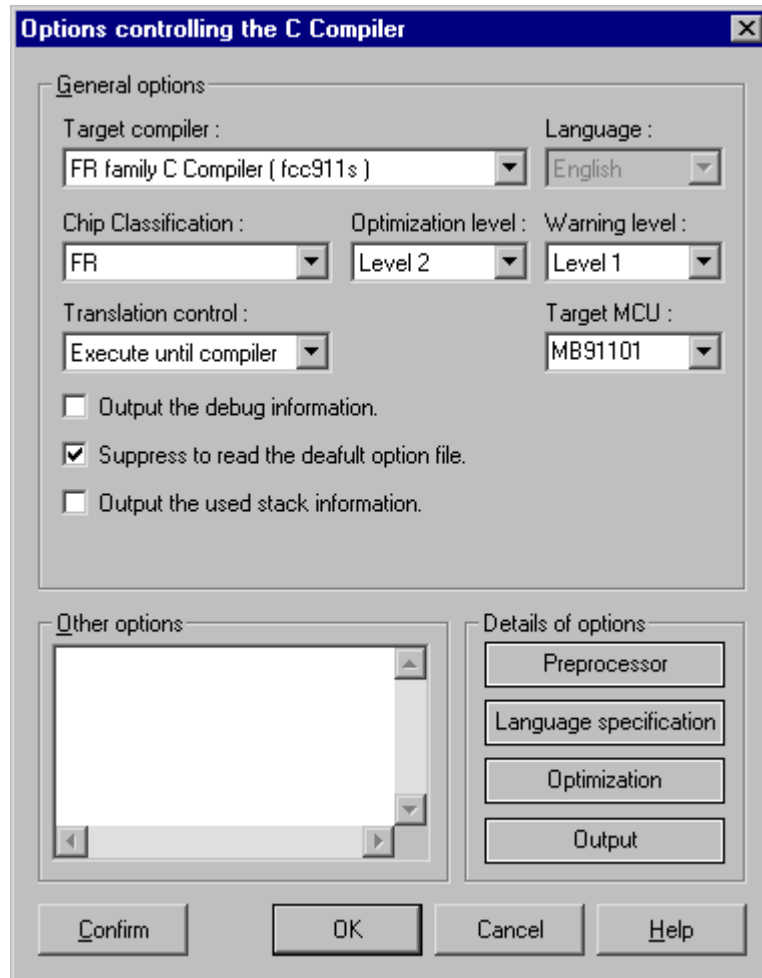
Figure 10.2-a Dialog box for setting up fcc911s compiler options

The compiler option dialog box provides the following:

❍ Target compiler

The target compiler is set in following compiler.

- fcc911s C Compiler for FR family

- fcc907s C Compiler for F^{2}MC family

- fcc896s C Compiler for F^{2}MC family

- ANSI C Compiler

99

❍ Chip classification

Select the chip type according to the MCU to be used.

The chip type that can be selected depends on the compiler type.

❍ Optimization level

Optimization allows setting of one of seven optimization levels.

The default is Level-2.

- None: Performs no optimization.

- Level-1: Performs level-1 optimization.

- Level-2: Performs level-2 optimization.

- Level-3: Performs level-3 optimization.

- Level-4: Performs level-4 optimization.

- Speed-preferred: Performs speed-preferred optimization.

- Size-preferred: Performs size-preferred optimization.

❍ Warning level

Warning allows setting of one of nine levels.

The default is Level-1.

- Level-0: Outputs no warning messages.

- Level-1: Outputs level-1 warning messages.

- Level-2: Outputs level-2 warning messages.

- Level-3: Outputs level-3 warning messages.

- Level-4: Outputs level-4 warning messages.

- Level-5: Outputs level-5 warning messages.

- Level-6: Outputs level-6 warning messages.

- Level-7: Outputs level-7 warning messages.

- Level-8: Outputs level-8 warning messages.

❍ Translation control

Translation control has two settings.

The default is configured as Execute up to Compilation.

- Execute up to preprocessing: Performs translation up to preprocessing.

- Execute up to Compilation: Performs translation up to compilation.

❍ Target MCU

Please set the MB number which the user uses in the Target MCU.

In default, the MB91101 type is set.

100

❍ Output the debug information

Specify whether to output the debug information.

The default is off.

❍ Suppress to read the default option file

Specify whether to suppress the reading of the default option file.

The default is on.

❍ Output the used stack information

Specify whether to output the used stack information file.

The default is off.

❍ Details of options

- Preprocessor: Set up the detailed options for the preprocessor.

- Language Specification: Set up the detailed options for the language specifications.

- Optimization: Set up the detailed options for optimization.

- Output: Set up the detailed options for output information.

❍ Other options

Allows setting of compiler-related options other than the above.

❍ Buttons

- [Confirm] button: Checks the settings of the options.

- [OK] button: Makes the settings effective.

- [Cancel] button: Cancels the changes.

- [Help] button: Displays the help file for the dialog box.

10.2.1 Detailed options for the fcc911s preprocessor

Set up the detailed compiler options related to the preprocessor of the fcc911s compiler. There are two dialog boxes, one for setting up include paths and the other for setting macro definitions. Set up the necessary options in both dialog boxes.

■ **Dialog box for setting up the detailed options (include paths) related to the preprocessor**

Choose [Preprocessor] command from the [Compile option] dialog box, the [Preprocessor] dialog box is displayed. Figure 10.2-b shows the dialog box for setting up Include Path when select [Include path] tab which exists in the [Preprocessor] dialog box.
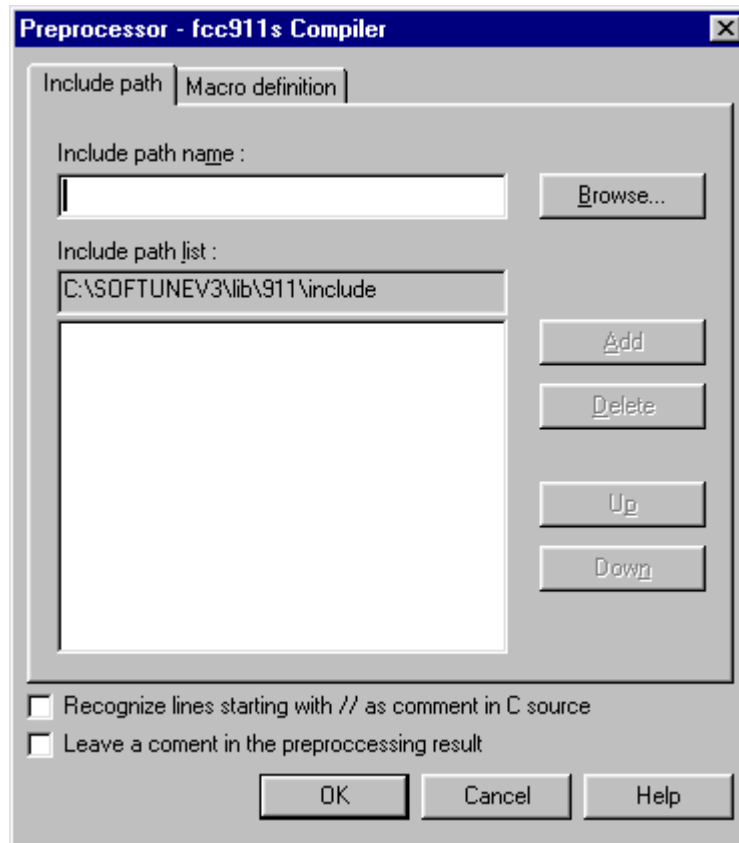
Figure 10.2-b Dialog box for setting up include paths

The dialog box for setting up include paths provides the following:

❍ Include path name field

Enter the include path used during compilation.

❍ Include path list

Displays the registered include paths.

The registered include paths are passed to the preprocessor, starting from the uppermost one.

❍ [Browse] button

Displays the dialog box for selecting an existing include path.

The path selected from the dialog box is displayed in the include path entry field.

❍ [Add] button

Adds the include path entered in the include path entry field to the include path register space.

❍ [Delete] button

Deletes an include path from the include path register space.

❍ [Up] button

Moves an include path registered in the include path register space, up one line.

❍ [Down] button

Moves an include path registered in the include path register space, down one line.

❍ "Recognize lines starting with // as comment in C source" check box

Select this option to recognize lines starting with // as comment in preprocessor.

When this option is set, this setting is applied to both the check and the compilation.

The default is off.

❍ "Leave a comment in the preprocessing result" check box.

Allows the user to leave a comment in the preprocessing results.

The default is off.

103

■ **Dialog box for setting up the detailed options (macro definitions) related to the preprocessor**

Figure 10.2-c shows the dialog box for setting up Macro definition when select [Macro definition] tab which exists in the [Preprocessor] dialog box.
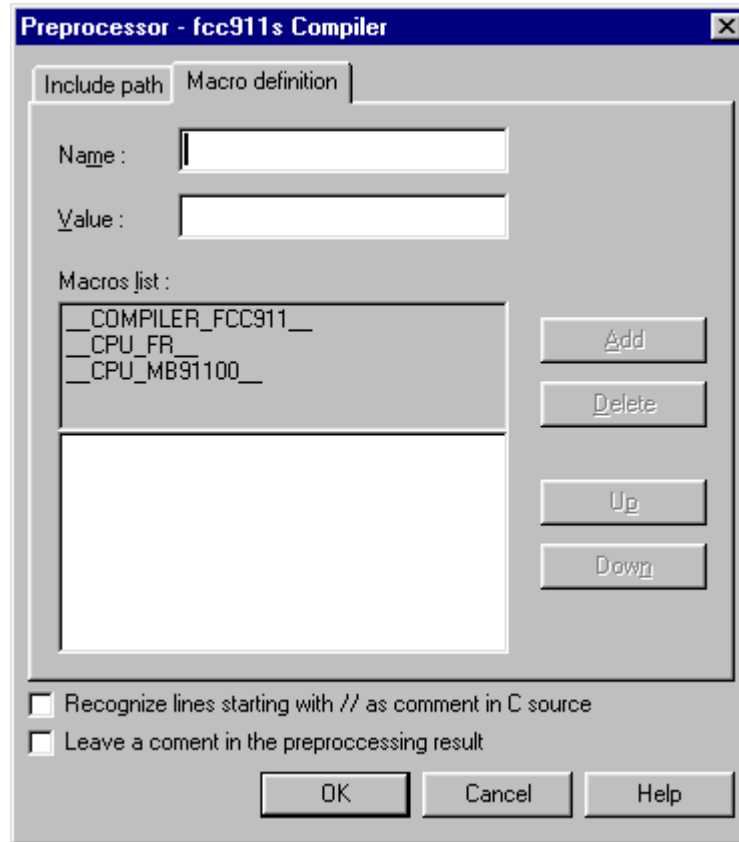
Figure 10.2-c Dialog box for setting up macro definitions

The dialog box for setting up macro definitions provides the following:

❑ Macro name field

Enter the macro to be registered during compilation.

❑ Macro value field

Enter the value of macro.

❑ Macros list

Displays the registered macros.

The registered macros are passed to the preprocessor, starting from the uppermost one.

❑ [Add] button

Adds the macro entered in the macro entry field to the macro register space.

❑ [Delete] button

Deletes a macro from the macro register space.

❍ [Up] button

Moves a macro registered in the macro register space, up one line.

❍ [Down] button

Moves a macro registered in the macro register space, down one line.

❍ "Recognize lines starting with // as comment in C source" check box

Select this option to recognize lines starting with // as comment in preprocessor.

When this option is set, this setting is applied to both the check and the compilation.

The default is off.

❍ "Leave a comment in the preprocessing result" check box.

Allows the user to leave a comment in the preprocessing results.

The default is off.

105

10.2.2 Detailed options related to the fcc911s language specifications

The following explains how to set up the detailed options related to the language specifications of the fcc911s compiler.

■ **Dialog box for setting up the detailed options related to the language specifications**

Choose [Language specifications] command from the [Compile option] dialog box, the [options related to the language specifications] dialog box is displayed. (See Figure 10.2-d)
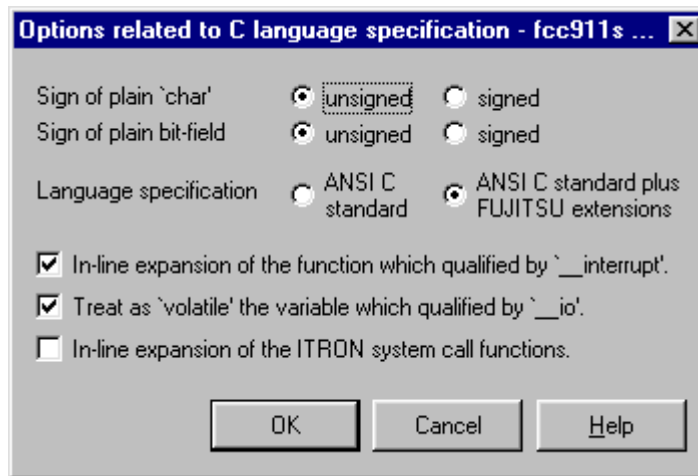
Figure 10.2-d Dialog box for setting up the options related to the language specifications

The dialog box for setting up the options related to the language specifications provides the following:

❍ Sign of plain 'char'

Specify whether the data type of char is signed or unsigned.

The default is unsigned.

❍ Sign of plane bit field

Specify whether the data type of int is signed or unsigned with the bit fields.

The default is unsigned.

❍ Language specification

Set the language specification level.

The default is ANSI C standard plus FUJITSU extensions

❍ In-line expansion of the function which qualified by '_interrupt'

Specify whether to perform in-line expansion of the function which qualified by '_interrupt'.

The default is on.

❍ Treat as 'volatile' the variable which qualified by '_io'

Specify whether to treat as 'volatile' the variable which qualified by '_io'.

The default is on.

106

❍ In-line expansion of the ITRON system call functions

Specify whether to perform in-line expansion of the ITRON system call functions.

The default is off.

10.2.3 Detailed optimization options of the fcc911s compiler

Set up the detailed (optimization) options of the fcc911s compiler.

■ **Dialog box for setting up the optimization options**

Choose [Optimization] command from the [Compile option] dialog box, the [Options controlling optimization] dialog box is displayed. (See Figure 10.2-e).
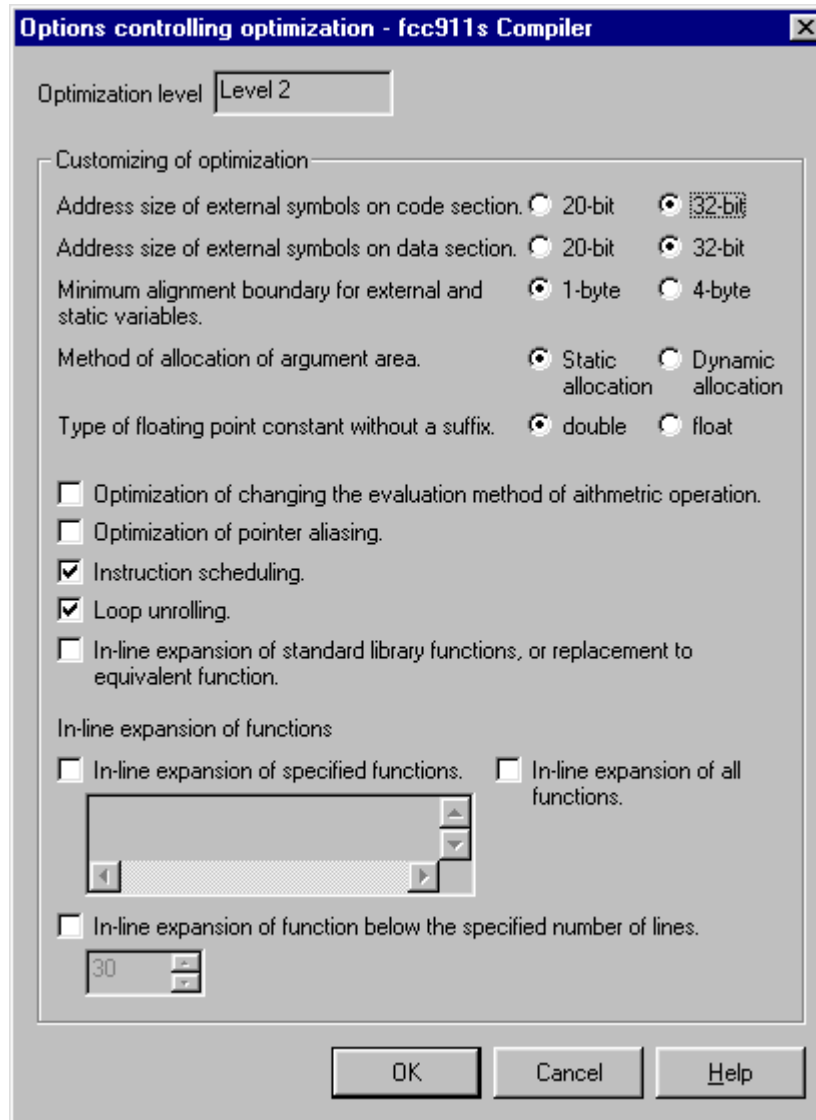
Figure 10.2-e Dialog box for setting up optimization options

The dialog box for setting up the optimization options provides the following:

❍ Address size of external symbols on code section

Set the address size of the external symbols on the code section.

The default is 32 bits.

❍ Address size of external symbols on data section

Set the address size of the external symbols on the data section.

The default is 32 bits.

❍ Minimum alignment boundary for external and static variables

Set the minimum alignment boundary for external and static variables.

The default is one byte.

❍ Method of allocation of argument area

Specify whether to allocate argument area statically or dynamically.

The default is static allocation.

❍ Type of floating point constant without a suffix

Set the type of floating point constant without a suffix.

The default is double.

❍ Optimization of changing the evaluation method of arithmetic operation

Specify whether to perform optimization of changing the evaluation method of arithmetic operation.

The default is off.

❍ Optimization of pointer aliasing

Specify whether to perform optimization of pointer aliasing.

The default is off.

❍ Instruction scheduling

Specify whether to perform instruction scheduling.

The default is on.

❍ Loop unrolling

Specify whether to perform loop unrolling.

The default is on.

❍ In-line expansion of standard library functions or replacement to equivalent function

Specify whether to perform in-line expansion of standard library functions or replacement to equivalent function

The default is off.

❍ In-line expansion of specified functions

Place a check mark in the check box and specify the functions subject to in-line expansion.

The default is off.

❍ In-line expansion of function below the specified number of lines

Place a check mark in the check box and specify the number of the line below which in-line expansion is to be performed.

The default is off.

10.2.4 Detailed options related to fcc911s output

The following explains how to set up the options related to the output of the fcc911sfcc907ss compiler.

■ **Dialog box for setting up the detailed options related to output**

Choose [Output] command from the [Compile option] dialog box, the [options related to output object] dialog box is displayed. (See Figure 10.2-f)
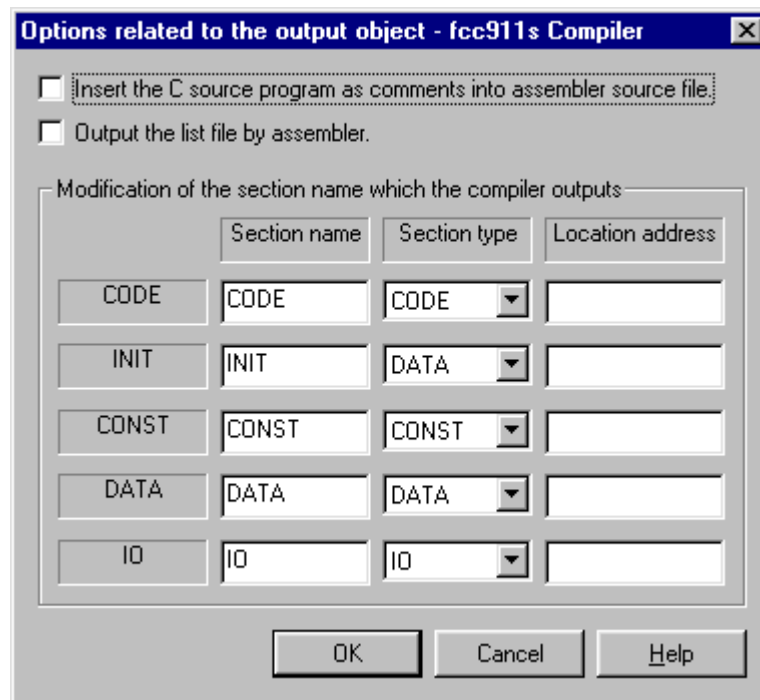
Figure 10.2-f Dialog box for setting up the options related to output

The dialog box for setting up the options related to output provides the following:

❍ Insert the C source program as comments into assembler source file

Specify whether to insert the C source program as comments into an assembler source file (*.asm).

The default is off.

❍ Output the list file by assembler

Specify whether to output an assembly list.

The default is off.

❍ Modification of the section name which the compiler outputs

The sections to be output by the compiler can be changed.

Section names, section types, and location addresses can be entered.

111

10.3   Setting up fcc907s Compiler Options

This section explains how to set up the fcc907s compiler options.

■ **Dialog box for setting up the fcc907s compiler options**

Choose [Compile Option] command from the [Compile] menu or [Compile Option] command from the pop-up menu in the project management window, the [Compile Option] dialog box is displayed. (See Figure 10.3-a)
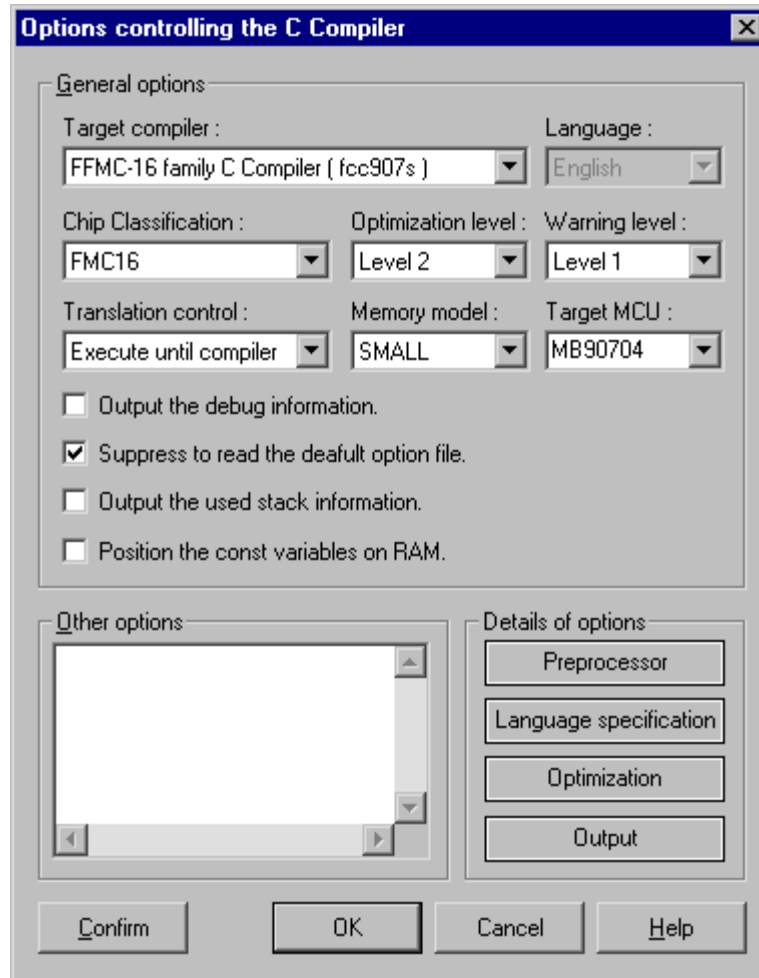
Figure 10.3-a Dialog box for setting up fcc907s compiler options

The fcc907s compiler option dialog box provides the following:

❍ Target compiler

The target compiler is set in following compiler.

- fcc911s C compiler for the FR family.

- fcc907c C compiler for the F^{2}MC family.

- fcc896s C compiler for the F^{2}MC family.

- ANSI C compiler

112

❍ Chip classification

Select the chip type according to the MCU to be used.

The chip type that can be selected depends on the compiler type.

❍ Optimization level

Optimization allows setting of one of seven optimization levels.

The default is Level-2.

- None: Performs no optimization.
- Level-1: Performs level-1 optimization.
- Level-2: Performs level-2 optimization.
- Level-3: Performs level-3 optimization.
- Level-4: Performs level-4 optimization.
- Speed-preferred: Performs speed-preferred optimization.
- Size-preferred: Performs size-preferred optimization.

❍ Warning level

Warning allows setting of one of nine levels.

The default is Level-1.

- Level-0: Outputs no warning messages.
- Level-1: Outputs level-1 warning messages.
- Level-2: Outputs level-2 warning messages.
- Level-3: Outputs level-3 warning messages.
- Level-4: Outputs level-4 warning messages.
- Level-5: Outputs level-5 warning messages.
- Level-6: Outputs level-6 warning messages.
- Level-7: Outputs level-7 warning messages.
- Level-8: Outputs level-8 warning messages.

❍ Translation control

Translation control has two settings.

The default is Execute until Compiler.

- Execute until preprocessing: Performs translation until preprocessing.
- Execute until Compiler: Performs translation until compilation.

113

❍ Memory model

Memory model allows setting of one of the following four options.

The default is SMALL.

- SMALL: Uses the SMALL memory model.

- MEDIUM: Uses the MEDIUM memory model.

- COMPACT: Uses the COMPACT memory model.

- LARGE: Uses the LARGE memory model.

❍ Target MCU

In the Target MCU field, enter the MB number used.

The default is MB90704.

❍ Output the debug information

Specify whether to output the debug information.

The default is off.

❍ Suppress to read the default option file

Specify whether to suppress the reading of the default option file.

The default is on.

❍ Output the used stack information

Specify whether to output the used stack information file.

The default is off.

❍ Position the const variables on RAM

Specify whether to position const variables in RAM.

The default is off.

❍ Details of options

- Preprocessor: Set up the detailed options for the preprocessor.

- Language Specification: Set up the detailed options for the language specifications.

- Optimization: Set up the detailed options for optimization.

- Output: Set up the detailed options for output information.

❍ Other options

Allows setting of compiler-related options other than the above.

❍ Buttons

- [Confirm] button: Checks the settings of the options.

- [OK] button: Makes the settings effective.

- [Cancel] button: Cancels the changes.

- [Help] button: Displays the help file for the dialog box.

114

10.3.1 Detailed options for the fcc907s preprocessor

The following explains how to set up the detailed compiler options related to the preprocessor of the fcc907s compiler. There are two dialog boxes, one for setting up include paths and the other for setting macro definitions. Set up the necessary options in both dialog boxes.

■ **Dialog box for the detailed options (include paths) related to the preprocessor**

Choose [Preprocessor] command from the [Compile option] dialog box, the [Preprocessor] dialog box is displayed. Figure 10.3-b shows the dialog box for setting up Include Path when select [Include path] tab which exists in the [Preprocessor] dialog box.
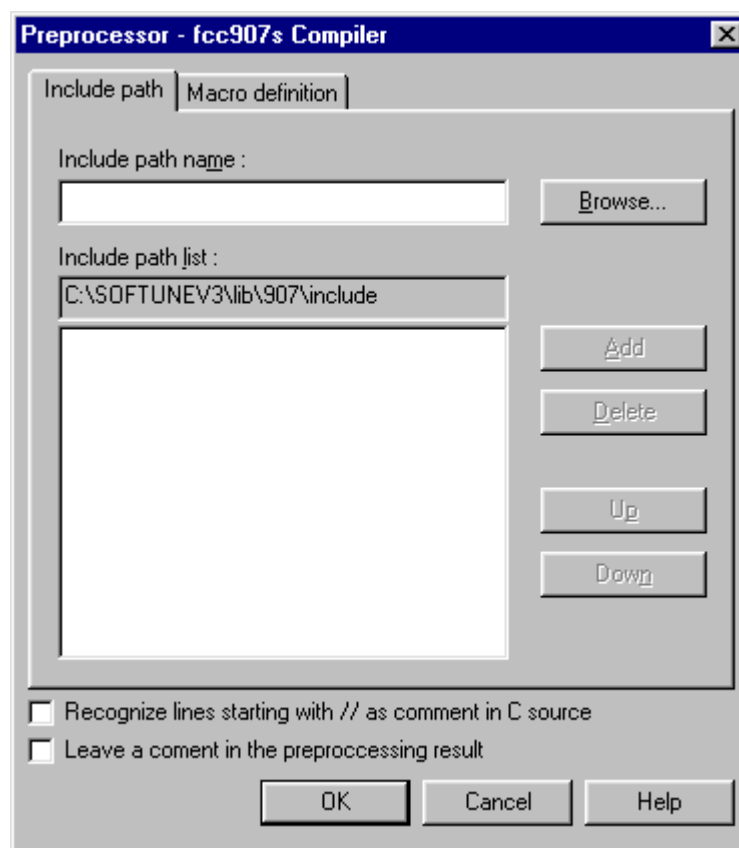
Figure 10.3-b Dialog box for setting up include paths

The dialog box for setting up include paths provides the following:

❍ Include path name field

Enter the include path used during compilation.

❍ Include path list space

Displays the registered include paths.

The registered include paths are passed to the preprocessor, starting from the uppermost one.

115

❍ [Browse] button

Displays the dialog box for selecting an existing include path.

The path selected from the dialog box is displayed in the include path entry field.

❍ [Add] button

Adds the include path entered in the include path entry field to the include path register space.

❍ [Delete] button

Deletes an include path from the include path register space.

❍ [Up] button

Moves an include path registered in the include path register space, up one line.

❍ [Down] button

Moves an include path registered in the include path register space, down one line.

❍ "Recognize lines starting with // as comment in C source" check box

Select this option to recognize lines starting with // as comment in preprocessor.

When this option is set, this setting is applied to both the check and the compilation.

The default is off.

❍ "Leave a comment in the preprocessing result" check box.

Allows the user to leave a comment in the preprocessing results

The default is off.

■ **Dialog box for the detailed options (macro definitions) related to the preprocessor**

Figure 10.3-c shows the dialog box for setting up Macro definition when select [Macro definition] tab which exists in the [Preprocessor] dialog box.
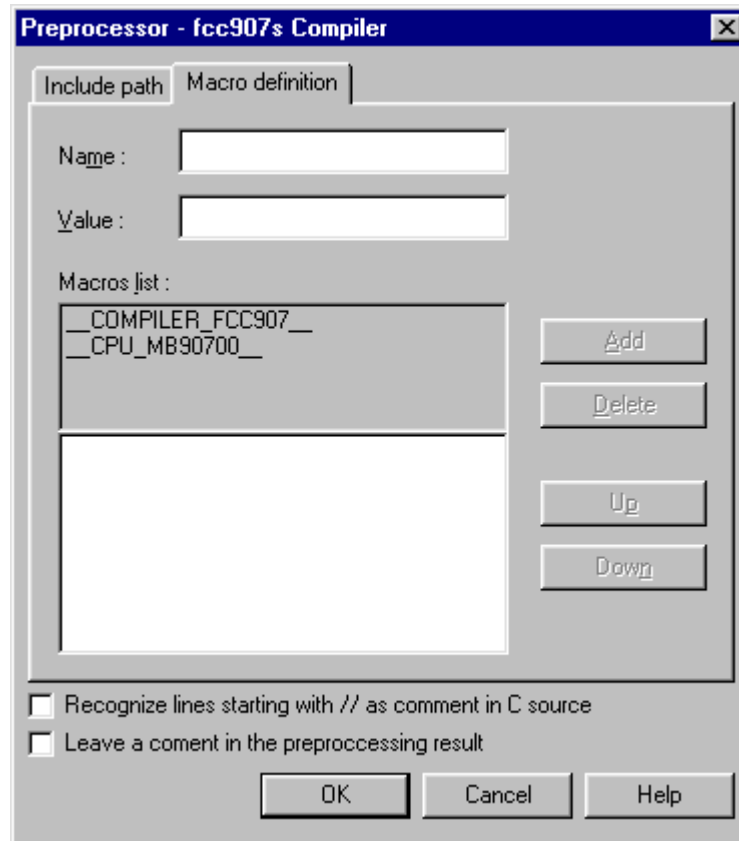
Figure 10.3-c Dialog box for setting up macro definitions

The dialog box for setting up macro definitions provides the following:

❍ Macro entry field

Enter the macro to be registered during compilation.

❍ Macro value field

Enter the value of macro.

❍ Macros list

Displays registered macros.

The registered macros are passed to the preprocessor, starting from the uppermost one.

❍ [Add] button

Adds the macro entered in the macro entry field to the macro register space.

❍ [Delete] button

Deletes a macro from the macro register space.

117

❍ [Up] button

Moves a macro registered in the macro register space, up one line.

❍ [Down] button

Moves a macro registered in the macro register space, down one line.

❍ "Recognize lines starting with // as comment in C source" check box

Select this option to recognize lines starting with // as comment in preprocessor.

When this option is set, this setting is applied to both the check and the compilation.

The default is off.

❍ "Leave a comment in the preprocessing result" check box.

Allows the user to leave a comment in the preprocessing results.

The default is off.

10.3.2 Detailed options related to the fcc907s language specifications

The following explains how to set up the detailed options related to the language specifications of the fcc907s compiler.

■ **Dialog box for the detailed options related to the language specifications**

Choose [Language specifications] command from the [Compile option] dialog box, the [options related to the language specifications] dialog box is displayed. (See Figure 10.3-d)
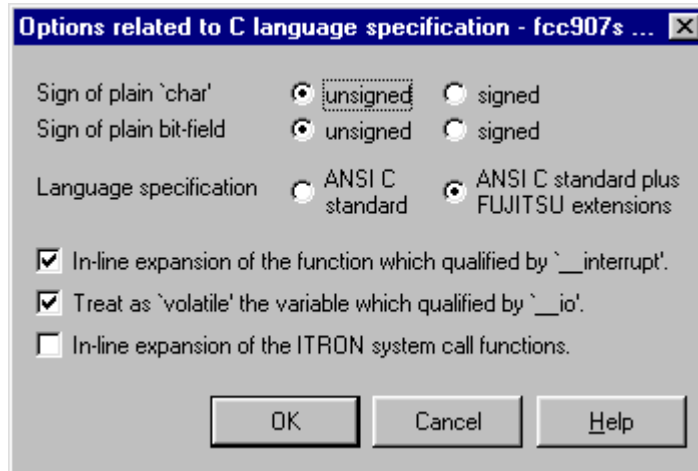
Figure 10.3-d Dialog box for setting up the options related to the language specifications

The dialog box for setting up the options related to the language specifications provides the following:

❍ Sign of plain 'char'

Specify whether the data type of char is signed or unsigned.

The default is unsigned.

❍ Sign of plane bit field

Specify whether the data type of int is signed or unsigned with the bit fields.

The default is unsigned.

❍ Language specification

Set the language specification level.

The default is ANSI C standard plus FUJITSU extensions

❍ In-line expansion of the function which qualified by '_interrupt'

Specify whether to perform in-line expansion of the function which qualified by '_interrupt'.

The default is on.

❍ Treat as 'volatile' the variable which qualified by '_io'

Specify whether to treat as 'volatile' the variable which qualified by '_io'.

The default is on.

119

❍ In-line expansion of the ITRON system call functions

Specify whether to perform in-line expansion of the ITRON system call functions.

The default is off.

10.3.3 Detailed optimization options of the fcc907s compiler

The following explains how to set up the detailed optimization options of the fcc907s compiler.

■ **Dialog box for setting up the optimization options**

Choose [optimization] command from the [Compile option] dialog box, the [options controlling optimization] dialog box is displayed. (See Figure 10.3-e)
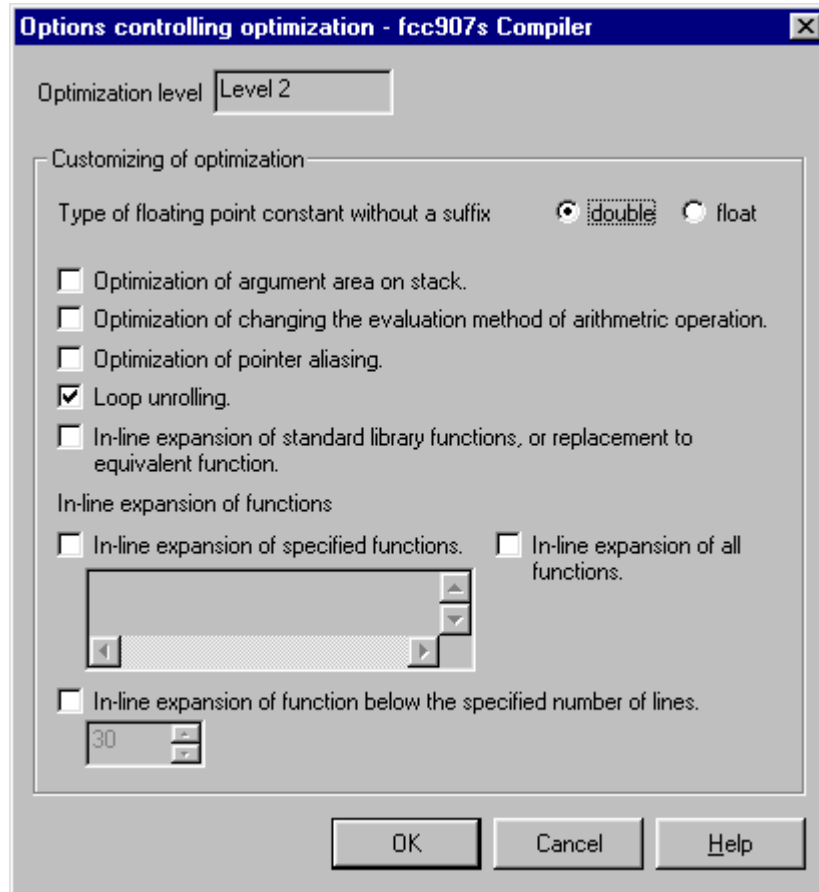
Figure 10.3-e Dialog box for setting up optimization options

The dialog box for setting up the optimization options provides the following:

❍ Minimum alignment boundary for external and static variables

Set the minimum alignment boundary for external and static variables.

The default is one byte.

❍ Type of floating point constant without a suffix

Set the type of floating point constant without a suffix.

The default is double.

❍ Optimization of argument area on stack

Specify whether to optimize the argument area.

The default is off.

❍ Optimization of changing the evaluation method of arithmetic operation

Specify whether to perform optimization of changing the evaluation method of arithmetic operation.

The default is off.

❍ Optimization of pointer aliasing

Specify whether to perform optimization of pointer aliasing.

The default is off.

❍ Loop unrolling

Specify whether to perform loop unrolling.

The default is on.

❍ In-line expansion of standard library functions or replacement to equivalent function

Specify whether to perform in-line expansion of standard library functions or replacement to equivalent function.

The default is off.

❍ In-line expansion of specified functions

Place a check mark in the check box and specify the functions subject to in-line expansion.

❍ In-line expansion of function below the specified number of lines

Place a check mark in the check box and specify the number of the line below which in-line expansion is to be performed.

10.3.4 Detailed options related to fcc907s output

The following explains how to set up the options related to the output of the fcc907s compiler.

■ **Dialog box for setting up the detailed options related to output**

Choose [Output] command from the [Compile option] dialog box, the [options related to output object] dialog box is displayed. (See Figure 10.3-f)
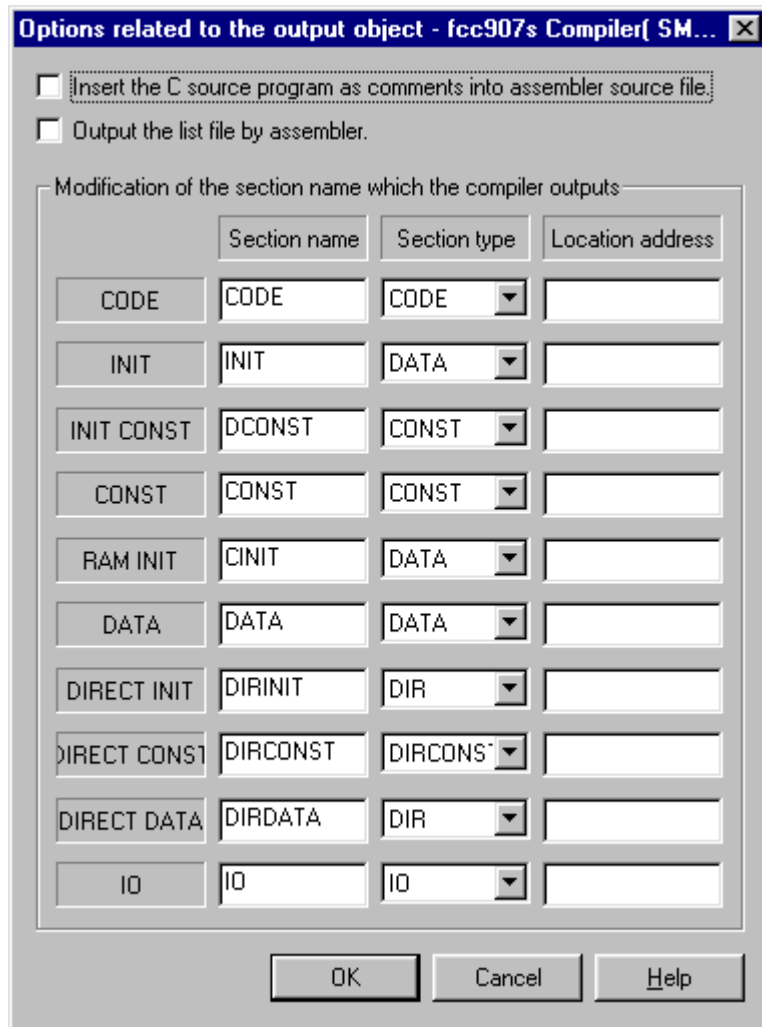
Figure 10.3-f Dialog box for setting up the options related to output

123

The dialog box for setting up the options related to output provides the following:

❍ Insert the C source program as comments into assembler source file

Specify whether to insert the C source program as comments into an assembler source file (*.asm).

The default is off.

❍ Output the list file by assembler

Specify whether to output an assembly list.

The default is off.

❍ Modification of the section name which the compiler outputs

The sections to be output by the compiler can be changed.

Section names, section types, and location addresses can be entered.

10.4 Setting up fcc896s Compiler Options

This section explains how to set up the fcc896s compiler options.

■ **Dialog box for setting up the fcc896s compiler options**

Choose [Compile Option] command from the [Compile] menu or [Compile Option] command from the pop-up menu in the project management window, the [Compile Option] dialog box is displayed. (See Figure 10.4-a)
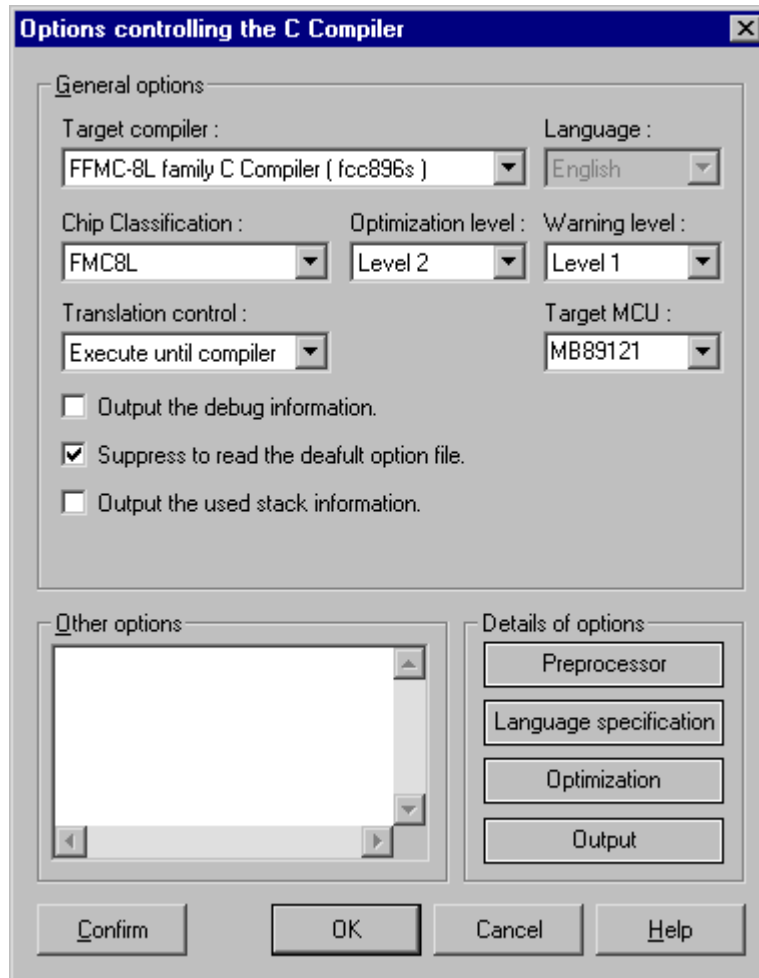
Figure 10.4-a Dialog box for setting up fcc896s compiler options

The compiler option dialog box provides the following:

❍ Target compiler

The target compiler is set in following compiler.

- fcc911s C compiler for the FR family.

- fcc907s C compiler for the F^2MC family.

- fcc896s C compiler for the F^2MC family.

- ANSI C compiler

125

❍ Chip classification

Select the chip type according to the MCU to be used.

The chip type that can be selected depends on the compiler type.

❍ Optimization level

Optimization allows setting of one of seven optimization levels.

The default is Level-2.

- None: Performs no optimization.
- Level-1: Performs level-1 optimization.
- Level-2: Performs level-2 optimization.
- Level-3: Performs level-3 optimization.
- Level-4: Performs level-4 optimization.
- Speed-preferred: Performs speed-preferred optimization.
- Size-preferred: Performs size-preferred optimization.

❍ Warning level

Warning allows setting of one of nine levels.

The default is Level-1.

- Level-0: Outputs no warning messages.
- Level-1: Outputs level-1 warning messages.
- Level-2: Outputs level-2 warning messages.
- Level-3: Outputs level-3 warning messages.
- Level-4: Outputs level-4 warning messages.
- Level-5: Outputs level-5 warning messages.
- Level-6: Outputs level-6 warning messages.
- Level-7: Outputs level-7 warning messages.
- Level-8: Outputs level-8 warning messages.

❍ Translation control

Translation control has two settings.

The default is configured as Execute up to Compilation.

- Execute up to preprocessing: Performs translation up to preprocessing.
- Execute up to Compilation: Performs translation up to compilation.

❍ Target MCU

Please set the MB number which the user uses in the Target MCU.

In default, the MB89121 type is set.

❍ Output the debug information

Specify whether to output the debug information.

The default is off.

❍ Suppress to read the default option file

Specify whether to suppress the reading of the default option file.

The default is on.

❍ Output the used stack information

Specify whether to output the used stack information file.

The default is off.

❍ Details of options

- Preprocessor: Set up the detailed options for the preprocessor.

- Language Specification: Set up the detailed options for the language specifications.

- Optimization: Set up the detailed options for optimization.

- Output: Set up the detailed options for output information.

❍ Other options

Allows setting of compiler-related options other than the above.

❍ Buttons

- [Confirm] button: Checks the settings of the options.

- [OK] button: Makes the settings effective.

- [Cancel] button: Cancels the changes.

- [Help] button: Displays the help file for the dialog box.

10.4.1 Detailed options for the fcc896s preprocessor

Set up the detailed compiler options related to the preprocessor of the fcc896s compiler. There are two dialog boxes, one for setting up include paths and the other for setting macro definitions. Set up the necessary options in both dialog boxes.

■ **Dialog box for setting up the detailed options (include paths) related to the preprocessor**

Choose [Preprocessor] command from the [Compile option] dialog box, the [Preprocessor] dialog box is displayed. Figure 10.4-b shows the dialog box for setting up Include Path when select [Include path] tab which exists in the [Preprocessor] dialog box.
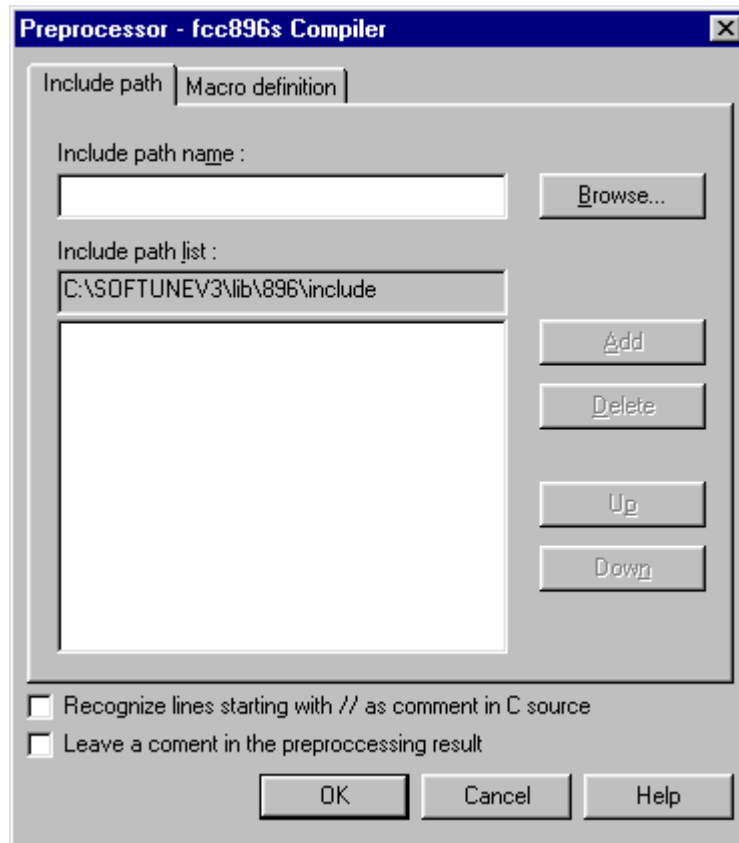
Figure 10.4-b Dialog box for setting up include paths

The dialog box for setting up include paths provides the following:

❍ Include path name field

Enter the include path used during compilation.

❍ Include path list

Displays the registered include paths.

The registered include paths are passed to the preprocessor, starting from the uppermost one.

128

❍ [Browse] button

Displays the dialog box for selecting an existing include path.

The path selected from the dialog box is displayed in the include path entry field.

❍ [Add] button

Adds the include path entered in the include path entry field to the include path register space.

❍ [Delete] button

Deletes an include path from the include path register space.

❍ [Up] button

Moves an include path registered in the include path register space, up one line.

❍ [Down] button

Moves an include path registered in the include path register space, down one line.

❍ "Recognize lines starting with // as comment in C source" check box

Select this option to recognize lines starting with // as comment in preprocessor.

When this option is set, this setting is applied to both the check and the compilation.

The default is off.

❍ "Leave a comment in the preprocessing result" check box.

Allows the user to leave a comment in the preprocessing results.

The default is off.

■ **Dialog box for setting up the detailed options (macro definitions) related to the preprocessor**

Figure 10.4-c shows the dialog box for setting up Macro definition when select [Macro definition] tab which exists in the [Preprocessor] dialog box.
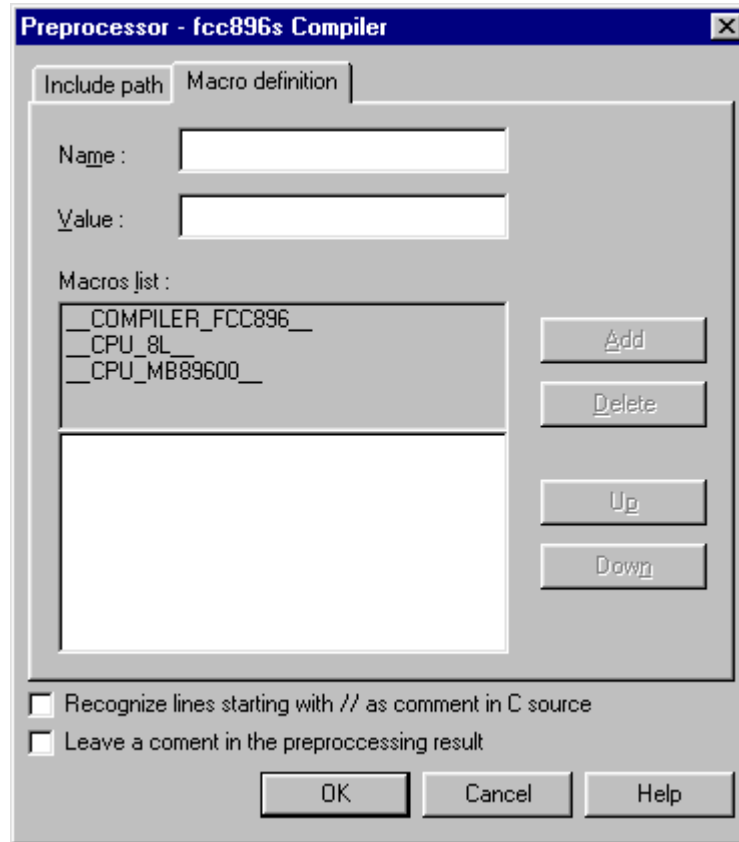
Figure 10.4-c Dialog box for setting up macro definitions

The dialog box for setting up macro definitions provides the following:

❍ Macro name field

Enter the macro to be registered during compilation.

❍ Macro value field

Enter the value of macro.

❍ Macros list

Displays the registered macros.

The registered macros are passed to the preprocessor, starting from the uppermost one.

❍ [Add] button

Adds the macro entered in the macro entry field to the macro register space.

❍ [Delete] button

Deletes a macro from the macro register space.

❍ [Up] button

Moves a macro registered in the macro register space, up one line.

❍ [Down] button

Moves a macro registered in the macro register space, down one line.

❍ "Recognize lines starting with // as comment in C source" check box

Select this option to recognize lines starting with // as comment in preprocessor.

When this option is set, this setting is applied to both the check and the compilation.

The default is off.

❍ "Leave a comment in the preprocessing result" check box.

Allows the user to leave a comment in the preprocessing results.

The default is off.

10.4.2 Detailed options related to the fcc896s language specifications

The following explains how to set up the detailed options related to the language specifications of the fcc896s compiler.

■ **Dialog box for setting up the detailed options related to the language specifications**

Choose [Language specifications] command from the [Compile option] dialog box, the [options related to the language specifications] dialog box is displayed. (See Figure 10.4-d)
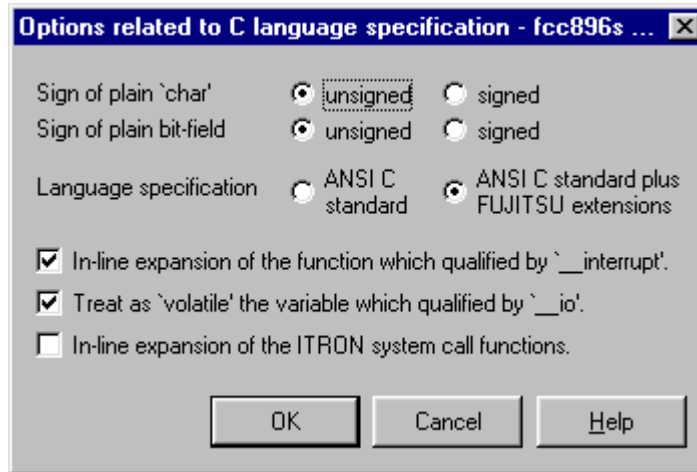
Figure 10.4-d Dialog box for setting up the options related to the language specifications

The dialog box for setting up the options related to the language specifications provides the following:

❍ Sign of plain 'char'

Specify whether the data type of char is signed or unsigned.

The default is unsigned.

❍ Sign of plane bit field

Specify whether the data type of int is signed or unsigned with the bit fields.

The default is unsigned.

❍ Language specification

Set the language specification level.

The default is ANSI C standard plus FUJITSU extensions

❍ In-line expansion of the function which qualified by '_interrupt'

Specify whether to perform in-line expansion of the function which qualified by '_interrupt'.

The default is on.

❍ Treat as 'volatile' the variable which qualified by '_io'

Specify whether to treat as 'volatile' the variable which qualified by '_io'.

The default is on.

❍ In-line expansion of the ITRON system call functions

Specify whether to perform in-line expansion of the ITRON system call functions.

The default is off.

10.4.3 Detailed optimization options of the fcc896s compiler

Set up the detailed (optimization) options of the fcc896s compiler.

■ **Dialog box for setting up the optimization options**

Choose [Optimization] command from the [Compile option] dialog box, the [Options controlling optimization] dialog box is displayed. (See Figure 10.4-e).
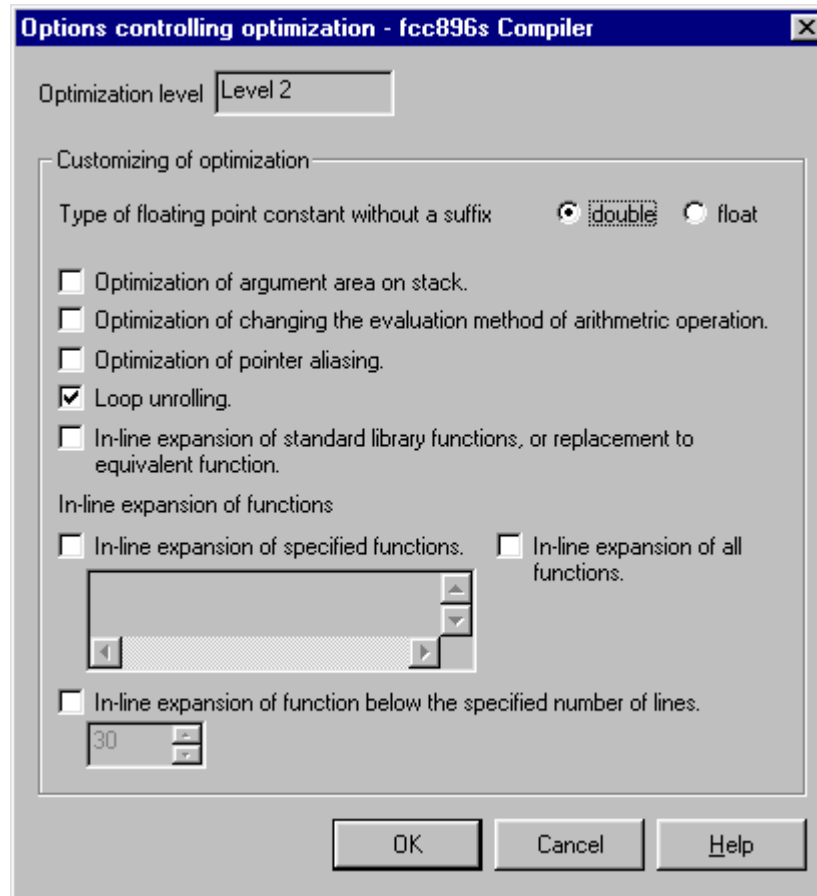
Figure 10.4-e Dialog box for setting up optimization options

The dialog box for setting up the optimization options provides the following:

❍ Type of floating point constant without a suffix

Set the type of floating point constant without a suffix.

The default is double.

❍ Optimization of argument area on stack

Specify whether to perform optimization of argument area on stack.

The default is off.

❍ Optimization of changing the evaluation method of arithmetic operation

Specify whether to perform optimization of changing the evaluation method of arithmetic operation.

The default is off.

❍ Optimization of pointer aliasing

Specify whether to perform optimization of pointer aliasing.

The default is off.

❍ Loop unrolling

Specify whether to perform loop unrolling.

The default is on.

❍ In-line expansion of standard library functions or replacement to equivalent function

Specify whether to perform in-line expansion of standard library functions or replacement to equivalent function

The default is off.

❍ In-line expansion of specified functions

Place a check mark in the check box and specify the functions subject to in-line expansion.

The default is off.

❍ In-line expansion of function below the specified number of lines

Place a check mark in the check box and specify the number of the line below which in-line expansion is to be performed.

The default is off.

135

10.4.4 Detailed options related to fcc896s output

The following explains how to set up the options related to the output of the fcc896s compiler.

■ **Dialog box for setting up the detailed options related to output**

Choose [Output] command from the [Compile option] dialog box, the [options related to output object] dialog box is displayed. (See Figure 10.4-f)
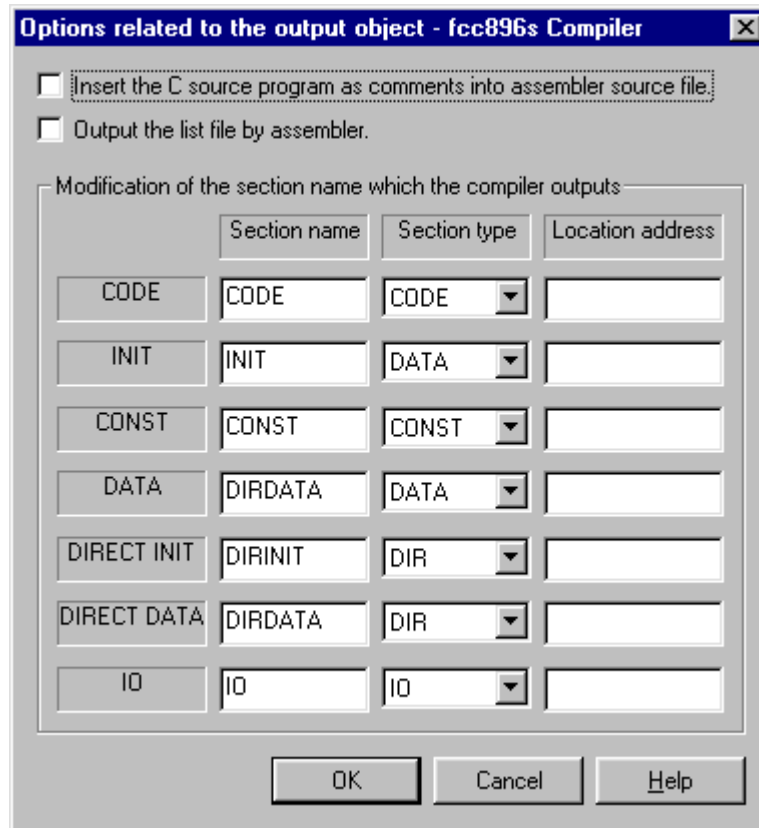
Figure 10.4-f Dialog box for setting up the options related to output

The dialog box for setting up the options related to output provides the following:

❍ Insert the C source program as comments into assembler source file

Specify whether to insert the C source program as comments into an assembler source file (*.asm).

The default is off.

❍ Output the list file by assembler

Specify whether to output an assembly list.

The default is off.

136

❍ Modification of the section name which the compiler outputs

The sections to be output by the compiler can be changed.

Section names, section types, and location addresses can be entered.

10.5 Setting up ANSI Compiler Options

This section explains how to set up the fcc911s compiler options.

■ **Dialog box for setting up the ANSI compiler options**

Choose [Compile Option] command from the [Compile] menu or [Compile Option] command from the pop-up menu in the project management window, the [Compile Option] dialog box is displayed. (See Figure 10.5-a)
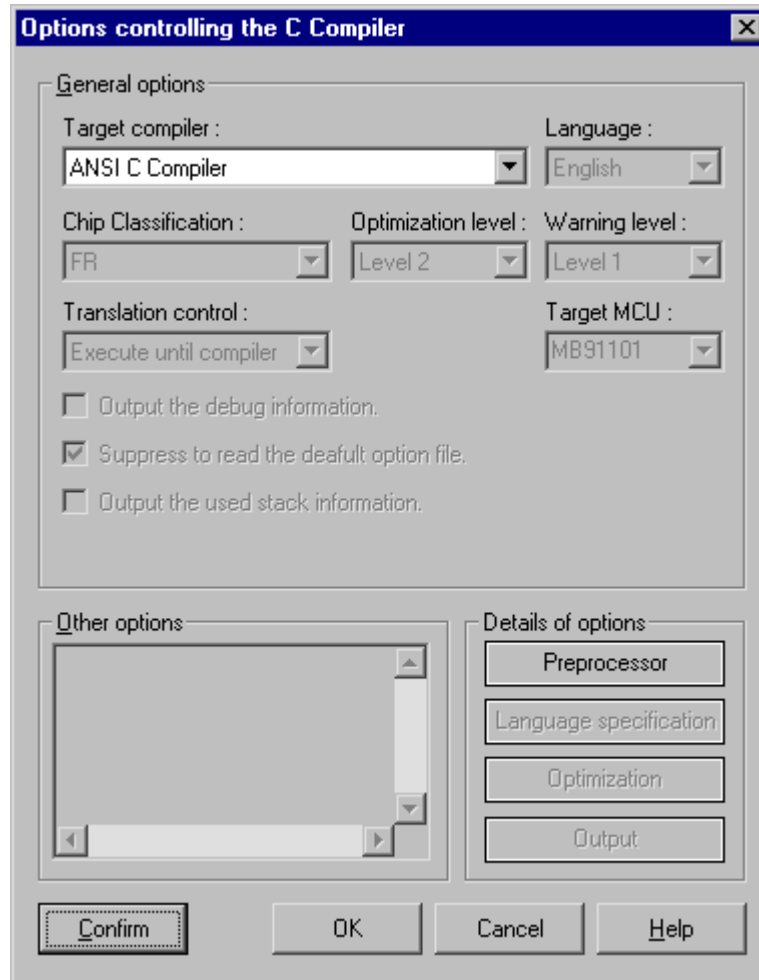
Figure 10.5-a Dialog box for setting up ANSI compiler options

The compiler option dialog box provides the following:

❍ Target compiler

The target compiler is set in following compiler.

- fcc911s C compiler for the FR family.

- fcc907s C compiler for the F^2MC family.

- fcc896s C compiler for the F^2MC family.

- ANSI C compiler

138

❍ Buttons

- • [Confirm] button: Checks the settings of the options.

- • [OK] button: Makes the settings effective.

- • [Cancel] button: Cancels the changes.

- • [Help] button: Displays the help fine for the dialog box.

10.5.1 Detailed options for the ANSI preprocessor

**Set up the detailed compiler options related to the preprocessor of the fcc911s
compiler. There are two dialog boxes, one for setting up include paths and the other for
setting macro definitions. Set up the necessary options in both dialog boxes.**

■ **Dialog box for setting up the detailed options (include paths) related to the preprocessor**

Choose [Preprocessor] command from the [Compile option] dialog box, the [Preprocessor]
dialog box is displayed.

Figure 10.5-b shows the dialog box for setting up Include Path when select [Include path] tab
which exists in the [Preprocessor] dialog box.

Figure 10.5-b Dialog box for setting up include paths

The dialog box for setting up include paths provides the following:

❍ Include path name field

Enter the include path used during compilation.

❍ Include path list

Displays the registered include paths.

The registered include paths are passed to the preprocessor, starting from the uppermost
one.

140

❍ [Browse] button

Displays the dialog box for selecting an existing include path.

The path selected from the dialog box is displayed in the include path entry field.

❍ [Add] button

Adds the include path entered in the include path entry field to the include path register space.

❍ [Delete] button

Deletes an include path from the include path register space.

❍ [Up] button

Moves an include path registered in the include path register space, up one line.

❍ [Down] button

Moves an include path registered in the include path register space, down one line.

❍ "Recognize lines starting with // as comment in C source" check box

Select this option to recognize lines starting with // as comment in preprocessor.

When this option is set, this setting is applied to both the check and the compilation.

The default is off.

❍ "Leave a comment in the preprocessing result" check box

Allows the user to leave a comment in the preprocessing results.

The default is off.

■ **Dialog box for setting up the detailed options (macro definitions) related to the preprocessor**

Figure 10.5-c shows the dialog box for setting up Macro definition when select [Macro definition] tab which exists in the [Preprocessor] dialog box.
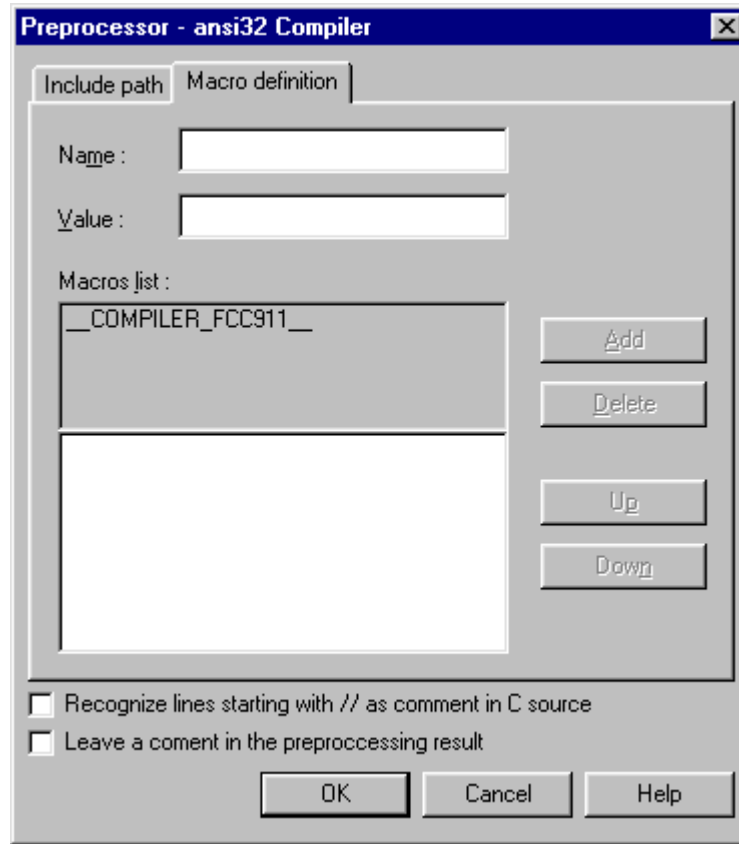
Figure 10.5-c Dialog box for setting up macro definitions

The dialog box for setting up macro definitions provides the following:

❍ Macro name field

Enter the macro to be registered during compilation.

❍ Macro value field

Enter the value of macro.

❍ Macros list

Displays the registered macros.

The registered macros are passed to the preprocessor, starting from the uppermost one.

❍ [Add] button

Adds the macro entered in the macro entry field to the macro register space.

❍ [Delete] button

Deletes a macro from the macro register space.

142

❍ [Up] button

Moves a macro registered in the macro register space, up one line.

❍ [Down] button

Moves a macro registered in the macro register space, down one line.

❍ "Recognize lines starting with // as comment in C source" check box

Select this option to recognize lines starting with // as comment in preprocessor.

When this option is set, this setting is applied to both the check and the compilation.

The default is off.

❍ "Leave a comment in the preprocessing result" check box

Allows the user to leave a comment in the preprocessing results.

The default is off.

144

CHAPTER 11   Setup

This chapter explains the setup operation.

11.1 Setup Menu

The compiler environment can be set up using setup commands.
The following setup commands can be used from the [Setup] menu:

* **[Editor Customize] command**
* **[Compilation Environment] command**

■ **Setup menu**
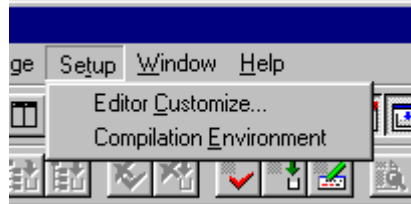
Figure 11.1-a Setup menu

❍ [Editor Customize] command

Displays the setup dialog box for the editor to be used in C checker.

❍ [Compilation Environment] command

Specifies the compilation environment that will use the C Checker compilation function.

To use this command, the Softune compiler must be installed in advance.

11.2 Editor Customization

The C checker has the standard editor function. An external editor can also be called using the customization function. This section explains how to set up the external editor.

■ **Setting External Editor**

With the graph flow window not displayed, select the [Editor Customize] command displayed in the [Setup] menu to display the [Editor Customization] dialog box.
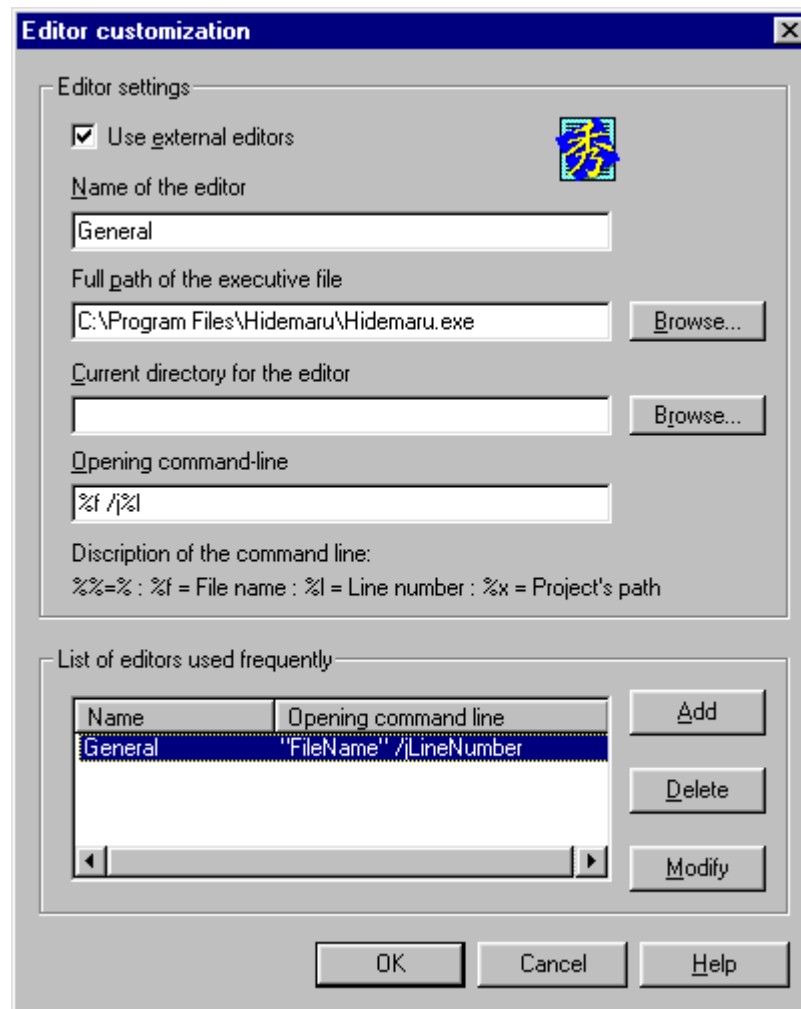
Figure 11.2-a [Editor Customization] dialog box

Check the [Use External Editor] check box to register and use a familiar external editor.

❍ Editor name

Register the name of the editor used.

❍ Full path of the executive file

Specify the full-path of the execution file of the editor.

147

❍ Current directory for the editor

Specify the directory which starts the editor.

❍ Opening command-line

Specify the argument passed to the editor.

❍ List

The registration situation of the editor is displayed.

11.3 Environment of Compiler

A compiler can be activated from the C Checker. After the environment of the compiler is set up, the compiler can be activated. This section explains how to set up the environment of the compiler.

■ **Environment of compiler**

Select the [Compilation Environment] command displayed in the [Setup] menu. The [Environment of the compiler] dialog box is displayed.
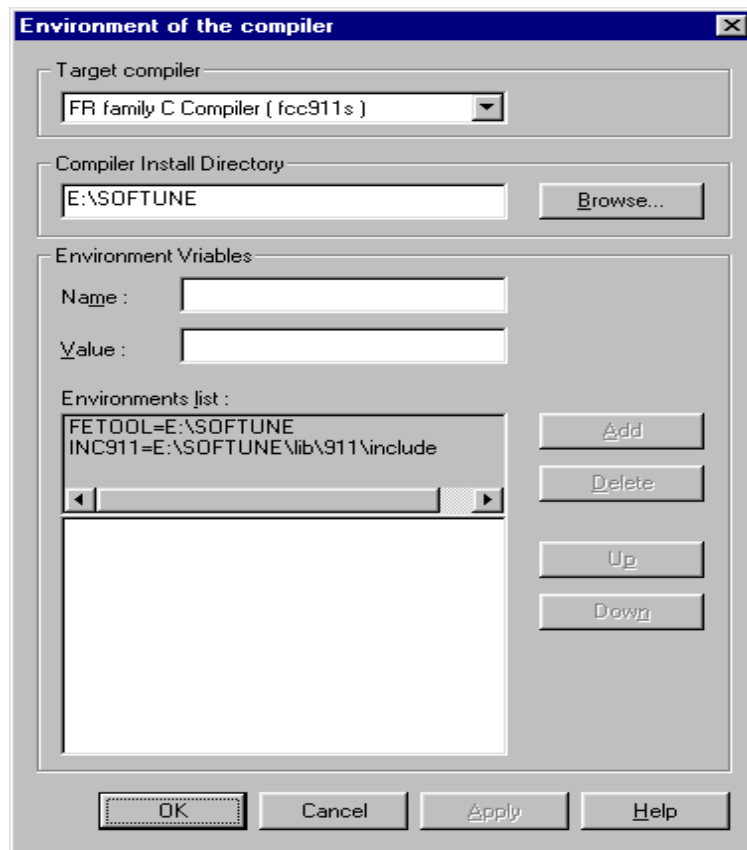
Figure 11.3-a [Environment of the compiler] dialog box

The [Environment of the compiler] dialog box has the following features:

❍ Target Compiler

The target compiler is set in following compiler.

The Environment of the compiler can be set in each target compiler.

- fcc911s C Compiler for the FR family
- fcc907s C Compiler for the F^2MC-16 family.
- fcc896s C Compiler for the F^2MC-8L family.

❍ Compiler Install Directory

Specify the name of the directory in which the compiler is installed.

149

❍ Environmental variables adopted in compilation

- Name: Enter the name of the environmental variable to be registered.

- Value: Enter the value of the environmental variable to be registered.

- Environment List: Displays the environment variables already registered.

- [Add] button: Adds the environment variable (entered in the environmental variable input field) to the list of registered environmental variables.

- [Delete] button: Deletes an environmental variable already registered in the list of registered environment variables. To delete the environmental variable, select the environmental variable to be deleted, click the left button of the mouse, then press the [Delete] button.

- [Up] button: Moves the target environmental variable upwards in the list of environmental variables.

- [Down] button: Moves the target environmental variable downwards in the list of environmental variables.

Remarks

To use the compilation function, a compiler manufactured by Fujitsu must be installed in advance. If a Fujitsu compiler is not installed, a message indicating a no-compiler-installed state is displayed when the compiler environment function is used.

CHAPTER 12 Window

This chapter explains how to display and manipulate windows.

12.1 Window Menu

12.1 Window Menu

The Window menu allows the user to manipulate windows and change the way windows are displayed.
The Window menu provides the following commands:

- **[Overlap] command**
- **[Arrange vertically] command**
- **[Arrange horizontally] command**
- **[Arrange icons] command**
- **[Window names 1, 2, …] command**

■ **Window menu**

❍ [Overlap] command

Overlaps multiple windows.

❍ [Arrange vertically] command

Arranges multiple windows vertically.

❍ [Arrange horizontally] command

Arranges multiple windows horizontally.

❍ [Arrange icons] command

Arranges the icons when individual windows are minimized at the bottom of the main window of the application.

If a file window is open, overlapping the bottom of the main window, the icons hidden by that window are not visible.

❍ [Window names 1, 2, …] command

Displays a list of file windows currently opened by the C Checker at the bottom of the [Window] menu.

A check mark appears next to the name of the currently active file window. To make a window active, select the window name.

CHAPTER 13   Help

This chapter explains how to display and manipulate help files.

13.1 Help Menu

13.1 Help Menu

The Help menu allows the user to search for hints and topics.
The Help menu provides the following commands:
- **[Tip of the Day] command**
- **[Help Topics] command**
- **[About SOFTUNE C Checker…] command**

■ **Help menu**

❑ [Tip of the Day] command

Displays the C Checker hint dialog box.

This dialog box can also be displayed when the C Checker is started.

❑ [Help topics] command

Displays the contents of the help files containing the function outlines and operating procedures for the C Checker.

It also prints help files and searches for keywords.

❑ [About SOFTUNE C Checker…] command

Displays the C Checker version, copyright, and other information.

CHAPTER 14　Sorting of the diagnostic message

This chapter explains how to sort of the diagnostic message

14.1 Pop-up menu of the diagnostic message

The pop-up menu is displayed in the diagnostic message window by clicking the right mouse button on the diagnostic message in the diagnostic message window.

■ **Pop-up menu of the diagnostic message**
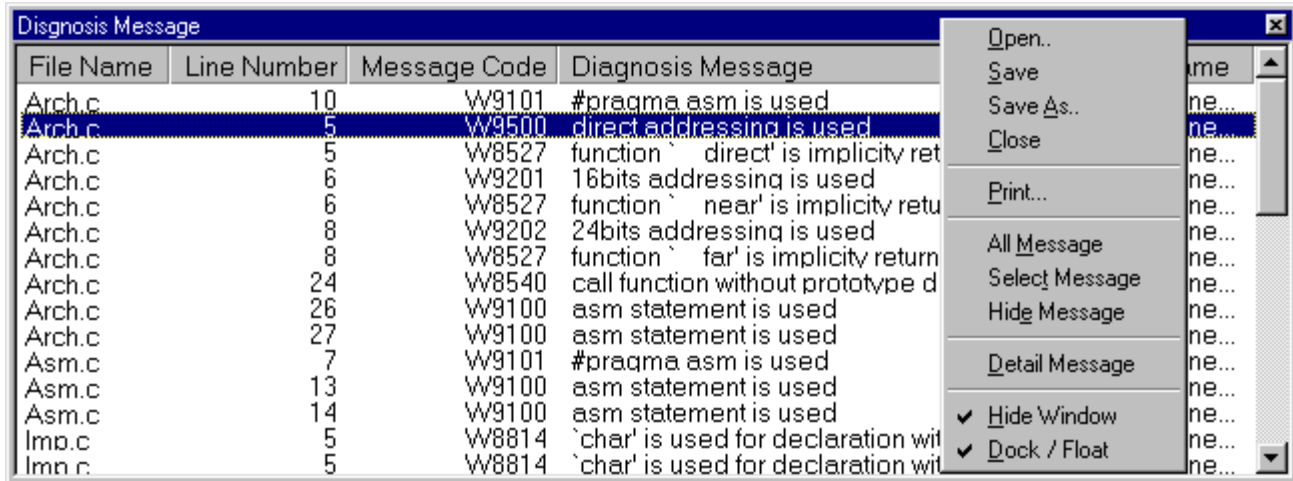
Figure 14.1-a popup menu of the diagnostic message

Displays the popup menu to allow the following operations to be performed quickly:

- Open: Open the diagnostic message file.

- Save: Save the diagnostic message into file.

- Save As: Save the diagnostic message with another name.

- Close: Close the diagnostic message.

- Print: Print the diagnostic message.

- All Message: All diagnostic messages are displayed.

- Select Message: Only the message of the same message code as the code of the selected diagnostic message is displayed.

- Hide Message: All of the message which has been [Hide Message] and all of the messages of the same message code as the code of the selected diagnostic message are made non-display.

- Detail message: Displayed a detailed message.

- Dock / Float: Connects a subwindow to the bottom of its parent window or disconnects it.

- Hide: Turns the window off.

[Procedure]

Click the right mouse button within the window.

156

14.2 Sorting of the diagnostic message

The diagnostic message can be sorted by selecting [All Message], [Select Message], [Hide Message] command from pop-up menu of the diagnostic message.

■ **Select Message**

Choose the [Select Message] command from popup menu on the diagnostic message window, Only the message of the same message code as the code of the selected diagnostic message is displayed.

When the message code W8004 is made [Select Message] in figure 14.2-a, only W8004 is displayed and screen becomes as shown in figure14.2-b.
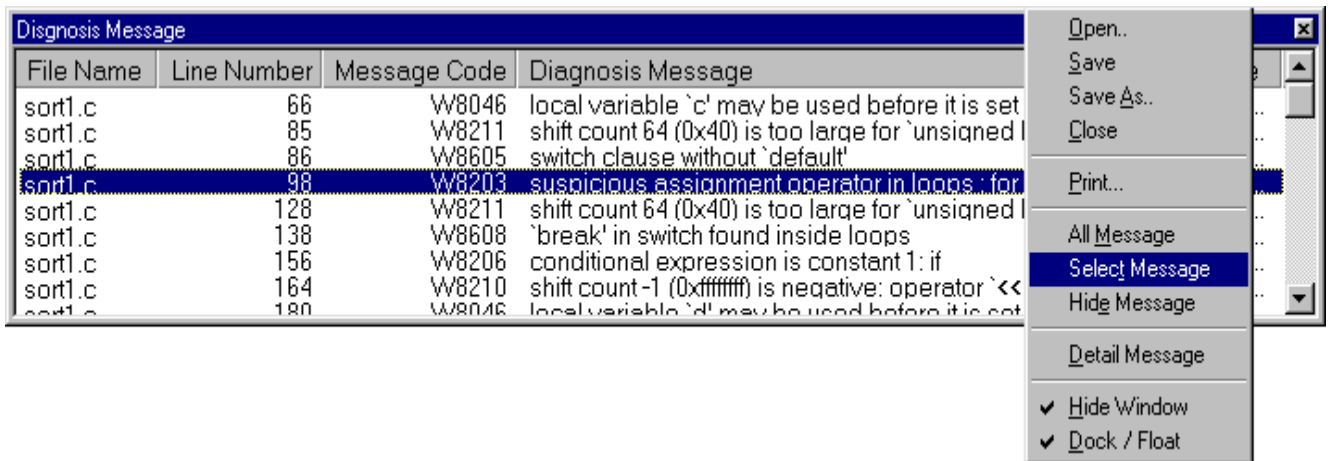
Figure 14.2-a Selection screen of the [Select Message] command
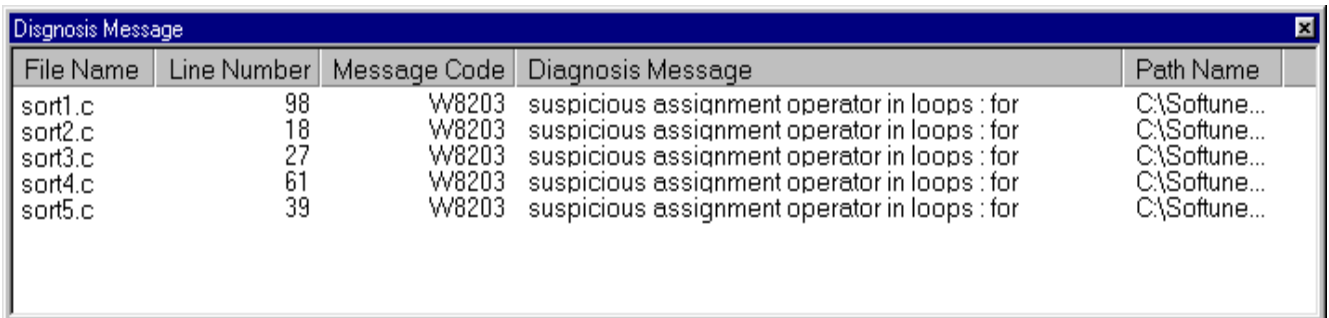
Figure 14.2-b [Select Message]

157

■ **Hide Message**

Choose the [Hide Message] command from popup menu on the diagnostic message window, all of the message which has been [Hide Message] and all of the messages of the same message code as the code of the selected diagnostic message are made non-display.

When the message code W8004 is made [Hide Message] in figure 14.2-c, all W8004 is not displayed and screen becomes as shown in figure 14.2-d.
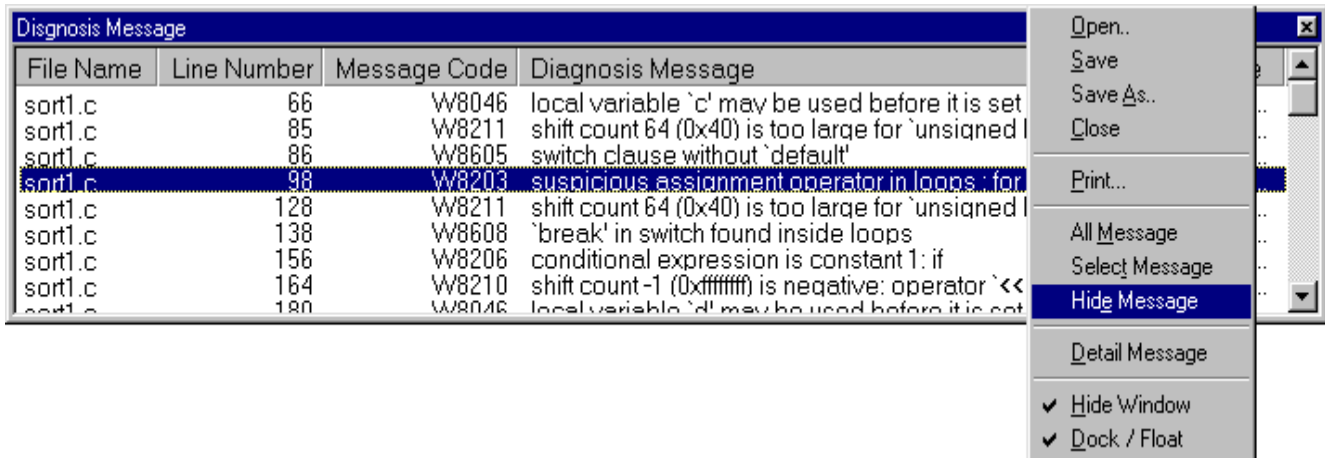
Figure 14.2-c Selection screen of the [Hide Message] command
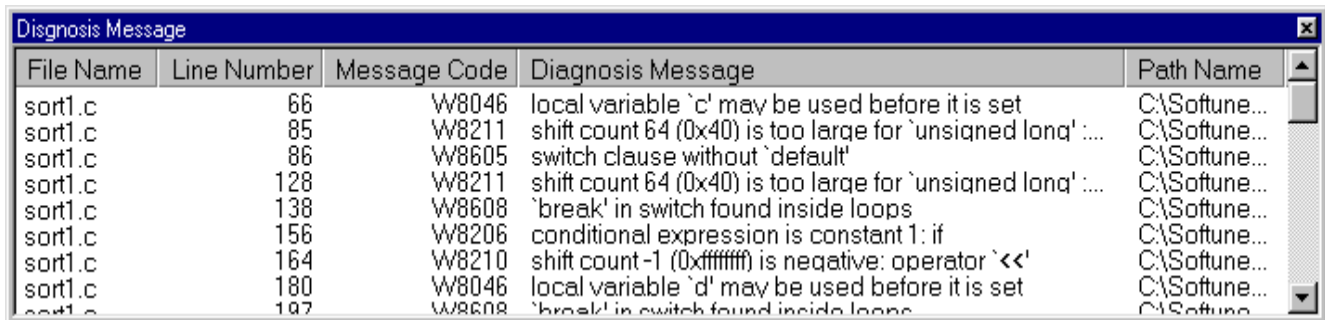
Figure 14.2-d [Hide Message]

158

■ **All Message**

Choose the [All Message] command from popup menu on the diagnostic message window, all diagnostic messages are displayed.

Choose the [All Message] command in figure 14.2-e, all messages are displayed and screen becomes as shown in figure14.2-f.
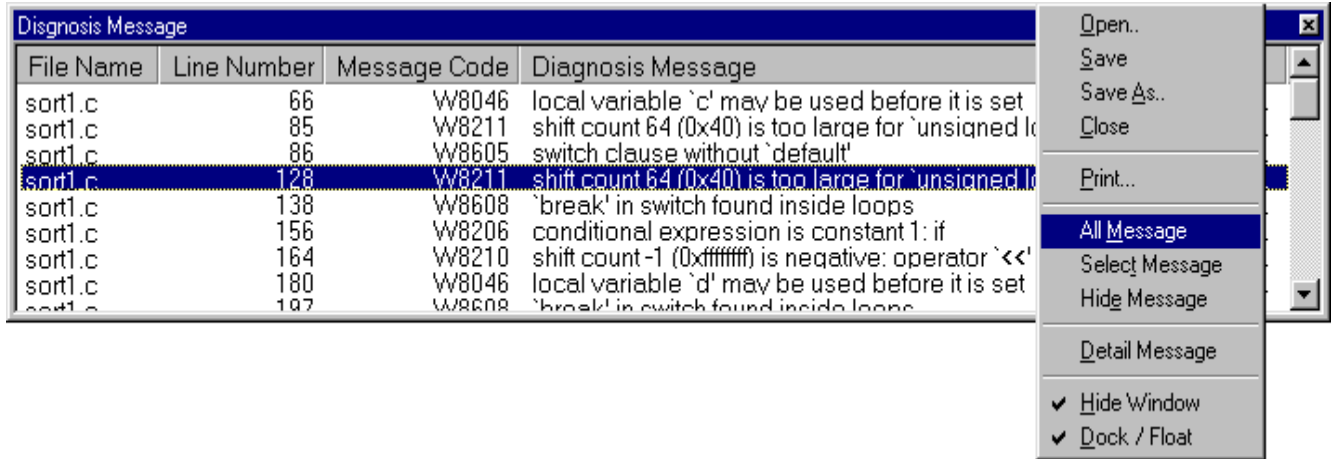
Figure 14.2-e Selection screen of the [All Message] command
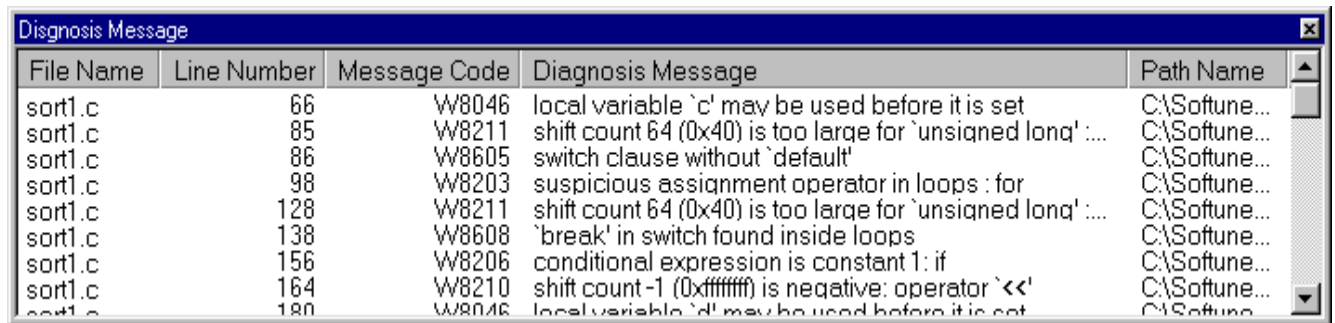
Figure 14.2-f [All Message]

159

CHAPTER 14 Sorting of the diagnostic message

APPENDIX

This appendix explains problems that may occur during the use of Softune C Checker.

APPENDIX A "Troubleshooting"

APPENDIX A   Troubleshooting

Solves technical problems that may be encountered when using Softune C Checker.
- **Activation problems**
- **Development environment-related**
- **Standard editor-related**
- **Frequently asked questions**

■ **Activation Problems**

❍ **Unable to start Softune C Checker**

Unable to start and the message "Loading...... Resource DLL -FAIL- "

[Answer]

There is a possibility that C Checker is not correctly installed.

Please install again after executing the uninstallation.

❍ **Period of use**

The message "[0] day of trial remaining" appears.

[Answer]

The version of Softune C Checker being used is an evaluation version.

This version can be used for 45 days after installment. To continue use, the full version must be purchased.

❍ **Unable to compile C source**

After the message "Starting compile .... ", the error message "Failed compile your programs. " is displayed.

[Answer]

There is a possibility that C Compiler is not correctly installed.

Please install the C Compiler in the same place as the C Checker.

■ **Development Environment-related**

❍ **"Invalid directory for Compiler"**

The message of "Invalid directory for Compiler" appears when you execute the compilation.

[Answer]

There is a possibility that the compiler has not been specified yet.

Please select "Compilation Environment" command of the "setup" menu, and Compilation Environment dialog box is displayed.

Then please specify the folder where the compiler exists.

Close this dialog box using the "Close" button if you do not install the compiler.

❍ **Can I use SAMBA ?**

Is it possible to analyze a UNIX source program on Windows using a SAMBA Server?

[Answer]

It is possible, but the following points must be noted.

- Standard file name

  - As the upper case/lower case distinction for directory names and file names on the UNIX Server no longer exists, it is necessary to modify as appropriate.

APPENDIX A Troubleshooting

■ Standard Editor-related

❍ **"Failed to create the internal editor"**

The message of "Failed to create the internal editor" appears when the edit window is started by a internal editor.

[Answer]

The standard editor may not be registered in the system. Within explorer, drag the library file SiiEd2.ocx included in the install directory and drop it over regsvr.exe.

If the message "DllRegisterServer in SiiEd2.ocx succeeded" is displayed, the window can be activated normally.

❍ **"member not found"**

When the creation of the new file and the customizing command of the editor are selected, the warning dialog "member not found" is displayed.

[Answer]

There is a possibility to have used the internal editor of an old version.

Library file included to an installation directory by Explorer.

And check that the version of SiiEd2.ocx is more than 2.3.2.6, drag and drop on regsvr32.exe.

If the message "DllRegisterServer in SiiEd2.ocx succeeded" is displayed, the window can be activated normally.

■ Frequently Asked Questions

❍ **Can I treat the line which starts by // as a comment ?**

[Countermeasures]

By the following setting, the line which starts by // can be treated as a comment.

When this option is set, the line which starts by // in both the check and the compilation is treated as a comment.

1. Specify the "Compile option" command of the "Compile" menu.

2. Specify the "Preprocessor" command of the "Details of options" menu in the "Options controlling the Compiler" dialog box.

3. Click the "Recognize lines starting with // as comment in C source" check box.

❍ **Can I use the project file of the Softune Workbench ?**

[Countermeasures]

In the C Checker, the project file for the C Checker can be newly made by misappropriating the project file of the Softune Workbench.

Specify the "New" command of the "project" menu, and "Create project" dialog box is opened.

Check the "Create to project from SOFTUNE project file" check-box, and select the project file of the WorkBench.

164

❍ **Can I see the table of the diagnostic message ?**

[Countermeasures]

The diagnostic message list can be seen from the help file.

Select "Help topics" command of the "Help" menu and the help file is opened.

Click the "Message" -> "Diagnostic message list", the list of the diagnostic message of each item can be seen.

❍ **Can I use the Softune compiler installed in another folder ?**

[Countermeasures]

In C Checker, the compiler used by the compilation function can be specified.

Select the "Compile Environment" command of the "Setup" menu, and open the "Compile Environment" dialog box. The compiler corresponding to the selected [target compiler] can be specified.

❍ **Can I use the external editor ?**

[Countermeasures]

Not only the editor built into C Checker but also an external editor can be used.

Select the "Editor Customize" command displayed in the "Setup" menu to display the EDITOR CUSTOMIZE dialog box.

Please refer to the Help file etc. of each editor for the setting in the editor and the detailed starting option.

❍ **Can I sort the diagnostic message ?**

[Countermeasures]

This softune C Checker sorts the diagnostic message in the diagnostic message window.

Select the diagnostic message and the pop-up menu is displayed by clicking the right mouse button.

The diagnostic message can be sorted by selecting "All message", "Select message", "Hide message" command from pop-up menu of the diagnostic message window.

APPENDIX A Troubleshooting

INDEX

INDEX

FUJITSU SEMICONDUCTOR • CONTROLLER MANUAL

FR/F^{2}MC FAMILY

SOFTUNE C CHECKER MANUAL

for V3

Free Manuals Download Website

http://myh66.com

http://usermanuals.us

http://www.somanuals.com

http://www.4manuals.cc

http://www.manual-lib.com

http://www.404manual.com

http://www.luxmanual.com

http://aubethermostatmanual.com

Golf course search by state

http://golfingnear.com

Email search by domain

http://emailbydomain.com

Auto manuals search

http://auto.somanuals.com

TV manuals search

http://tv.somanuals.com