

Quickstart for Embedded Systems

Featuring Hiware Technology

Because of last-minute software changes, some information in this manual may be inaccurate. Please read the Release Notes, on the CodeWarrior CD, for the latest information.

Revised: February/22/2002

©Metrowerks, Inc., 1993, 2002; All Rights Reserved.

Documentation stored on the compact disks may be printed by licensee for personal use. Otherwise, no part of this documentation may be reproduced or transmitted in any form by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from Metrowerks, Inc.

Metrowerks, the Metrowerks logo, and CodeWarrior are registered trademarks of Metrowerks, Inc.

All other trademarks belong to their respective owners.

How to Contact Metrowerks:

U.S.A.	Metrowerks Corporation 9801 Metric Blvd., Suite #100 Austin, TX 78758 - U.S.A.
Europe	Metrowerks Europe Riehenring 175 - CH-4058 Basel (Switzerland)
ASIA/PACIFIC	Metrowerks Japan - Shibuya Mitsuba Building 5F - Udagawa-cho 20-11, Shibuya-ku - Tokyo 150-0042 Japan
World Wide Web	http://www.metrowerks.com
Registration Information	http://www.metrowerks.com/register mailto:register@metrowerks.com
Desktop Technical Support	http://www.metrowerks.com/support/desktop/ mailto:cw_support@metrowerks.com
Embedded Technical Support	http://www.metrowerks.com/support/embedded/ U.S.A: mailto:cw_emb_support@metrowerks.com Europe: mailto:support_europe@metrowerks.com Asia/Pac mailto:j-emb-sup@metrowerks.com
Sales, Marketing, & Licensing	mailto:sales@metrowerks.com
Ordering	Voice: (800) 377-5416 Fax: (512) 873-4901
Intl. Sales, Mkt & Licensing	mailto:intlsls@metrowerks.com
International Ordering	Voice: +1 512 873 4724 Fax: +1 512 873 4901

Table of Contents

Table of Contents	3
1 Introduction to CodeWarrior	5
Features	5
Project Manager	5
Source Code Editor	6
Browser.	6
Functions.	6
2 Installing Code Warrior	7
System Requirements	7
Installation	8
Setup	8
Installation Directories	10
Tips and Tricks	10
3 CodeWarrior Projects	11
Projects and Targets	11
Launching CodeWarrior	12
CodeWarrior Main Menu	12
Build Targets	13
Target Customization and Configuration	13
Project Window.	13
Files View	14
Link Order View	14
Targets View	15
Adding Files to your Project	16
Removing Files from your Project	17
Touching Files.	17
4 Creating a Project	19
Building a Project	19
Making a Project.	21
Simulating and Debugging a Project	22

5 Specifying Target Settings	25
Target Settings Panel	26
Access Paths Panel	27
Build Extras Panel	28
Runtime Settings Panel	29
Files Mappings Panel	30
Source Trees Panel	31
Hiware Build Tools	32
Assembler Panel	33
Burner Panel	34
Compiler Panel	36
Importer Panel	39
Linker Panel	40
Libmaker Panel	41
Metrowerks Build Tools	43
Language Settings Panels	44
C/C++ languages panel	44
C/C++ warnings panel	44
Assembler panel	45
Code Generation Panels	46
Global optimizations panel	46
Processor panel	46
Disassembly panel	47
Linker Panels	47
Linker panel	47
6 Technical Support	49
E-mail	49
Fax	49
Mail	50
Internet	50
Index	51

Introduction to CodeWarrior

The CodeWarrior Integrated Development Environment (IDE) for embedded microprocessors is a powerful, easy-to-use tool suite for increasing your software development productivity.

NOTE The CodeWarrior IDE for embedded targets is hosted only on Microsoft Windows 9X, Windows NT, and Windows 2000 or later.

This chapter describes the features and functions of the CodeWarrior IDE.

This chapter contains the following sections:

- [Features](#)
- [Functions](#)

Features

The software provides an intuitive Graphical User Interface (GUI) with these main features:

- Project Manager
- Source Code Editor
- Browser

Project Manager

Using the CodeWarrior IDE Project Manager, you can gather program files and options into a single project file and specify which plug-in compilers and linkers to use for creating the final program.

Source Code Editor

Use the Code Warrior IDE Source Code Editor to edit source code and text. The Source Code Editor lets you:

- Edit, search, and replace text in one file or several files
- Split an editor window into multiple panes
- Toggle between a source file and its related interface file
- Open an interface file referred to by a source file
- Set markers at arbitrary locations in a text file
- Jump to any routine in any file instantly
- Customize the source code display

Browser

Use the commands in any source code view and controls in the browser's window for quick, intuitive access to variables, routines, enumeration, and definitions of classes, and other source-code elements.

Functions

The software has these main functions:

- Highly optimizing C/C++ Compiler
- Powerful Macro Assembler
- SmartLinker, which only links objects that are really referenced
- Burner to create Motorola S-Records, Intel Hex files, or Binary files
- Decoder to decode object and absolute files
- Libmaker to generate libraries
- Multipurpose Debugger, which allows
 - Simulation and debugging of an embedded application.
 - Simulation and debugging of a real time embedded application.
 - Simulation and/or cross-debugging of an embedded application.
- Multi-Language Debugging: Assembly, C, and C++
- True Time Stimulation
- Simulation of a hardware design (such as a board, processor, or I/O chip)

Installing Code Warrior

This chapter provides CodeWarrior installation, setup, and maintenance instructions, as well as an installation directory listing.

This chapter includes the following sections:

- [System Requirements](#)
- [Installation](#)
- [Setup](#)
- [Installation Directories](#)
- [Tips and Tricks](#)

System Requirements

The Windows-hosted version of the CodeWarrior IDE requires:

- Pentium processor or greater (recommended: Intel Pentium-class processor or AMD K6-class microprocessor)
- At least 64 megabytes of RAM
- Approximately 90 megabytes of free hard disk space for minimal installation
- Approximately 450 megabytes of free hard disk space for full installation
- Microsoft Windows 9X, Windows NT 4.0 Service Pack 3, Windows 2000, or later
- CD-ROM drive from which to install the software

Installation

Install the software as follows:

1. Load the CD-ROM into the CD-ROM drive

If **CD AutoPlay** in Windows is active (default setting), an install program appears. If **CD AutoPlay** is not active, run `install.exe` on the CD-ROM.

The CD-Browser menu appears

2. Click **Installations**.

The installation process starts. In the dialog boxes that appear, you can customize the installation to suit your own needs.

Setup

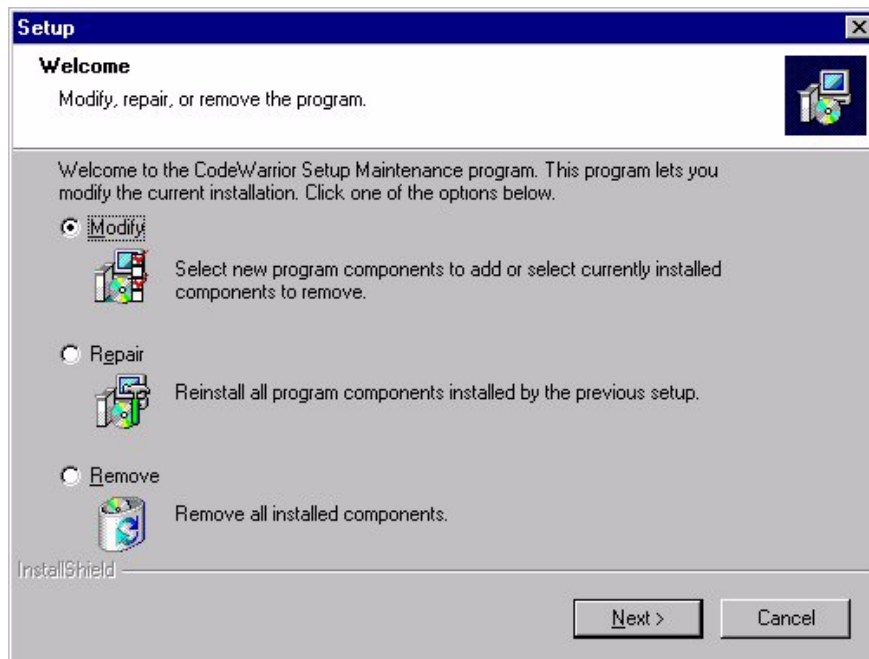
Use the Setup dialog to modify, repair, remove your installation after initial setup. Proceed as follows:

1. Run the **Add/Remove Programs** control panel.
2. Select the CodeWarrior entry in the program list.
3. Click **Add/Remove**.

The Setup dialog appears (Figure 2.1).

4. Follow the directions in the Setup dialog:
 - a. Click **Modify** to add or remove selected components.
 - b. Click **Repair** to reinstall CodeWarrior and fix problems.
 - c. Click **Remove** to uninstall CodeWarrior and its components.

Figure 2.1 Setup Dialog



NOTE The entry in the add/remove dialog box (Control Panel) always refers to the last installation you made/modified. You can use the shortcut generated in the Program group to select another installation.

Installation Directories

Table 2.1 lists the directories produced during CodeWarrior IDE installation:

Table 2.1 Code Warrior IDE Installation Directories

Directory Name	Contents
Bin	IDE executables
Bin\Plugins	All plugin DLLs
CodeWarrior Help	CodeWarrior help files
CodeWarrior Manuals	CodeWarrior online manuals
Examples	Directory for creating your examples
Lib	Processor libraries
\lib\ <target>c\prm< td=""><td>target prm file templates</td></target>c\prm<>	target prm file templates
Prog	Build tools and Debugger/Simulator
Other Metrowerks Tools	Additional programs such as registration tools
Release Notes	General information
Stationery	Project stationery
Template	Generic default files templates

Tips and Tricks

Here are some tips and tricks for using the CodeWarrior IDE:

- If you cannot launch the simulator debugger, check the settings in the Build Extras Preference Panel. Check also if the Debugger is enabled in the **Project** menu entry.
- If you delete the data folder of the project, then you delete also some project related settings. One of these settings is whether the debugger/simulator is enabled. Check the menu **Project > Enable/Disable Debugger**.
- If you cannot add a file to the project, this may be because the file extension is not present in the File Mappings Preference Panel.
- If you think that project data is corrupted, you can export and re-import your project using **File > Export Project** and **File > Import Project**.
- If you cannot run a Codewarrior installation and you have an old CodeWarrior installed, try running the *regservers.bat* batch file in the bin directory first.

CodeWarrior Projects

This chapter provides information about CodeWarrior projects and targets, as well instructions for adding, removing, and touching files.

This chapter contains the following sections:

- [Projects and Targets](#)
- [Launching CodeWarrior](#)
- [CodeWarrior Main Menu](#)
- [Build Targets](#)
- [Adding Files to your Project](#)
- [Removing Files from your Project](#)
- [Touching Files](#)

Projects and Targets

The CodeWarrior Integrated Development Environment (IDE) uses projects and build targets to organize the files and settings for creating a program. A project is a file that contains one or more build targets.

A build target can contain these elements:

- Source code files
- Libraries
- Settings
- Other projects

These target elements describe how to create software for a particular processor or operating system. Build targets in a project can share the same files, but each build target has its own settings.

The CodeWarrior IDE also has preconfigured stationery projects. Creating a new project is as easy as deciding on a platform target and then choosing the corresponding stationery project.

The CodeWarrior IDE lets you:

- Set options to choose the platform target for which you are developing code
- Customize compiler optimizations and other object code details
- Configure source-code translation
- Specify the kinds of files added to a build target
- Set additional options depending on the platform target

Launching CodeWarrior

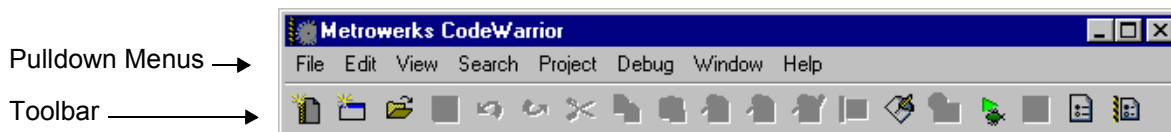
To launch CodeWarrior, make this selection from the Windows **Start** menu at the lower left of your screen:

Start > Programs > Metrowerks > CodeWarrior IDE

CodeWarrior Main Menu

Figure 3.1 shows the Code Warrior main menu and toolbar.

Figure 3.1 CodeWarrior Main Menu and Toolbar



Below the main menu is the toolbar, which provides shortcuts to many menu commands.

NOTE If you place the cursor over an icon or GUI screen object, a tooltip appears that briefly describes the object under the cursor.

Build Targets

A project contains one or more build targets. Each build target in a project is a collection of files that the IDE uses to build an output file. Project build targets can share some or all of their files.

Target Customization and Configuration

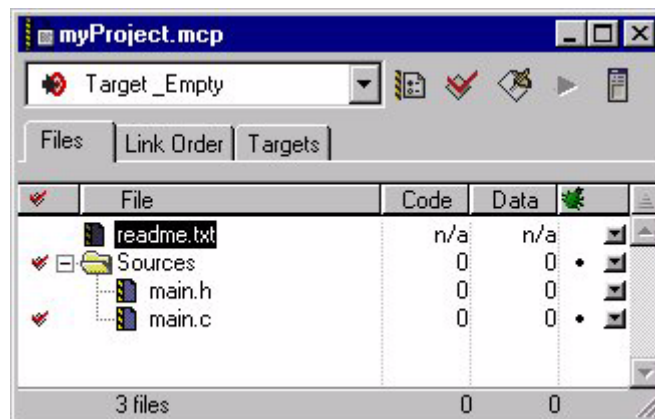
Each project's build target has its own options that customize how the IDE builds the output file. There are many options that control code optimization, browsing, debugging, compiler warnings, and more.

You can configure project build targets to depend on other build targets in the project. This makes it possible to build software that combines the output files for different platform targets into a single output file.

Project Window

The **Project** window (Figure 3.2) gives information about the files and build targets in a project file.

Figure 3.2 The Project Window



The **Project** window has three different views that you can select by clicking the tabs with corresponding labels:

- Files
- Link Order
- Targets

Files View

The **Project** window Files view shows a list of all the files in a project. Items in this view may be organized into hierarchical groups that you create and arrange.

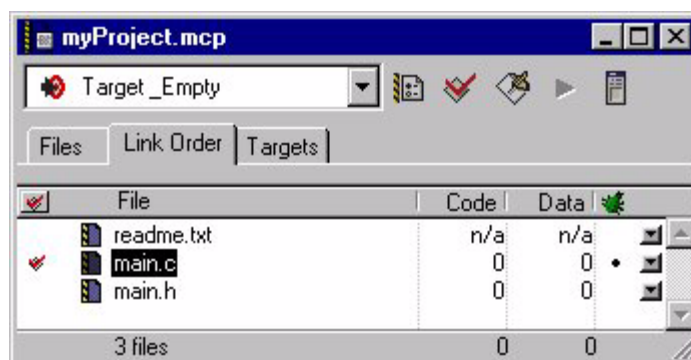
Create a group as follows:

1. Make this menu selection: **Project > Create Group...**
The Create Group window displays.
2. Enter a name for the new group in the Create Group dialog box and click **OK**.
3. Move one or more files or groups in a project's Files View as follows:
 - a. Select the files or groups to be moved
 - b. Drag the selected files or groups to their new location in the Project window.
A focus bar (an underline) indicates where the selected files will be moved when you release the mouse button.
 - c. When the focus bar is at the desired file or group position, release the mouse button.
The selected files or groups are moved to the new position

Link Order View

The Project window **Link Order** view (Figure 3.3) shows information about how the IDE will compile or link the final output file for the project's current build target.

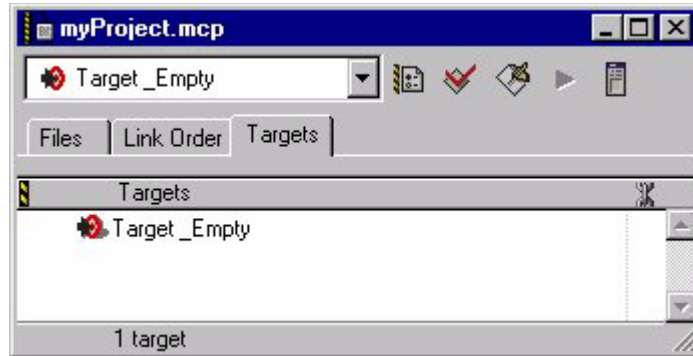
Figure 3.3 Project Window Link Order View



Targets View

The Project window **Targets** view (Figure 3.4) shows information about the active build target build target dependencies, and link-compatible build targets.

Figure 3.4 Project Window Targets View



Adding Files to your Project

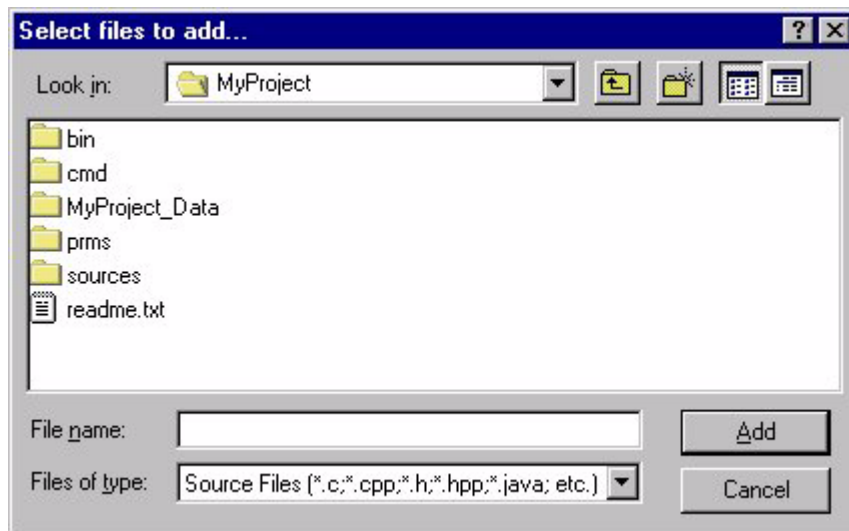
Insert files to your project as follows:

1. Make this menu selection:

Project>Add Files

The **Select files to add...** dialog box displays.

Figure 3.5 Select Files to Add... Dialog Box



- a. To select multiple files, press the **Control** key while clicking the file names in the dialog box.
 - b. To select a contiguous group of files, click the first file name in the group; then press the **Shift** key and click the last file in the group.
2. When you have selected the files you wish to add, click **Add**.

If your project contains multiple build targets, a prompt asks you to select the build targets to which you want the files added. There may be a delay while the IDE locates the selected files and adds them to your project.

Clicking **Cancel** closes the dialog box without adding any files to the project.

NOTE File extensions have to be available in **File Mappings**

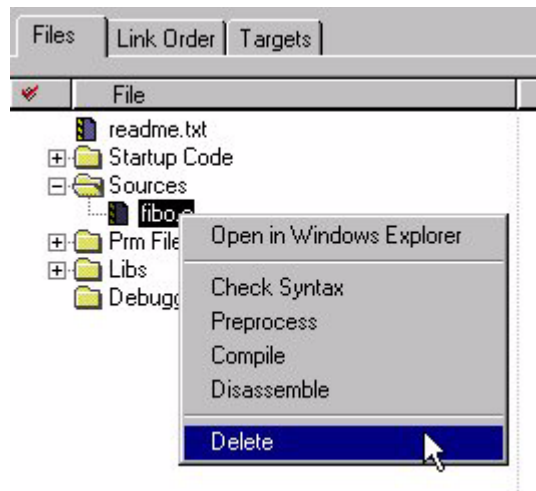
You can also add library (.lib) files to your project. In this case, the library is linked with your application. You don't have to specify it in your linker parameter (.prm) file.

Removing Files from your Project

Remove files from your project as follows:

1. Select the desired files in the Project Windows
2. Click the right mouse button and select **Delete** from the pop-up menu (Figure 3.6).

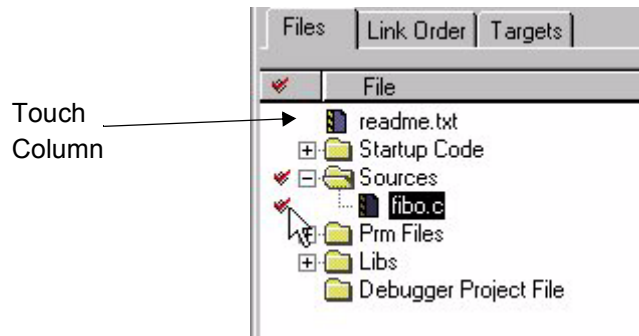
Figure 3.6 Removing Files From a Project



Touching Files

To select and mark files to be compiled, use the Touch column at the left of the Project window (Figure 3.7).

Figure 3.7 Touching Files from a project



NOTE If the CodeWarrior IDE does not recognize file changes, it may not automatically recompile all files required, which makes the Touch column feature useful.

There are three possible ways to make sure files get compiled:

- Click in the Touch column beside the file name in the Project window's Files view. A check displays in the Touch column next to the file name. This check indicates that the file will be recompiled the next time you build your project.
- Select the **Touch** command from the Interface pop-up menu.
- Click the check icon at the top of the Touch column. The IDE resynchronizes the state of the files in the project depending on the dates they were last modified.

To unmark files so that they are not compiled, click again in the Touch column left of the file name, or choose Untouch from the Interface pop-up menu.

NOTE Clicking the check icon at the top of the Touch column touches all files in the entire project.

Creating a Project

This chapter describes how to build, make, and debug a project using project stationery.

This chapter contains the following sections:

- [Building a Project](#)
- [Making a Project](#)
- [Simulating and Debugging a Project](#)

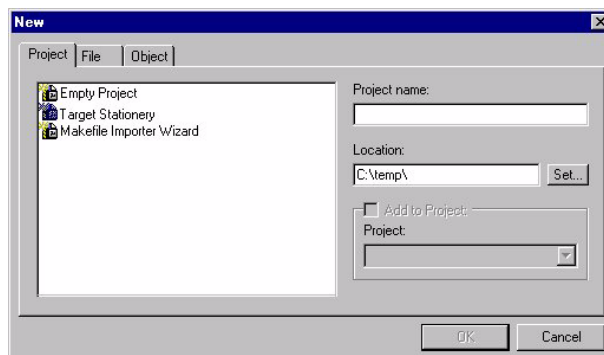
Building a Project

Build a project as follows:

1. Select **File > New** from the CodeWarrior pulldown menus.
The **New** window displays (Figure 4.1).

NOTE The New window may look different on your screen, depending on installed targets.

Figure 4.1 New Window



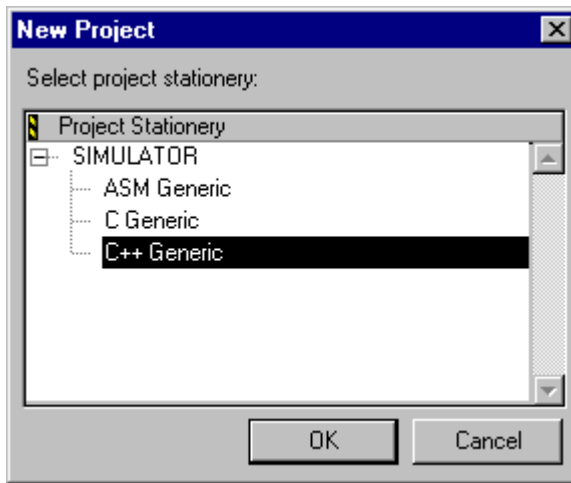
2. Click the **Project** tab to display the **Project** panel (Figure 4.1).
3. Select **Target Stationery** from the window list.

4. Type a name for the new project in the **Project Name** field.
5. Specify a path to save the project in the **Location** field.
6. Click **OK** to confirm your settings.

The **New Project** dialog displays (Figure 4.2).

NOTE The New Project window may look different on your screen, depending on installed targets.

Figure 4.2 **New Project Window**



NOTE If you save a project, it receives an **.mcp** extension.

7. Select the desired stationery file from the **Project Stationery** list.
8. Expand the list items and choose the desired target interface from the sublist.
9. Click **OK**

The IDE automatically sets up the project.

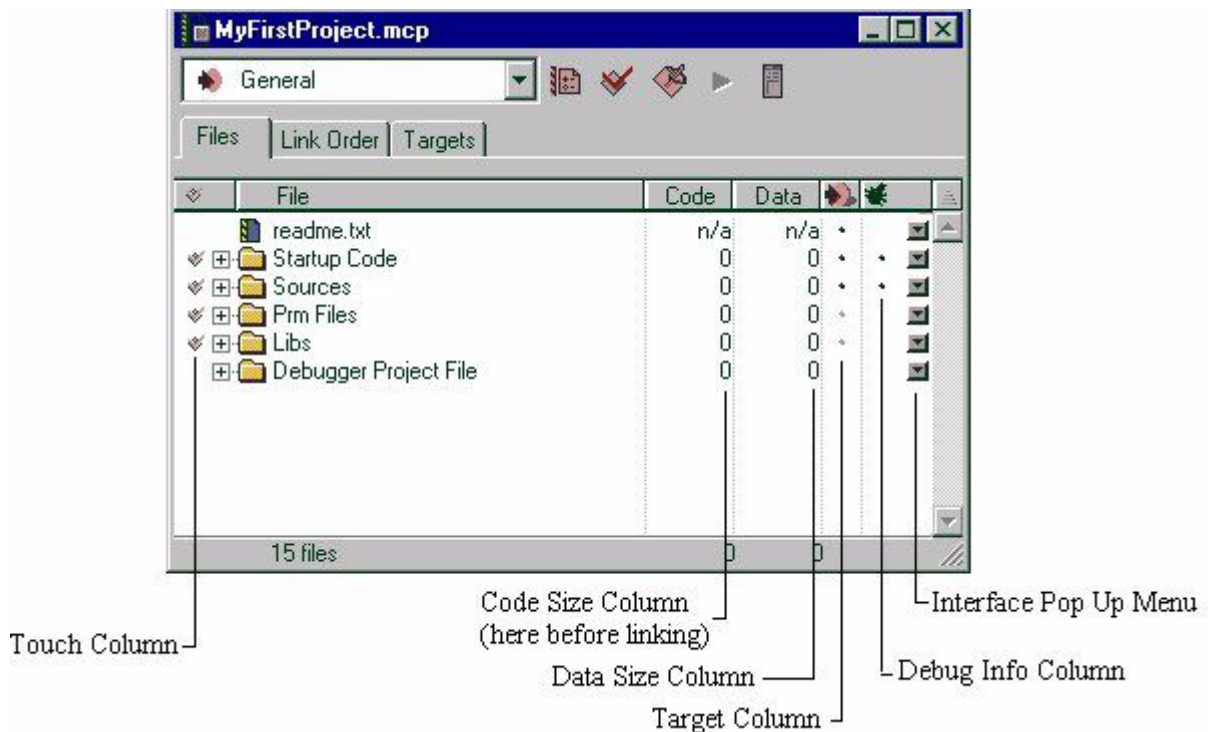
Making a Project

Make a project as follows:

1. Click the **Files** tab in the **Project** window.

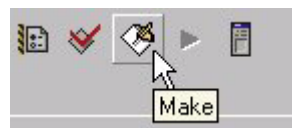
The **Files** panel (Figure 4.3) displays. The **Files** panel contains all the files in your project.

Figure 4.3 Project Window Files Panel



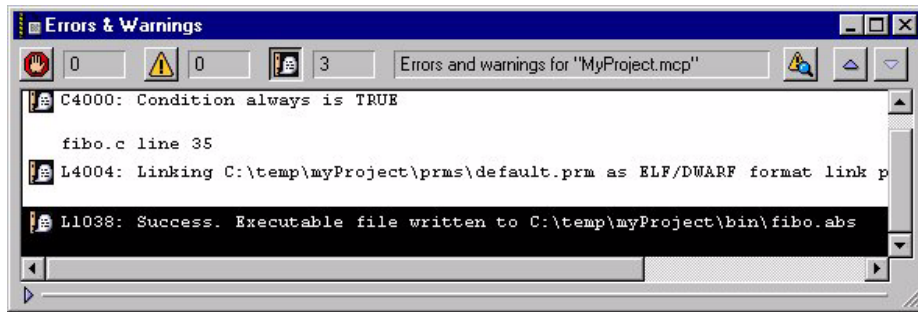
2. Click the **Make** button (Figure 4.4) in the **Project** window's toolbar to compile and link your project.

Figure 4.4 Make button



The **Error & Warnings** window (Figure 4.5) displays, containing the results of the build process.

Figure 4.5 Errors & Warnings Window



Simulating and Debugging a Project

Use the absolute file created with the **Make** command for simulation and debugging. Simulate and debug your project as follows:

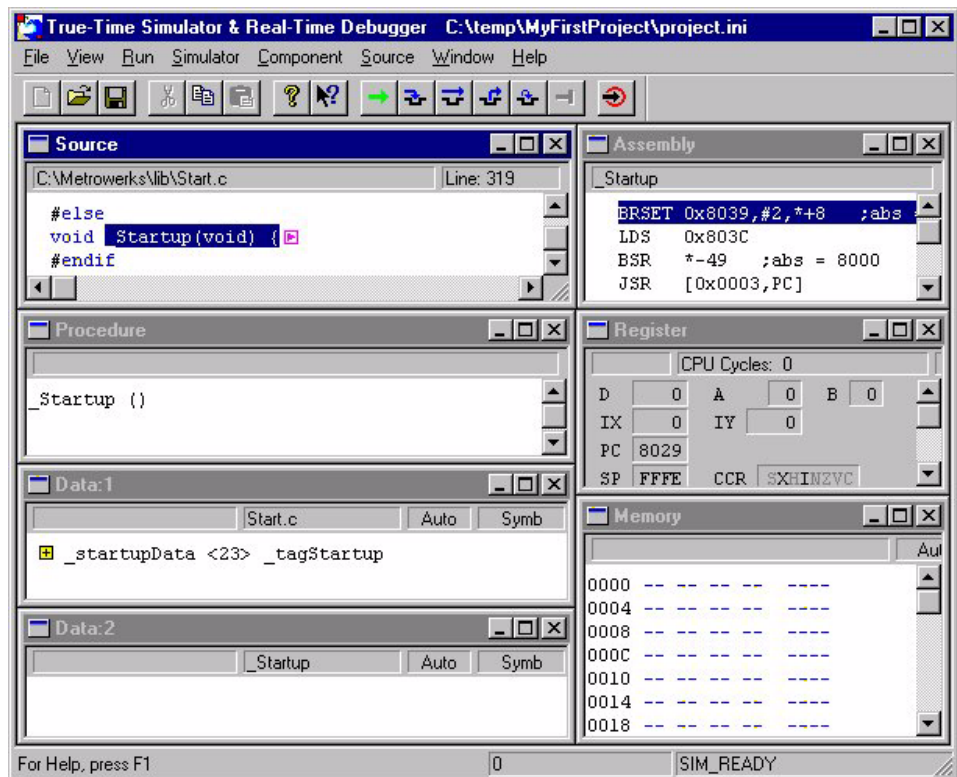
Click the **Debug** button (Figure 4.6) in the **Project** window to start the debugger.

Figure 4.6 The Debug Button



The Debugger Main Window (Figure 4.7) displays. The IDE loads your application.

Figure 4.7 The Debugger Main Window



3. Point to a C statement in the **Source** component of the Simulator/Debugger and press the right mouse button.
A pop-up menu displays.
4. Select **Set Breakpoint** from the pop-up menu.
You have just set a breakpoint for the corresponding C statement.
5. Click the **Run** button in the toolbar.
The application runs until it encounters the breakpoint.

Specifying Target Settings

This chapter shows how to make target settings when you create a project using stationery.

Depending if you are using HIWARE Technology or Metrowerks Technology, the settings may be different.

This chapter contains the following sections:

- [Target Settings Panel](#)
- [Access Paths Panel](#)
- [Build Extras Panel](#)
- [Runtime Settings Panel](#)
- [Files Mappings Panel](#)
- [Source Trees Panel](#)
- [Hiware Build Tools](#)
- [Metrowerks Build Tools](#)

Target Settings Panel

Display the **Target Settings** panel as follows:

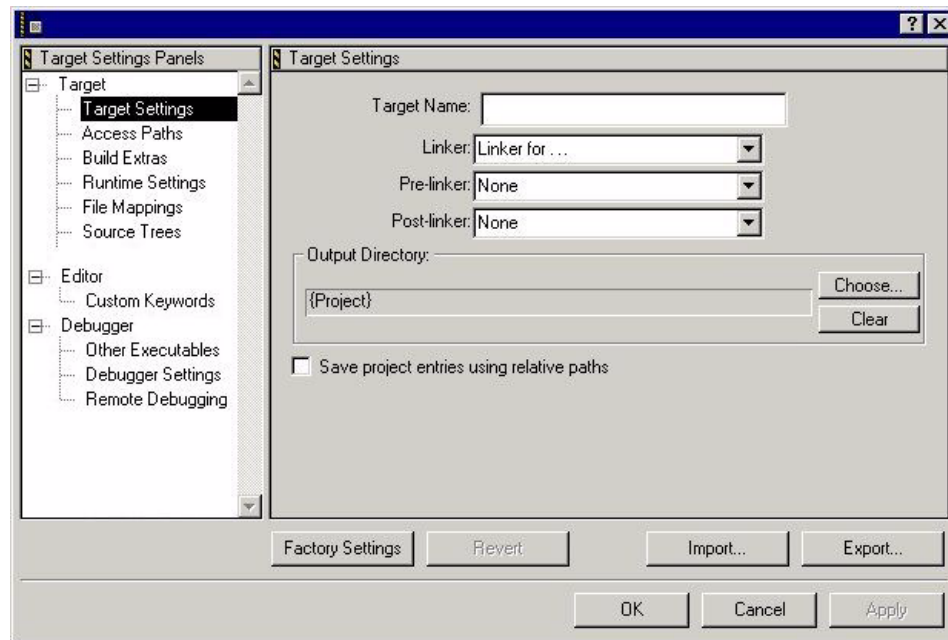
1. Click the Targets General Settings button (Figure 5.1) in the **Project** window toolbar.

Figure 5.1 Targets General Settings Button



The **Target Settings** panel displays in the **Project** window (Figure 5.2).

Figure 5.2 Target Settings Panel



2. Set up the linker for the selected CPU in the **Target Settings** panel (Figure 5.2).

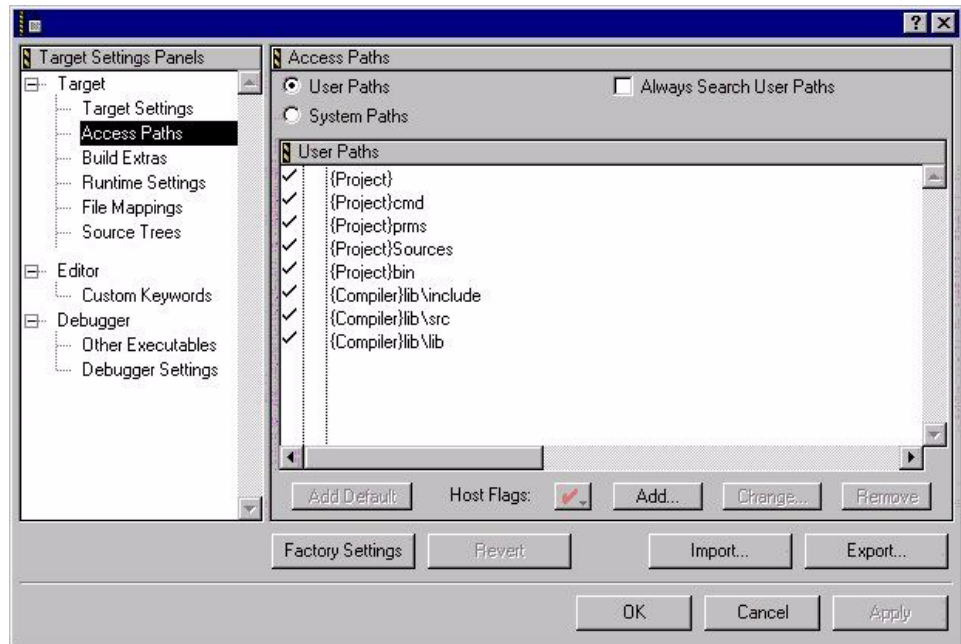
Depending on the installed CPU targets, you can choose from the linkers in the linker drop box. If you choose a linker, use the linker to build an absolute (.abs) file.

You can also select a libmaker. By setting a libmaker, the build target will be a library (.lib) file.

Access Paths Panel

Display the **Access Paths** panel (Figure 5.3) by clicking **Access Paths** in the **Target Settings Panels** list.

Figure 5.3 Access Paths Panel

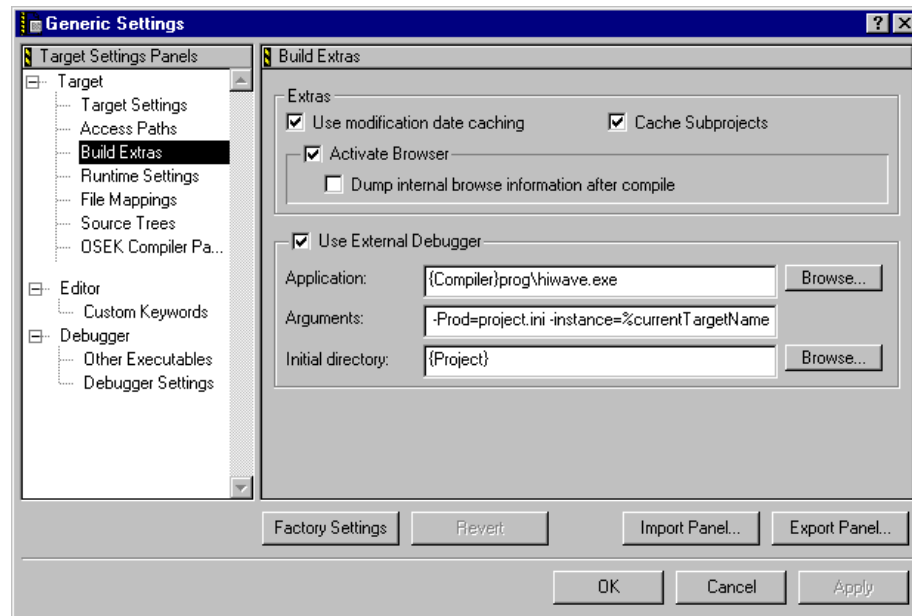


Use the **Access Paths** panel to define additional paths for the CodeWarrior Integrated Development Environment (IDE) to search while compiling and linking your project.

Build Extras Panel

Display the **Build Extras** panel (Figure 5.4) by clicking **Build Extras** in the **Target Settings Panels** list.

Figure 5.4 Build Extras Panel



Use the **Build Extras** panel to specify build extras options.

Table 5.1 explains the controls in this panel.

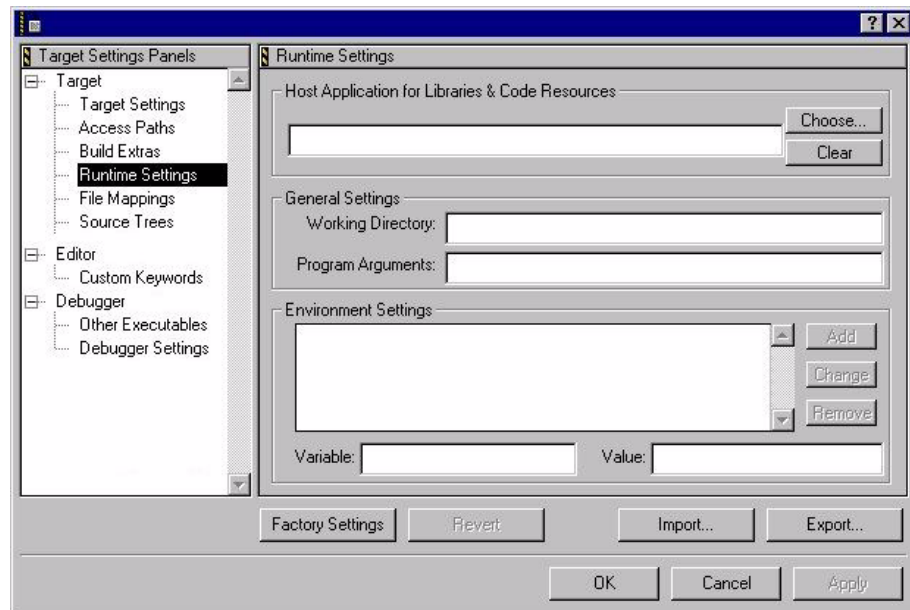
Table 5.1 Build Extras Panel Controls

Control Name	Function
Activate Browser checkbox	If checked, compiler and assembler generate browse information.
Use External Debugger checkbox	If checked, external debugger is selected. You must specify a relative or absolute path with the Browse... button.
Arguments text box	Specifies additional command-line arguments to pass to the debugger. Besides normal arguments, you can specify the following % macros: <pre> %sourceFilePath : File path of source %sourceFileDir : Directory of source %sourceFileName : File name of source %projectFilePath : Directory of the project source %projectFileDir : Directory of the project %projectFileName : Name of the project %targetFilePath : Path of the absolute file %targetFileDir : Directory of the absolute file %targetFileName : Name of the absolute file </pre>

Runtime Settings Panel

Display the **Runtime Settings** panel (Figure 5.5) by clicking **Runtime Settings** in the **Target Settings Panels** list.

Figure 5.5 Runtime Settings Panel



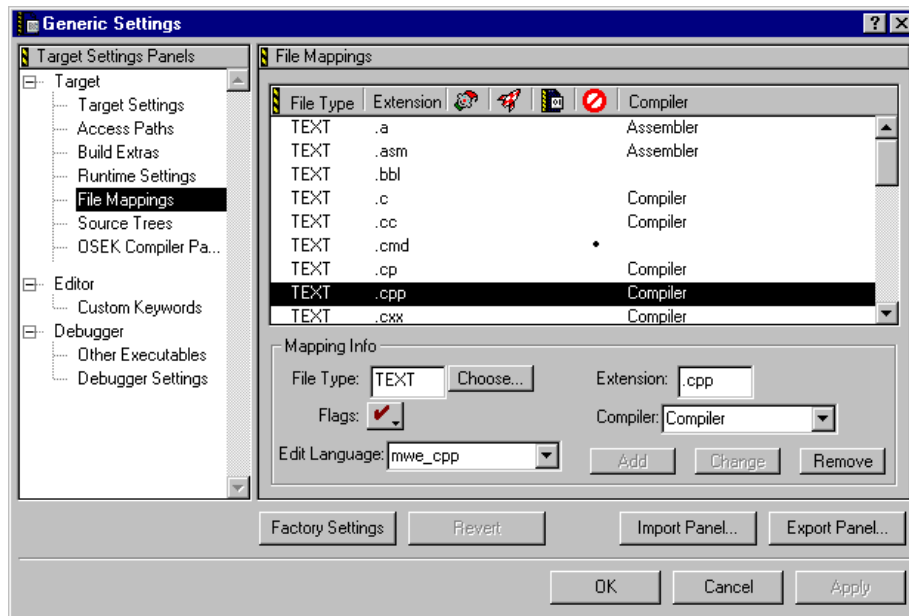
Use the **Runtime Settings** panel to get arguments that your application must have in order to work correctly after installation.

NOTE You do not need this panel for embedded applications.

Files Mappings Panel

Display the **File Mappings** panel (Figure 5.6) by clicking **File Mappings** in the **Target Settings Panels** list.

Figure 5.6 File Mappings Panel



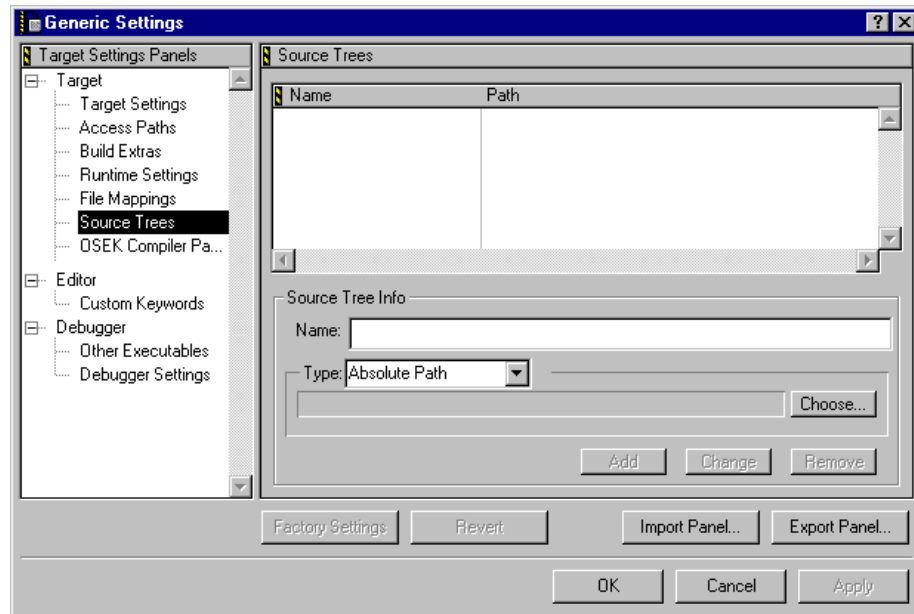
Use the **File Mappings** panel (Figure 5.6) to list current file mappings and their settings, including:

- File type
- Extensions
- Flag
- Associated compiler
- Edit Language (used for syntax coloring)

Source Trees Panel

Display the **Source Trees** panel (Figure 5.7) by clicking **Source Trees** in the **Target Settings Panels** list.

Figure 5.7 Source Trees Panel



Use the **Source Trees** panel to define access paths and build-target output in terms of source trees. You can also create a source tree to define global source trees (root paths) for use in your projects.

With this approach, you can share projects across various hosts. You need only make minor changes to the source-tree paths to maintain project functionality.

Hiware Build Tools

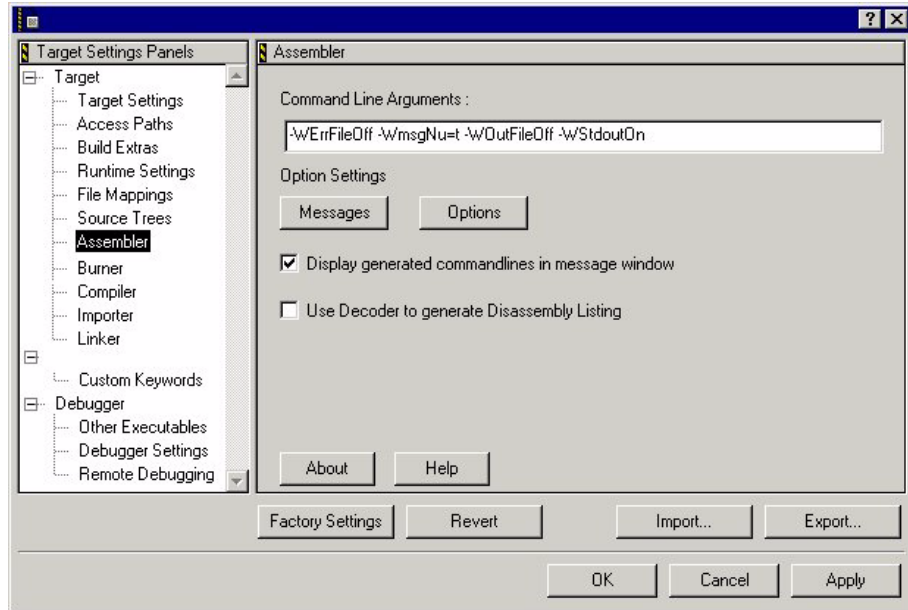
The following panels customize the way the IDE generates code using HIWARE build tools:

- [Assembler Panel](#)
- [Burner Panel](#)
- [Compiler Panel](#)
- [Importer Panel](#)
- [Linker Panel](#)
- [Libmaker Panel](#)

Assembler Panel

Display the **Assembler** panel (Figure 5.8) by clicking **Assembler** in the **Target Settings Panels** list.

Figure 5.8 Assembler Panel



Use the **Assembler** panel to control assembler behavior. Table 5.2 explains the controls in this panel.

Table 5.2 Assembler Panel Controls

Control Name	Function
Messages button	Displays the Assembler Message Settings dialog (Figure 5.9). Use this dialog to filter messages.
Options button	Displays the Assembler Options Settings dialog (Figure 5.10). Use this dialog to add, delete, or modify options.
Display generated command lines in message window checkbox	If checked, only Warning, Information, and Error messages display in the Errors & Warnings window. The complete command line also passes to the tool.
Use Decoder to generate Disassembly Listing checkbox	If checked, the external decoder produces a disassembly listing.
About button	Displays status and version information.
Help button	Displays helpfiles.

Figure 5.9 Assembler Message Settings Panel

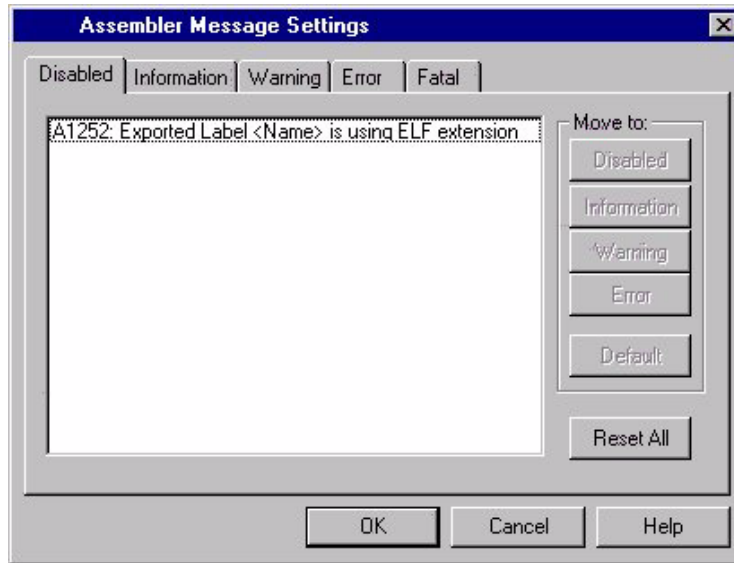
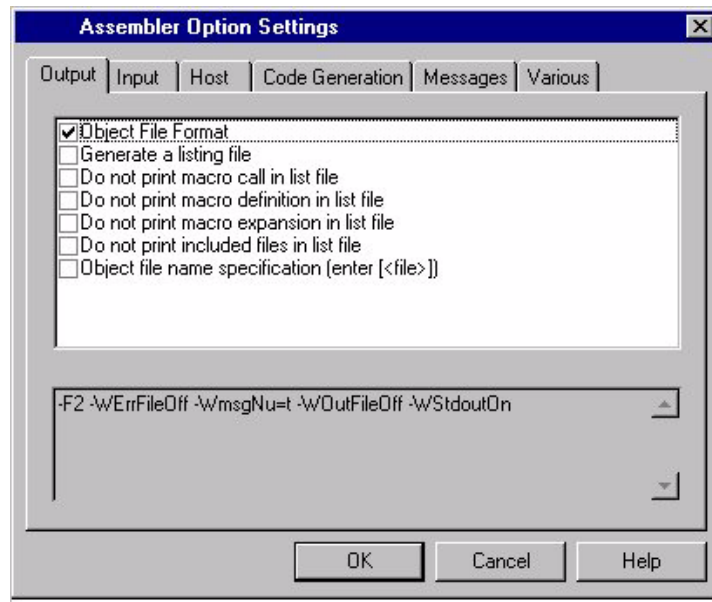


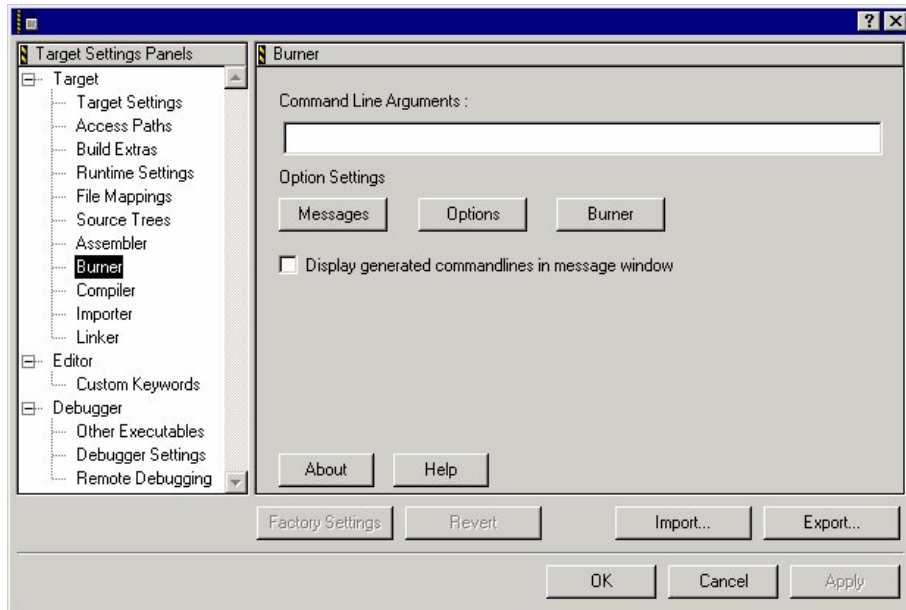
Figure 5.10 Assembler Options Settings Panel



Burner Panel

Display the **Burner** panel (Figure 5.11) by clicking **Burner** in the **Target Settings Panels** list.

Figure 5.11 Burner Panel



Use the **Burner** panel to control the burner to generate S-Records, Intel, or binary files.

The **Burner** plug-in has special functionality: in the **File Mappings** Panel (Figure 5.6), the *.bbl (batch burner language) files are mapped to the **Burner** plug-in. Whenever a *.bbl file is in the project file, the *.bbl file is processed during the post-link phase using the settings in the **Burner** panel.

Table 5.3 explains the controls in this panel.

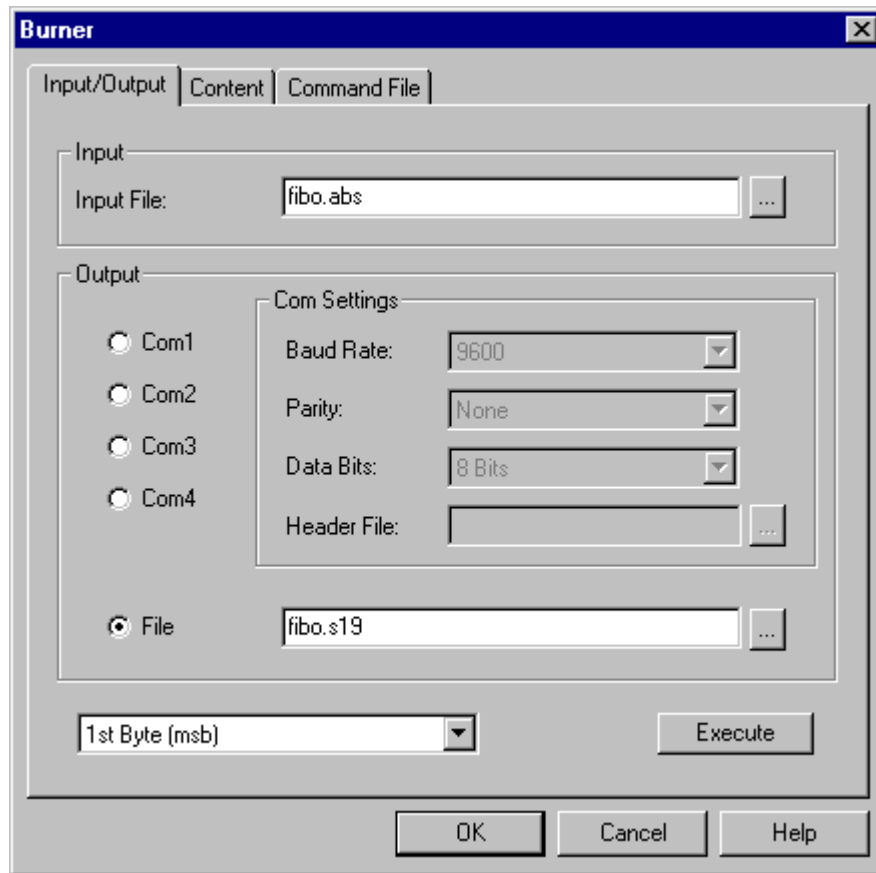
Table 5.3 Burner Panel Controls

Control Name	Function
Command Line Arguments text field	Displays command-line options. You can add, delete, or modify these options by hand or by using the Messages , Options , Type Sizes , and Smart Sliders buttons.
Messages button	Displays the Assembler Message Settings dialog (Figure 5.9). Use this dialog to filter messages.
Options button	Displays the Assembler Options Settings dialog (Figure 5.10). Use this dialog to add, delete, or modify options.
Burner button	Displays Burner dialog (Figure 5.12).
Options button	Displays Options dialog.

Table 5.3 Burner Panel Controls (continued)

Control Name	Function
Display generated command lines in message window checkbox	If checked, only Warning, Information, and Error messages display in the Errors & Warnings window. The complete command line also passes to the tool.
About button	Displays status and version information.
Help button	Displays helpfiles.

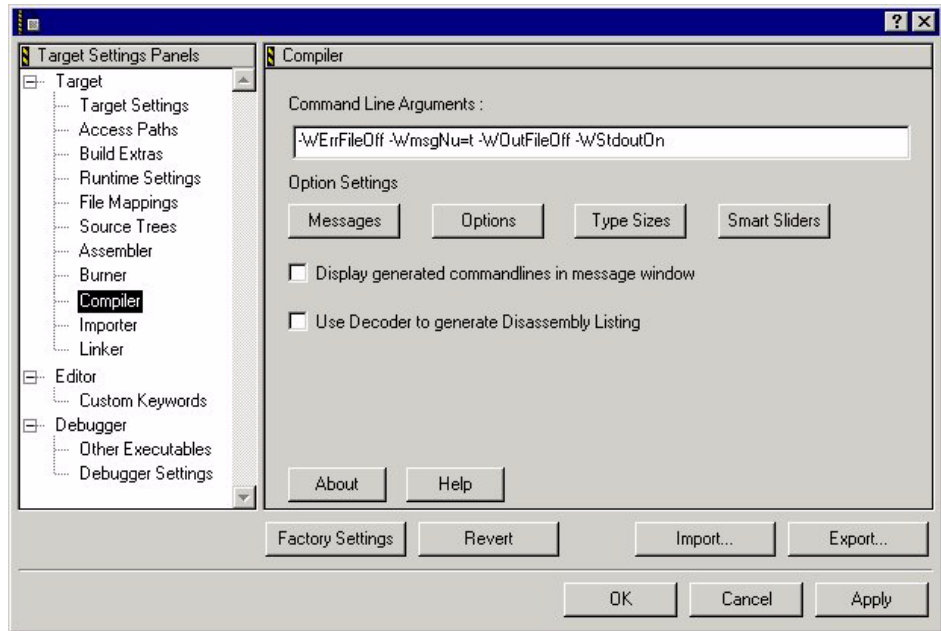
Figure 5.12 Burner Dialog Box



Compiler Panel

Display the **Compiler** panel (Figure 5.13) by clicking **Compiler** in the **Target Settings Panels** list.

Figure 5.13 Compiler Panel



Use the **Compiler** panel to control compiler behavior. Table 5.4 explains the controls in this panel.

Table 5.4 Compiler Panel Controls

Control Name	Function
Command Line Arguments text field	Displays command-line options. You can add, delete, or modify these options by hand or with the Messages , Options , Type Sizes , and Smart Sliders buttons.
Messages button	Displays the Assembler Message Settings dialog (Figure 5.9). Use this dialog to filter messages.
Options button	Displays the Assembler Options Settings dialog (Figure 5.10). Use this dialog to add, delete, or modify options.
Type Sizes button	Displays the Standard Type Size dialog box (Figure 5.14).
Smart Sliders button	Displays the Compiler Smart Control dialog box (Figure 5.15).
Display generated command lines in message window checkbox	If checked, only Warning, Information, and Error messages display in the Errors & Warnings window. The complete command line also passes to the tool.
Use Decoder to generate Disassembly Listing checkbox	If checked, the external decoder produces a disassembly listing.

Table 5.4 Compiler Panel Controls (continued)

Control Name	Function
About button	Displays status and version information.
Help button	Displays helpfiles.

Figure 5.14 Standard Types Settings Dialog

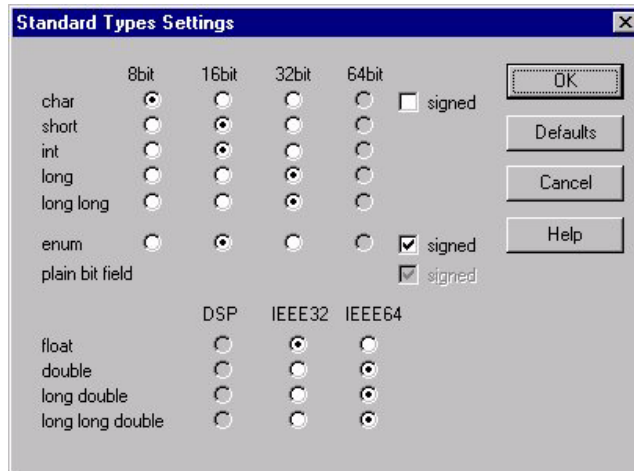
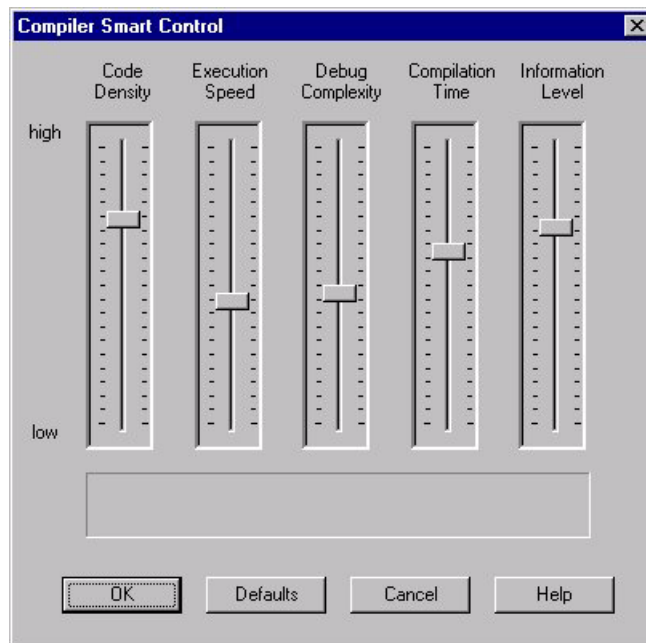


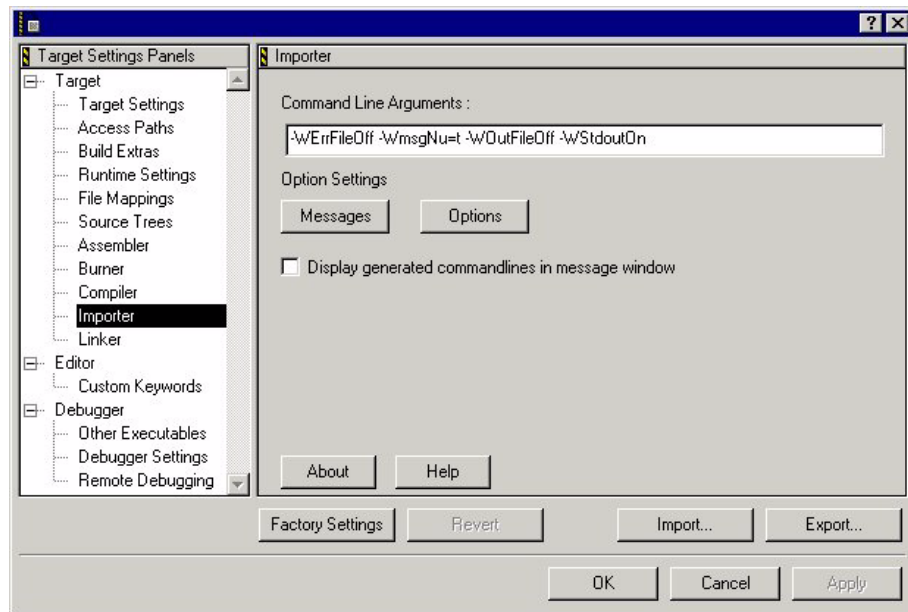
Figure 5.15 Compiler Smart Control Dialog



Importer Panel

Display the **Importer** panel (Figure 5.16) by clicking **Importer** in the **Target Settings Panels** list.

Figure 5.16 Importer Panel



Use the **Importer** panel for decoding objects, as well as absolute or library files. Table 5.5 explains the controls in this panel.

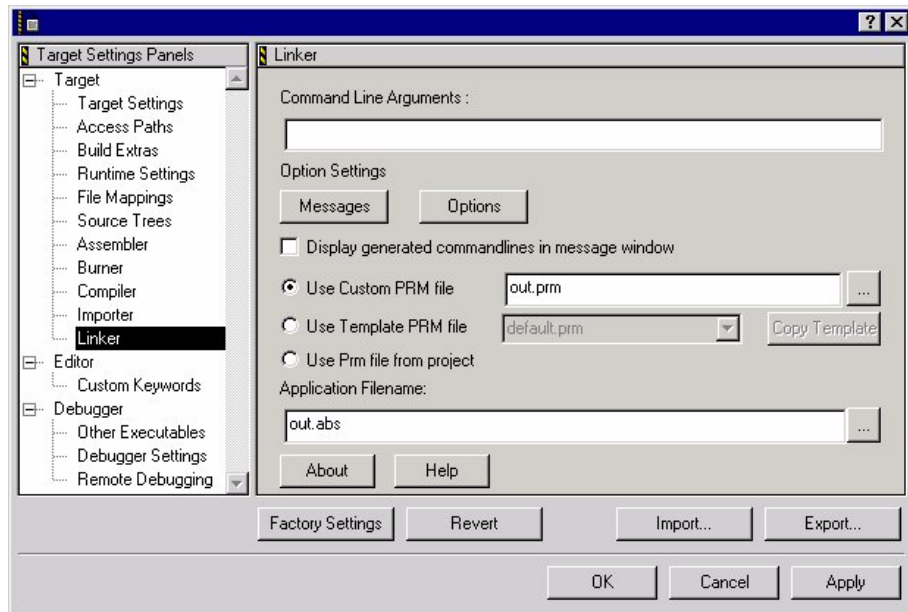
Table 5.5 Importer Panel Controls

Control Name	Function
Command Line Arguments text field	Displays command-line options. You can add, delete, or modify these options by hand or by using the Messages , Options , Type Sizes , and Smart Sliders buttons.
Messages button	Displays the Assembler Message Settings dialog (Figure 5.9). Use this dialog to filter messages.
Options button	Displays the Assembler Options Settings dialog (Figure 5.10). Use this dialog to add, delete, or modify options.
Display generated command lines in message window checkbox	If checked, only Warning, Information, and Error messages display in the Errors & Warnings window. The complete command line also passes to the tool.
About button	Displays status and version information.
Help button	Displays helpfiles.

Linker Panel

Display the **Linker** panel (Figure 5.17) by clicking **Linker** in the **Target Settings Panels** list.

Figure 5.17 Linker Preference Panel



Use the **Linker** panel to control the linker. The **Linker** panel displays only if you select a linker in the **Target Settings** panel (Figure 5.2).

Table 5.6 explains the controls in this panel.

Table 5.6 Linker Panel Controls

Control Name	Function
Command Line Arguments text field	Displays command-line options. You can add, delete, or modify these options by hand or by using the Messages , Options , Type Sizes , and Smart Sliders buttons.
Messages button	Displays the Assembler Message Settings dialog (Figure 5.9). Use this dialog to filter messages.
Options button	Displays the Assembler Options Settings dialog (Figure 5.10). Use this dialog to add, delete, or modify options.
Display generated command lines in message window checkbox	If checked, only Warning, Information, and Error messages display in the Errors & Warnings window. The complete command line also passes to the tool.

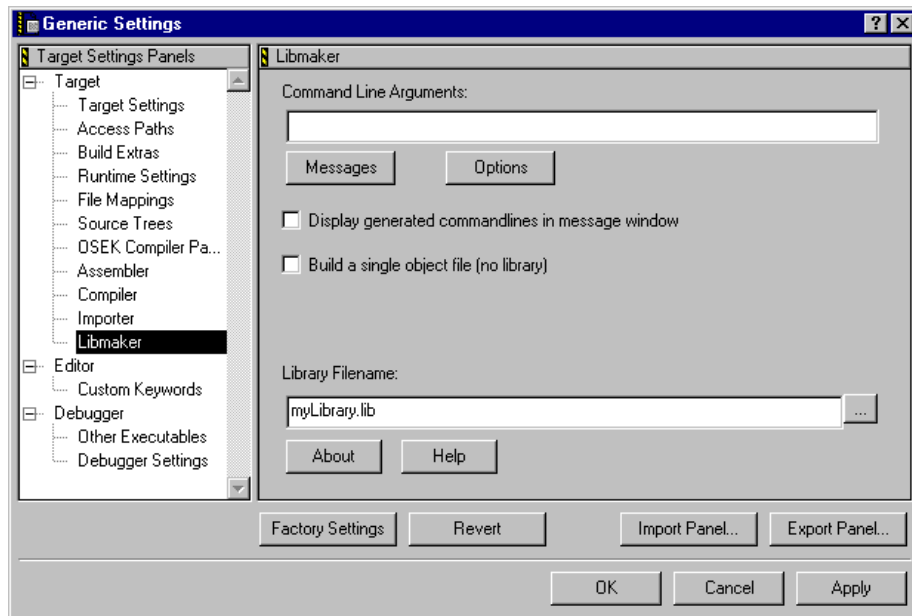
Table 5.6 Linker Panel Controls (continued)

Control Name	Function
Use custom PRM file radio button	If selected, you can specify a custom linker parameter file in the edit box on the right. To browse for a file, click the Browse... button.
Use template PRM file radio button	If selected, you can choose a premade prm file located in the templates directory: (c:\metrowerks\templates\<>target>\prm) You can also use the Copy Template button to copy the prm file into your project so that a local copy is available.
Use PRM file from Project radio button	If selected, the linker PRM file present in the project is used.
About button	Displays status and version information.
Help button	Displays helpfiles.

Libmaker Panel

Display the **Libmaker** panel (Figure 5.18) by clicking **Libmaker** in the **Target Settings Panels** list.

Figure 5.18 Libmaker Panel



Use the **Libmaker** panel to control the libmaker for building libraries, as well as adding and removing files from libraries. The **Libmaker** panel

Specifying Target Settings

Hiware Build Tools

displays only if you select **Libmaker** in the **Target Settings** panel (Figure 5.2).

Table 5.7 explains the controls in this panel.

Table 5.7 Libmaker Panel Controls

Control Name	Function
Command Line Arguments text field	Displays command-line options. You can add, delete, or modify these options by hand or by using the Messages , Options , Type Sizes , and Smart Sliders buttons.
Messages button	Displays the Assembler Message Settings dialog (Figure 5.9). Use this dialog to filter messages.
Options button	Displays the Assembler Options Settings dialog (Figure 5.10). Use this dialog to add, delete, or modify options.
Display generated command lines in message window checkbox	If checked, only Warning, Information, and Error messages display in the Errors & Warnings window. The complete command line also passes to the tool.
Build a single object file (no library) checkbox	If checked, only a single object file is generated (instead of a library). Useful to generate startup object files or to create an absolute assembly application.
Library File Name text box	Specify the output file name. Use the Browse... button if necessary.
Use template PRM file radio button	If selected, you can choose a premade prm files located in the templates directory: (c:\metrowerks\templates\<<target>\prm) You can also use the Copy Template button to copy the prm file into your project so that a local copy is available.
About button	Displays status and version information.
Help button	Displays helpfiles.

Metrowerks Build Tools

The following panels customize the way the IDE generates code using Metrowerks build tools:

- [Language Settings Panels](#)
- [Code Generation Panels](#)
- [Linker Panels](#)

NOTE This manual does not explain the panels in detail. To learn more about these panels, please refer to the CodeWarrior manuals.

Language Settings Panels

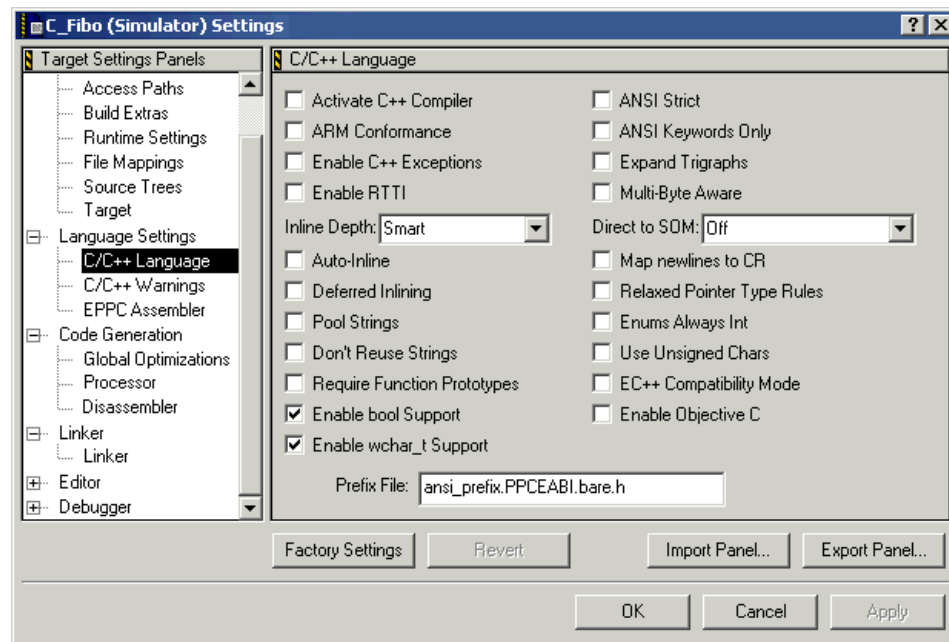
The Language Settings panels that appear in the **Target Settings Panels** list depend on the linker chosen for the build target in the Target Settings panel. The panels are organized into the followings groups:

- [C/C++ languages panel](#)
- [C/C++ warnings panel](#)
- [Assembler panel](#)

C/C++ languages panel

This panel, shown in Figure 5.19, provides language-specific configuration settings.

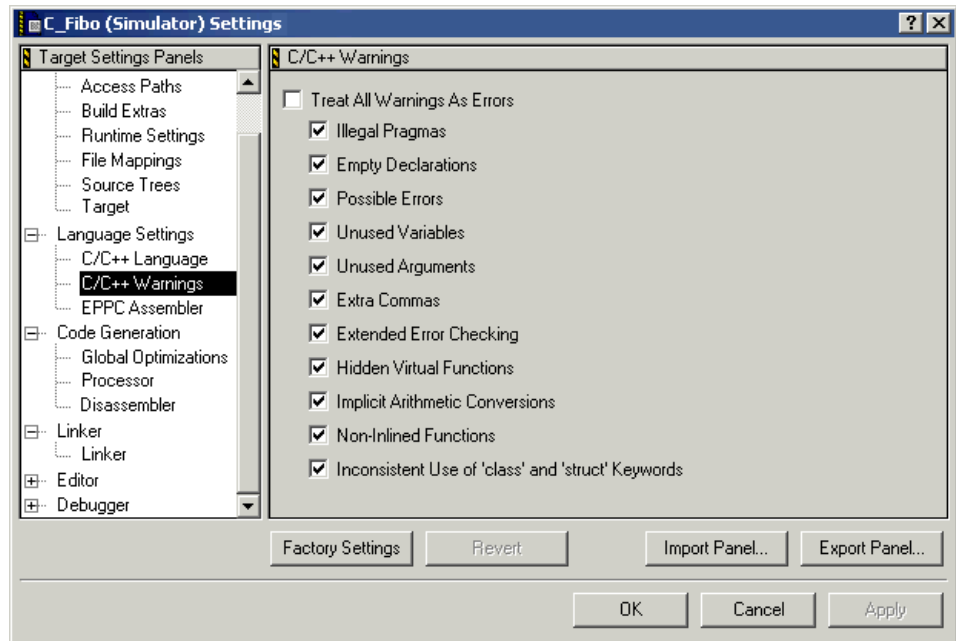
Figure 5.19 C/C++ languages panel



C/C++ warnings panel

The C/C++ Warnings panel shown in Figure 5.20 provides control over the languages warning messages and the strictness of their implementation.

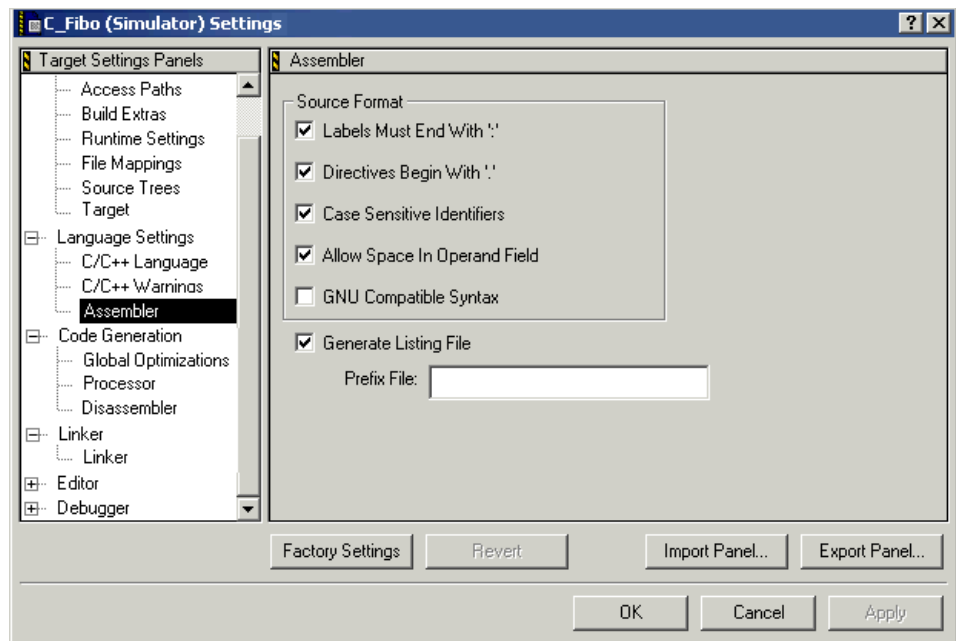
Figure 5.20 C/C++ Warnings panel



Assembler panel

The Assembler panel shown in Figure 5.21 provides assembler configuration settings.

Figure 5.21 Assembler panel



Code Generation Panels

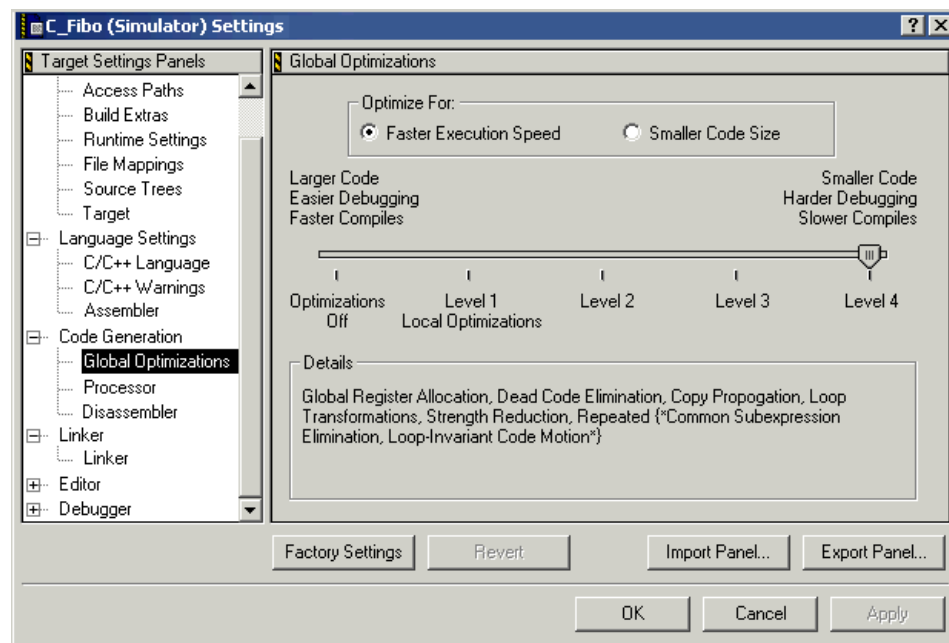
The linker selected in the **Target Settings Panels** determines the Code Generation panels displayed in the Target Settings Panels list. The Code Generation panels are organized into the followings groups:

- [Global optimizations panel](#)
- [Processor panel](#)
- [Disassembly panel](#)

Global optimizations panel

The global optimizations panel, shown in Figure 5.22 is used to configure how the compiler rearranges its object code to produce smaller and faster-executing object code.

Figure 5.22 Global optimizations panel



Processor panel

The processor panel provides processor-level code generation settings.

Disassembly panel

The disassembly panel provides control over the information displayed when disassembling code modules.

Linker Panels

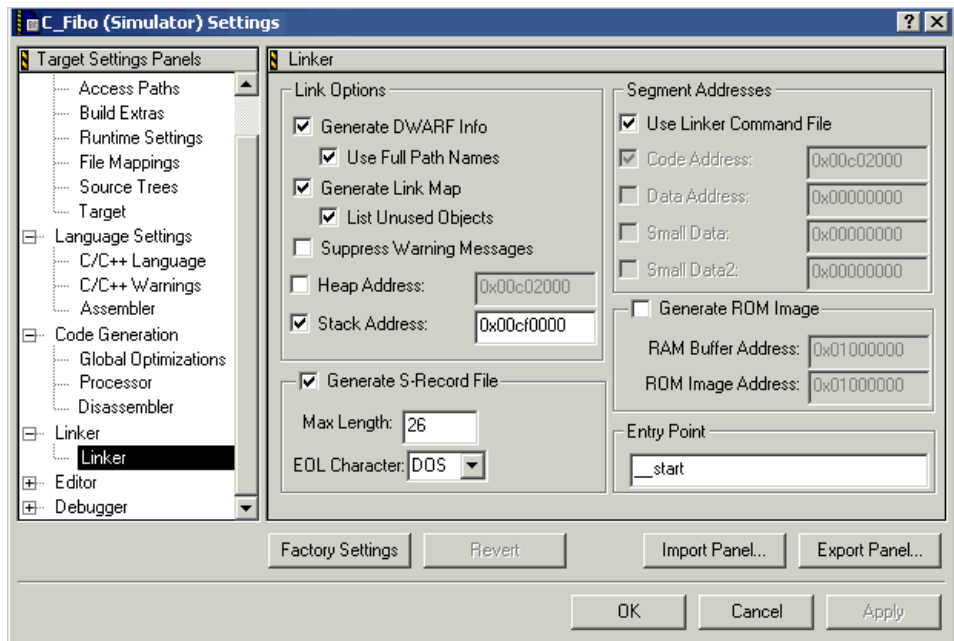
The linker selected in the **Target Settings Panels** determines the Linker panels displayed in the Target Settings Panels list. The Linker panels is organized into the following group:

- [Linker panel](#)

Linker panel

The linker panel, shown in Figure 5.23 provides control over the actions of the chosen linker. The IDE calls the linker to combine the object code into a single executable file.

Figure 5.23 Linker panel



Specifying Target Settings

Metrowerks Build Tools

Technical Support

This chapter provides different methods for you to receive technical support for the CodeWarrior Interactive Development Environment (IDE). Whichever method you choose, we at Metrowerks listen and act.

This chapter contains the following sections:

- [E-mail](#)
- [Fax](#)
- [Mail](#)
- [Internet](#)

E-mail

The best way to get technical support is through e-mail. You can attach examples to the email using a compression utility or simply uuencode.

The email addresses are:

EUROPE: support_europe@metrowerks.com

USA: support@metrowerks.com

ASIA/PACIFIC: j-emb-sup@metrowerks.com

Fax

You can fax your problem to the following numbers:

EUROPE: Fax: +41 61 690 7501

USA: Fax: +512 997 4901

ASIA/PACIFIC: +: 3-3780-6092

Mail

To reach technical support by normal mail, use the addresses below:

EUROPE: **Metrowerks Europe**, Riehenring 175, CH-4058 Basel,
Switzerland

USA: **Metrowerks**, 9801 Metric Blvd, Austin, TX 78758

ASIA/PACIFIC: **Metrowerks Japan - Metrowerks Co., Ltd.** - Shibuya
Mitsuba Building 5F - Udagawa-cho 20-11, Shibuya-ku
-Tokyo 150-0042 Japan

Internet

For the latest updates and product-enhancement information, point your browser to: <http://www.metrowerks.com>

Index

Symbols

%projectFileDir 28
%projectFileName 28
%projectFilePath 28
%sourceFileDir 28
%sourceFileName 28
%sourceFilePath 28
%targetFileName 28
%targetFilePath 28
*.bbl 35
.abs 26
.lib 26
.mcp 20

A

About 33, 36, 38, 39, 41, 42
Access Paths 27
Access Paths Panel 27
Adding Files to your Project 16
Assembler Panel 33
Assembler panel 45

B

Bin 10
browse information 28
Build Extras Panel 27
Burner Preference Panel 34

C

C/C++ languages panel 44
C/C++ warnings panel 44
CodeWarrior Help 10
CodeWarrior Manuals 10
Command Line Arguments 35, 37, 39, 40, 41, 42
Compiler Preference Panel 36
Copy Template 41, 42

D

Disassembly panel 47

Display generated command lines in message window 39, 41, 42

E

Examples 10

F

Files Mappings Panel 30

G

Global optimizations panel 46

H

HIWARE Build Tools 32

I

IDE

Files view 14
Link Order 14
projects 11
targets 11
Targets view 15

Importer Preference Panel 39

J

j-emb-sup@metrowerks.com 49

L

Lib 10
lib<target>cprm 10
libmaker 26
Libmaker Preference Panel 41
library 26
Linker Preference Panel 40

M

Mapping 30
Messages 33

O

Options 35, 37, 39, 40, 42
Other Metrowerks Tools 10

P

Processor panel 46
project
 Making 21

R

Release Notes 10
Removing 17
Requirements 7
Runtime Settings 29

S

Setup 8
Simulating/Debugging 22
Simulator/Debugger 23
source tree 31
Source Trees Panel 31
Stationery 10
stationery 12
Support 49
support@metrowerks.com 49
support_europe@metrowerks.com 49

T

Technical Support 49
Template 10
tool tip 12
Touching 17

U

Use custom PRM file 41, 42
Use Decoder to generate Disassembly
 Listing 28, 33, 36, 37, 39, 40, 41, 42