

Guide to CodeWarrior Java for CS100 and CS211

Hal Perkins

Department of Computer Science, Cornell University

Spring 1998

1 CodeWarrior Basics

The main software for CS100 and CS211 programming projects is Metrowerks' CodeWarrior. The latest version, CodeWarrior Pro 2, is installed in the CIT labs for Spring 1998. CodeWarrior Pro 2 supports Sun's Java version 1.1.3.

CodeWarrior Pro is available for both Macintosh and Windows 95/NT, and it works the same in both environments, except for some minor differences due to the Windows and Macintosh user interfaces.

CIT has installed CodeWarrior in all of its labs that have machines that can support it. The only difference between the CIT version and the regular one is that some additional files added for simple Java programs, particularly for CS100. If you have a copy of CodeWarrior you will want to get these files. See the instructions in Section 3 at the end of this handout.

In addition to Java, the CodeWarrior environment can be used to run C, C++, and Pascal programs. In this guide we restrict ourselves to Java.

We assume you already know how to use Macintosh or Windows to, for example, create and print a document using a word processor.

In this handout, a `)` is used to specify menu commands. For example, "select PROJECT)RUN" means select command RUN from menu PROJECT.

1.1 Projects and Programs

CodeWarrior uses two major kinds of files: *source files* contain Java program code, while *project files* contain additional information CodeWarrior needs to run the program. Each program has one project file and one or more source files.

By convention, CodeWarrior project files have names that end in ".mcp" (standing for Metrowerks CodeWarrior Project). Java source file names must end in ".java."

1.1.1 Opening an Existing Project

For some programming assignments, we will give you a folder containing a project file and one or more source files. If you need to create a project from scratch, see the instructions later in this missive.

To use a project that you get from us, or to continue working on one you created earlier, double-click the project file icon. This will automatically start CodeWarrior if it is not already running. *Don't* double-click a .java file icon. This may open a different programming environment (particularly likely under Windows); further, a project file must be open to run a program.

1.1.2 Starting CodeWarrior By Itself

You can start CodeWarrior without using a project file.

Macintosh. In the CIT labs, CodeWarrior is buried inside several layers of folders. If the hard drive window is not open, double-click the hard drive icon. Then open the following succession of folders to find the CodeWarrior application: Applications; CodeWarrior; Metrowerks CodeWarrior. (Use Option-double-click to reduce clutter on the desktop by closing unneeded windows as you pass through them.) Start CodeWarrior by double-clicking the CodeWarrior IDE application icon.

Windows. Click START, then select PROGRAMS) CODEWARRIOR)CODEWARRIOR IDE to start CodeWarrior. (On some machines, the CODEWARRIOR IDE may be in a menu named PROGRAMS) METROWERKS CODEWARRIOR.)

When CodeWarrior finishes launching, its menu bar will appear at the top of the screen. In the public labs, a dialog box may be displayed to remind you that this copy of CodeWarrior can be used only in the CIT labs. Click OK to continue.

1.1.3 Opening Projects Inside CodeWarrior

To open a project from within CodeWarrior, use FILE)OPEN... to get a standard file dialog box. The current folder will probably be the folder containing CodeWarrior. Move to the folder containing the desired project file and click Open.

1.1.4 The Project Window — Opening Files

When a project file is open, CodeWarrior displays the file's *project window*. This window lists all of the files and libraries included in the project. To open an editor window for a source file, double-click the file name in the project window. (You may have to first click on the triangle to the left of a group name to see the names of files in that group.)

1.2 Editing Programs

The CodeWarrior editor works much like any other Macintosh or Windows text editor. Source files are displayed in *editor windows*. The mouse is used to select text by clicking and dragging. Use the CUT, COPY, and PASTE commands under EDIT to move text around. UNDO can reverse the effect of many commands if you make a mistake.

The most noticeable difference from most editors is the way CodeWarrior displays programs on a color monitor. Program code, keywords, and comments are displayed in different colors. This effect is called *syntax coloring*. Another visual effect that the editor provides is *balancing*. Whenever you type), }, or], the editor will momentarily highlight the corresponding symbol (, {, or [. If the highlighted left bracket is not the one you expected, or if the editor beeps because it can't find a matching bracket, check for extra or missing brackets.

There are editor commands to help indent code. The Tab key increases the indentation level, while Backspace decreases it. When you hit Return, the editor starts the new line with the same indentation as the previous one. To change the indentation of whole sections of code, select the lines and use EDIT)SHIFT LEFT or EDIT)SHIFT RIGHT.

To save source files, use FILE)SAVE. CodeWarrior saves project files automatically; you do not need to explicitly save them.

IMPORTANT: Be sure to save your work frequently, particularly if you are creating a large program or have made many changes. Don't wait until you are done for the day. A power failure or machine crash could wipe out hours of work if you haven't saved files recently. ■

1.3 Running Programs

To compile and run your program, select PROJECT)RUN (Command-r (Mac) or Shift-F5 (Windows)).

If CodeWarrior detects errors in your program, it will display appropriate messages. To get to the trouble spot in the code, double-click an error message.

If no errors are found, CodeWarrior will start the Java Virtual Machine and instruct it to run your program. A console window will appear (this is where text output is written). If your program uses other windows, say for drawing or text input, these will also appear. If they are hidden behind other windows, you can bring them forward by clicking on any visible part of the window or (Windows only) clicking on the window name at the bottom of the screen.

Note: The Macintosh version of CodeWarrior Pro 2 sometimes displays the message "Apple MRJ could not be found. Metrowerks VM used as Java VM" the first time you try to run a new Java project. Just click OK and the program should run properly. ■

When the last statement in a Java program has been executed, the Java environment continues running to let you examine the program's output and any other program windows remaining on the screen. When you are done you should quit the Java runtime environment by selecting FILE)QUIT (Macintosh) or clicking the console window's Close box (Windows).

Note: If you forget to quit the Java environment, the next time you select PROJECT)RUN, the old version of the program will resume instead of a new one. Quit the Java environment and select RUN again to run the new version. ■

1.3.1 Errors, Bugs, and Crashes

Debugging. As you develop your programs you are almost certain to encounter errors, including incorrect results, nonterminating execution, or (rarely) machine crashes. When this happens, the Metrowerks debugger can be used to help locate problems. The debugger will be described in a later handout.

Macintosh Freezes. Occasionally an error in a program will cause the Macintosh to hang — everything stops and the mouse and keyboard don't have any effect. In this case, first try to force the program to quit by typing Option-Command-Esc. If that works, save any open files and restart the machine to reinitialize the Mac.

If the forced quit doesn't work, the only solution is to restart the machine. On older Macs the restart button is the leftmost of the two located on the left side of the front panel. If your Mac does not have

a restart button, try holding down the **Control** and **Command** keys and then pressing the **power-on** button at the top of the keyboard. When you release these keys the machine should restart.

Warning: Don't restart unless there is no other option. You will have to restart all applications you were running and you will lose changes to documents that were modified but not saved. ■

Windows Freezes. While Windows doesn't lock up very often, we have seen situations where the Java environment starts behaving strangely, usually after some other problem has occurred. The best solution is to select **START)SHUTDOWN...** then select **RESTART THE COMPUTER** to reinitialize Windows.

1.4 Printing

Once you have a correctly running program you will want to print a copy of the program and its output, perhaps because you need to hand it in for credit.

1.4.1 Printing in CIT Labs

Printers in the labs are usually shared among three or four computers. Be sure nobody else is using the printer; then, turn the rotary switch connected to the printer to select your machine. Check that the printer is on and ready.

You can also use NetPrint to print on a laser printer (there is a charge for this). See the CIT documentation for instructions.

1.4.2 Printing Programs and Other Files

Java source programs and other files can be printed from CodeWarrior the same way that text files are printed in a word processor. Select the CodeWarrior window containing the file, then select **FILE)PRINT...**. This brings up the standard print dialog box. If you're using an inkjet or dot-matrix printer, use either **Faster** or **Best** quality — **Draft** quality can be almost illegible. Click **Print** to print the file.

CodeWarrior can also print part of a file. Select the lines you want to print, then check the **Print Selection Only** box in the **PRINT...** dialog.

1.4.3 Printing Program Output

Java and its class libraries were designed for use in web applets, which run in a browser, or embedded computers that control gadgets like appliances or smart credit cards. In these environments, there is little need for printed output, and Java provides little direct support for it.

It is, however, possible to get printouts of the console window or a picture of the entire screen indirectly by creating files containing the desired output.

1.4.4 Printing Console (Text) Output

The basic idea is to select the text in the console window and copy it to a text editor, then use the text editor's print command to print it. The output will look best if a monospaced font like Courier or Monaco is used in the text editor.

Macintosh. Select the console window, click at the bottom of the output, and drag the mouse to the top of the window. If text has scrolled off the top of the window, drag the mouse to the window's title bar and hold the mouse button down to scroll the window and select the text. Then select **EDIT)COPY**. Paste the text into an editor window (a new CodeWarrior editor window will do fine) and print it.

Windows. Click the right mouse button on the console window's title bar. Select **EDIT)MARK** from the pop-up menu that appears. Click at the beginning of the text you want to copy, press and hold **shift**, then click at the end of the text. Click the right mouse button in the title bar again and select **EDIT)COPY** in the popup menu. Click on a text editor window (a new CodeWarrior editor window, for example) and use **EDIT)PASTE** to copy the text. Print the window.

1.4.5 Printing Screen Snapshots

Both Macintosh and Windows have the ability to create a file containing a picture of the entire screen. Among other things, this provides a convenient way to print graphics output. First, arrange the windows on the screen so the ones you want to include in the printout are visible, then do the following.

Macintosh. Press **Command-Shift-3** to take a snapshot of the screen. The output will be placed in a file named **Picture 1** in the main window of the hard disk. If a file named **Picture 1** already exists, the file will be named **Picture 2**, **Picture 3**, etc.

To view the snapshot, double-click the picture file to launch **SimpleText** to display the file. To print the snapshot, select **FILE)PRINT...** in **SimpleText**.

Windows. Press **Alt-Print Scrn** to take a snapshot of the currently active window, or **Print Scrn** to take a snapshot of the entire screen. Open **PAINT** from menu **START)PROGRAMS)ACCESSORIES**. Select **EDIT)PASTE** to copy the screen image to **PAINT**, then print it.

1.5 Backups

Remember to save your work frequently, not just when you are done for the day. You should also keep a second floppy disk as a backup in case your first disk becomes corrupted or unreadable. You should copy your main floppy disk to the backup regularly.

IMPORTANT: You **must** keep your files on your own disks. User files are removed from the public machines regularly. ■

1.6 Quitting

When you are done, leave the CodeWarrior environment by selecting FILE)QUIT. If there are any open files that you haven't saved yet, you will be asked if you want to save them now. Click Save or Don't Save.

2 CodeWarrior Tips

For some programming assignments you will receive pre-built project and source code files. But if no files are supplied, or if you want to run additional programs for practice, you will need to know how to create your own projects.

2.1 Creating New Projects

To create a new project from scratch, start CodeWarrior and select FILE)NEW PROJECT. . . . Use the window that appears to select appropriate *project stationary*. The different project stationary choices specify appropriate options, libraries, and sample code files needed to create different kinds of programs. CodeWarrior comes with several project stationary choices. In the CIT labs, we have added a few extra choices to create simple Java programs for Computer Science classes. (See the end of this hand-out for information about adding them to your copy of CodeWarrior.)

In the project stationary window, click the triangle to the left of the word Java. This should display a list of available Java stationary. If you are creating a simple Java program that does text input and output, select CUCS Java Application. If you want to generate graphical output, select CUCS Java Graphics Application. Click OK.

After selecting appropriate stationary, use the dialog box that appears to give the project file a name and choose where to put it. Type a name ending in `.mcp` for the project. Move to the disk or folder where you want to save the new project and click OK. This creates a new folder in the location you

selected. The folder contains the new project file, copies of other files included with the stationary, and a folder containing data used by CodeWarrior for its internal bookkeeping. Once the files and folders are created, CodeWarrior will open the new project file.

IMPORTANT: When you save a new file or folder, *always* check that you are saving it where you want it. The default folder initially displayed in the dialog box is usually somewhere in the folders containing CodeWarrior itself, which is almost certainly *not* where you want to store your project. ■

2.2 Working With Projects

Once you've created or opened a project, you can modify the source file(s) supplied with the project stationary to create your program. You may also want to add additional source files to the program or remove some of the ones that are already there.

2.2.1 Creating New Java Source Files

Java programs are built from a collection of classes. Each class should normally be stored in a file whose name is the class name followed by `.java` (e.g., class DrawPicture should be stored in file DrawPicture.java).

To create a new file, select FILE)NEW, enter some Java code in the new window, then save the file with the appropriate name. Double-check to be sure you save it in the correct folder, which should usually be the one that contains the project file.

2.2.2 Adding Java Files to Projects

Before a Java file can be used as part of a program, it must be added to the project window — just creating a `.java` file isn't enough. You can add to the project files that you create or files containing Java code that you get from us or other sources.

Adding open windows. If the Java file is open in a CodeWarrior editor window, bring the window to the front, then select PROJECT)ADD WINDOW.

Adding other files. To add other Java files to the project, select PROJECT)ADD FILES. . . . This brings up a dialog window, which you use to pick the files. Find the Java file(s) by moving to the correct folder(s) if necessary. Select the file name(s) in the top window and click Add. The selected file names should appear in the list at the bottom of the window.

You can select as many files from as many folders as you like. When you're finished, click Done to add them to the project.

2.2.3 Rearranging Project Files

The files in the project window can be arranged in any order you want. To move a file, drag it to the desired place.

2.2.4 Selecting Method main

Every Java application program begins execution in method `main` in one of its classes. Since every Java class can have a `main` method (often a good place to put code to test the class), one has to specify which `main` should be used to start the program. When you create a new Java project from stationary, this is done automatically. For example, the CUCS Java Application and CUCS Java Graphics Application stationary begin execution in method `main` of class `Application` or `GraphicsApplication` respectively.

If you want to begin execution in another class (perhaps one you've added to the project), you ***MUST!!*** change the project settings to identify the class containing the appropriate `main` method. Select `EDIT)PREFERENCES...` (shortcut: click on the leftmost square icon in the middle of the project window). This opens the preferences panel. Select `JAVA TARGET` in the list of options at the left of this window. Enter the name of the class containing the desired `main` method in field `MAIN CLASS:`. Click `Save`, then close the window.

IMPORTANT: If the `MAIN CLASS` setting is wrong, CodeWarrior may exhibit very bizarre behavior when you try to run the program. If you're lucky, you'll get an error message complaining that some class you've never heard of can't be found.

However, particularly in the current Windows version of CodeWarrior, the Java Virtual Machine may begin executing whatever code it can find, strange windows may appear on the screen, or you may get error messages from Windows itself complaining about system-level issues that seem totally unrelated to your Java program. If you see such bizarre behavior, the `MAIN CLASS:` setting is one of the first places to look for trouble. ■

2.2.5 Removing Files From Projects

You can remove files from the project window if you no longer need them. A common reason would be to remove files included with the stationary (like `Application.java` or `GraphicsApplication.java`) when you want to substitute another class containing a different `main` method.

To remove files, select the file names in the project window, then select `PROJECT)REMOVE FILES`.

2.2.6 Changing Project Settings

You may want to change some of the other CodeWarrior settings for your project. You can change cosmetic things like the editor's font, text size, and keyword colors, as well as more substantive things like compiler options. Select `PREFERENCES...` and `SETTINGS...` from menu `EDIT` to find out what's available. You can change things like the editor colors and fonts freely, but be careful about altering options you don't understand. Inappropriate changes can make it impossible to run your program.

2.2.7 Project File Compatibility

CodeWarrior Pro 2 uses a new internal format for project files. While old project files can be updated to work with this release, new project files, including ones you get from us, will not work with previous versions of CodeWarrior. We suggest you upgrade to the current version, although you can create a new project file with an older version and add to it the source files you get from us.

2.2.8 Damaged Project Files

Occasionally, usually after a particularly bad crash, a project file becomes damaged and CodeWarrior behaves strangely when using it. If the project file is badly damaged, CodeWarrior might crash when `RUN` is selected — even before the program gets a chance to crash on its own — or changes in the source program seem to have no effect on what happens when the program is executed. (But be sure this is not because you forgot to quit the Java runtime environment after your program finished.)

The first step when you suspect the project file is damaged is to force CodeWarrior to recreate some of the information in it. Select `PROJECT)REMOVE OBJECT CODE`. This command causes CodeWarrior to discard the machine-language version of the program and some other information in the project file. This information will be recreated the next time you try to run the program.

If this doesn't solve the problem, the only solution may be to discard the damaged project file and use a new one. Get a new copy of the project file, if it was supplied as part of an assignment, or create a new project from stationary; add the program's other `.java` files to the project; check the `MAIN CLASS:` option; and delete from the project any stationary files you don't need.

2.3 Editing Hints

The CodeWarrior editor has many features to help edit programs. Here are some useful ones.

2.3.1 Opening Related Files

There are several ways to open files associated with a project without using FILE)OPEN. . . .

To open a file whose name appears as text in an editor window, select the name of the file and select FILE)OPEN SELECTION.

If you are editing a Java class, you can display a menu of files containing other classes used by that class by pressing the button containing an **h** in the upper left corner of the editor window. Another way to do this is to press the triangle to the right of the file name in the project window. In either case, you'll see a list of files containing related classes; select a name from the list to open the file. (This doesn't work until the program has been compiled at least once. CodeWarrior builds the list of related files when it compiles a file.)

2.3.2 Searching

The editor window provides a quick way to go to the definition of any function (method) in the current file. Press the **{}** button in the upper left corner to see a list of functions in the file. Choose a function name to jump to the definition of that function.

To find text strings, use the *find window*. Select SEARCH)FIND. . . . The window that appears contains text boxes for the *find* and *replace* strings. As in any text editor, when you enter a find string and click Find, CodeWarrior selects the next occurrence of that string. Click Replace to replace it with the string in the *replace* field.

These commands provide the basic operations needed to search for text in source files. Experiment with the other commands under SEARCH as well. Some of the commands can be modified by pressing the Shift key. For example, Shift changes FIND NEXT to FIND PREVIOUS, which searches backward instead of forward through the file.

2.4 Reclaiming Disk Space

A lot of the space used on your disk isn't needed after you've finished a CodeWarrior project. Feel free to delete file JavaClasses.zip, which contains the compiled Java machine code, as well as the folder that has the same name as the project file followed by Data, which contains information used by CodeWarrior to

run the project. These can easily be recreated by running the program again.

3 CS CodeWarrior Extensions

If you have your own copy of CodeWarrior, you may want to add the CS Java Application and CS Java Graphics Application stationary to it. Point your web browser to www.cs.cornell.edu. The home pages for CS100 and CS211 should contain instructions for accessing and installing these files.

The files are also available via FTP in directory `pub/cucs_codewarrior_extras` on ftp.cs.cornell.edu.

Files are in standard formats for the web: self-extracting BinHex archives for Macintosh and zip files for Windows. Installation is simple. Unpack the files. On the Macintosh, drag the new additional stationary folders to folder (Project Stationery):Java; on Windows, the new stationary should be placed in CodeWarrior's Stationary\Java folder. Full instructions are included with the files.

4 For More Information . . .

Web pages from Sun documenting the full Java class library can be reached from the CS100 web pages. This is a convenient way to look up information about standard Java classes.

Electronic versions of CodeWarrior manuals and other documentation are included on the CodeWarrior CD-ROMs. Be aware that the CodeWarrior documentation is written *by* experienced programmers *for* experienced programmers, so it may be quite cryptic or obscure in some places.

For CodeWarrior updates and information about Metrowerks, check Metrowerks' Web site: <http://www.metrowerks.com>.

5 Getting Help

If you have questions about CodeWarrior, you should first contact staff members for your course — consultants, teaching assistants, or instructors.

If you find errors in this missive, please send email to hal@cs.cornell.edu describing the problem.

Metrowerks provides email technical support for CodeWarrior. If you think you've found a CodeWarrior bug, you can contact them directly, but it would be best to check with a member of the course staff first to verify that there really is a bug and not some other problem.