

## **Tutorial Introduction**

**PURPOSE:**

- Provides an overview of the Metrowerks CodeWarrior toolset and explains how the toolset is organized and operates

**OBJECTIVES:**

- Describe the CodeWarrior integrated development environment (IDE)
- Identify the components of the CodeWarrior build system.
- Identify the available targets of the IDE.
- Define CodeWarrior terminology.

**CONTENT:**

- 10 pages
- 3 questions

**LEARNING TIME:**

- 5 minutes

Welcome to this introduction to the Metrowerks CodeWarrior software development tools. This tutorial explains how CodeWarrior combines the programs required to write software into one seamless, integrated work environment. CodeWarrior tools run on several different platforms and can generate machine code for a variety of desktop and embedded systems.

Upon completion of this tutorial, you'll be able to describe the components of the CodeWarrior build system and the general organization of the software package. You'll also be able to define a target and describe the operating systems that will host the software.

## What is an IDE?

- The IDE consists of tools that are used throughout the software development process.

Open Architecture	Project Manager	Full Featured Editor
Code Navigation System	Graphical Source Debugging	Register/Bit Level Support
GNU Tools Support	Graphical Differencing	Kernel Awareness
Graphical and Command Line Build System		

- The tools are integrated fully and seamlessly.
- A single environment is provided for software development.
  - consistent operation
  - move among all the tools freely
  - consistent look and feel, across environments and architectures

In the CodeWarrior toolset, the Editor, Compiler, Linker, Debugger, and other software modules operate within an Integrated Development Environment, or IDE. The IDE oversees the control and execution of the tools. It provides an interface to the tools that is consistent and predictable.

The operation of the development tools is seamless. There are no separate programs to run. You can readily switch between the Editor, Compiler, and Debugger by simply pointing and clicking or using a menu command.

## IDE Components

- Project Manager - scalable and unbounded
- Full Featured Editor - any supported language
- Search Engine - any text file or group of files
- Source Browser - any supported language
- Build System - plug-in compilers, assemblers, linkers, command line tools
- Debugger - any supported language and target
  - Displays source, machine code disassembly, or both

Next, let's discuss the IDE components.

The Project Manager can handle projects as small as "Hello World!" and as large and complex as CodeWarrior itself. You can use CodeWarrior to manage any kind of project, not just source code.

The Editor is a complete editing environment that can handle any text file. Customizable syntax coloring clearly identifies keywords for all supported languages.

The Search Engine searches any text file or group of files.

The Source Browser is a powerful navigation and editing tool for all supported languages.

The Build System includes all compilers, linkers, and assemblers for a given target platform. Its plug-in architecture allows third parties to write their own tools that integrate with the CodeWarrior IDE. This mechanism is similar to how third-party plug-ins add capabilities to a Web browser.

The Debugger enables you to view your source code, either as is or in assembly. Using the Debugger, you can set breakpoints, watch specific variables change, and move line-by-line through your program.


### CodeWarrior Host Environments

- CodeWarrior runs on the following operating systems:  
Windows 95/98, 2000, NT, XP, ME  
Solaris            Mac OS            Linux
- The user interface operates identically on all hosts.
- Project files and source files operate on all hosts.
- Some project data is not transferable.
  - precompiled headers
  - compiled data such as object code, debug information, browser information

CodeWarrior runs on many different operating systems including Windows, Mac OS, Solaris, and Linux, with others to be added in the near future. The interface looks and operates the same on all environments. There are, of course, minor differences in screen appearance among host operating systems, but the functional capabilities are effectively the same.

One of the greatest benefits of CodeWarrior's multi-platform approach is that your project files are completely transferable from one host operating system to another. For example, if you were writing a project for Motorola's StarCore architecture using the Windows hosted tools, you could transfer your project and files to a Solaris system for final testing without needing to convert your project in any way.

## CodeWarrior Development Studio



The screenshot shows the CodeWarrior Development Studio website interface. The main content area is titled "Embedded" and lists various products supported by the development studio. The products are categorized by manufacturer and architecture.

Manufacturer	Product
ARM	ARM
Motorola	68K
Motorola	Coldfire
Motorola	Suite56 DSP563xx/DSP566xx
Motorola	SG800/LE Hybrid
Motorola	HC05
Motorola	HC(S)08
Motorola	HC11
Motorola	HC(S)12
Motorola	HC16
Motorola	M*CORE
Motorola	MPC500 / MPC5500
Motorola	MPC5200
Motorola	MPC60x/MPC7xx/MPC74xx
Motorola	PowerQUICC I/II
Motorola	PowerQUICC III
Motorola	StarCore
Motorola	StarQUICC+
STMicroelectronics	ST7
Sun Microsystems	Java Card

This list shows some of the Metrowerks products. Not all Development Studios are available on all Hosts.

Check the Metrowerks website for the complete list.

## What Is a Target?

- A platform target is the chip or operating system selected for development.
  - processor examples are HC(S)08, ColdFire, and StarCore
  - operating system examples are Palm OS, Windows, and Linux
- A build target is a file, library, or build option for a program.
  - build target examples are debug, release, optimize
- A build target is analogous to a makefile.

Next, let's take a detailed look at terminology, for instance, targets. In CodeWarrior, the word "target" can refer to two different things. First, the platform target is the specific operating system or chip architecture you want your code to run on. Second, the build target refers to the specific libraries and source files you use and what options you set.

A CodeWarrior project contains all of your source code, libraries and compiler/linker settings. A single CodeWarrior project can have multiple build targets. Each build target can be set up completely differently. For example, you may have separate debug and release builds, using the exact same source files, but different compiler settings. The debug build might have all compiler optimizations turned off, debug code generation turned on, and could use Debugger-friendly libraries to construct the executable code. The release build might have all compiler optimizations turned on, debug code generation switched off, and would use optimized libraries to generate the executable code.

A build target functions like a Makefile in that it specifies the files, the compiler and linker settings, and the libraries necessary to generate a program.

### Question

Which of the following operations does the IDE manage? Click on your choice.

- A) Source code writing and editing
- B) Compiling and linking
- C) Debugging
- D) Tracking files that were changed and must be recompiled
- E) All of the above

Let's complete this tutorial with a few questions.

What operations does the IDE manage?

**Answer:**

The IDE automatically manages all of the operations listed.

### Question

Which of the following platforms will host the IDE? Click on your choice.

- A) Mac OS only
- B) Windows only
- C) Mac OS, Windows, and Solaris
- D) Linux
- E) C and D

Which of the following platforms will host the IDE? Click on your choice.

Answer:

The IDE runs on virtually every desktop platform.

### Question

What is a CodeWarrior build target? Click on your choice.

- A) Settings and options used to generate a program.
- B) Embedded processor selected for development.
- C) RTOS selected for development.

What is a CodeWarrior build target?

Answer:

A build target is the files, the compiler and linker settings, and the libraries necessary to generate a program. It is analogous to being a graphical makefile.

## Tutorial Completion

- Access Paths define the path to the file.
  - runs on most desktop platforms
  - combines all development tools into an integrated environment
  - automatically manages and stores the compiler, linker, and build settings
- The IDE assists through the entire development cycle –
  - writing code: Editor, Code Browser
  - generate application: Compiler, Linker, and Libraries
  - debug application: Graphical Debugger, Register/Bit Level Support

Let's review what we discussed in this tutorial.

The CodeWarrior IDE operates just about anywhere. It can be used to develop applications for a wide variety of embedded processors.

The IDE hosts a suite of tightly-coupled tools that work in unison to speed the code development cycle. The advantage of using CodeWarrior, is that you can switch between tools immediately. For example, within the Debugger use the Code Browser or use the editor to set breakpoints for the Debugger.