

**Chapter
2**

**USING
VISUAL C++**

*Get on the
Fast Track!*

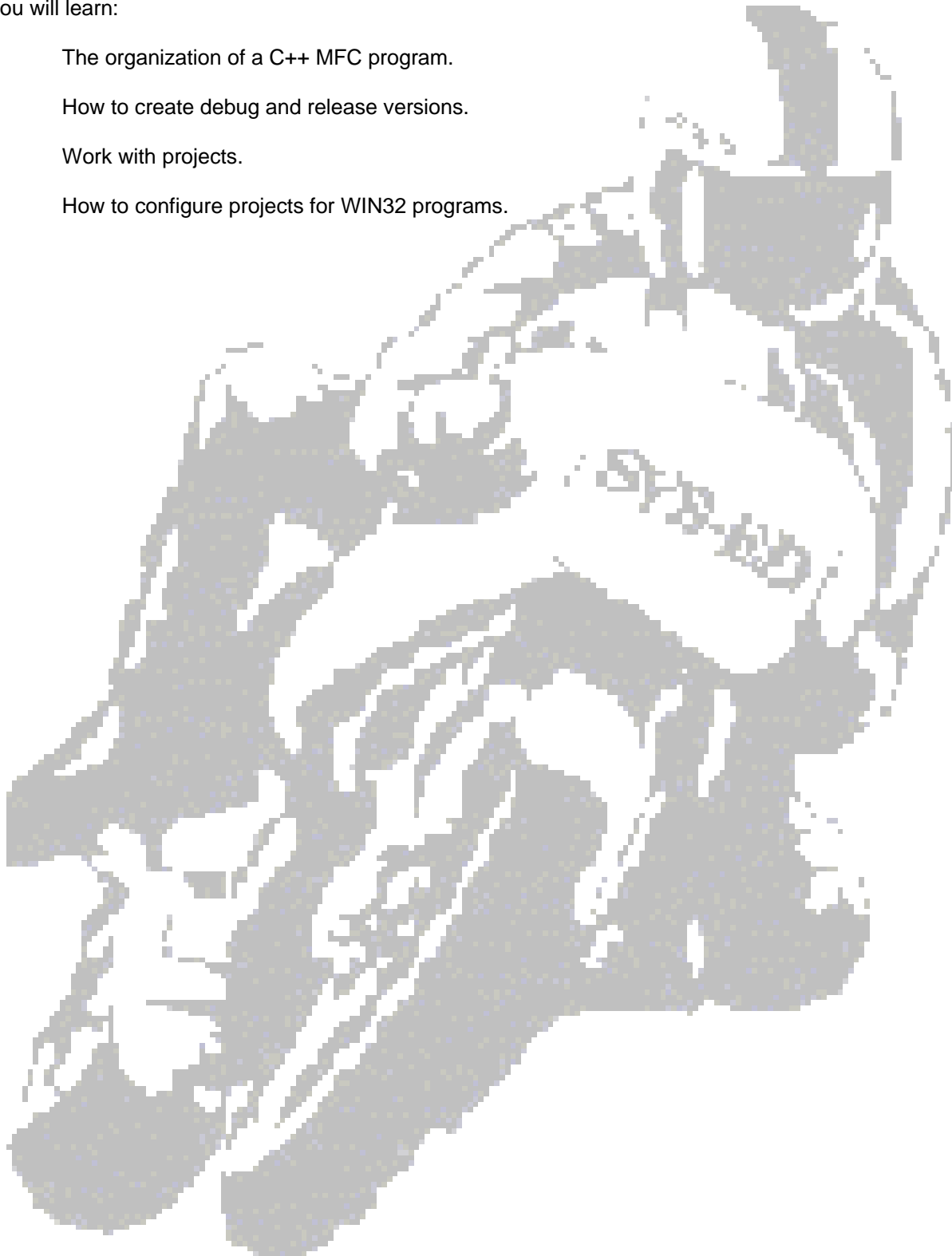


**SYS-ED/
COMPUTER
EDUCATION
TECHNIQUES, INC.**

Objectives

You will learn:

- C The organization of a C++ MFC program.
- C How to create debug and release versions.
- C Work with projects.
- C How to configure projects for WIN32 programs.



1 Project Workspace

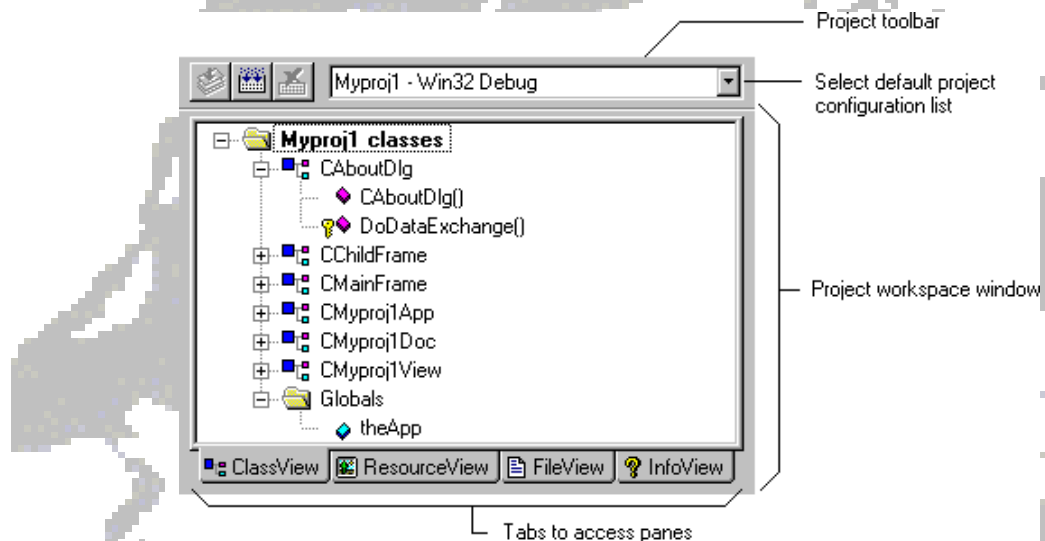
The project workspace organizes projects and their elements, and maintains preferences for the display of information.

- C The project workspace consists of a subdirectory and various files.
- C The files describe the individual projects in the project workspace, and how to display them.

There are three basic scenarios for using project workspaces:

- C A top-level project only.
- C A top-level project with a single subproject.
- C An empty top-level project with multiple subprojects, which also may have subprojects.

When you create or open a project workspace, Microsoft Developer Studio displays the elements of your Project Workspace in the project workspace window.



When you open a project workspace file, Developer Studio displays the Project Workspace window, along with other windows, in the last locations and prompts their selection.

- C The Project Workspace window can be docked or undocked, sized, moved, or hidden.
- C In the Project Workspace window, Developer Studio creates panes, which can be accessed from the tabs at the bottom of the window.
 - Information can be accessed about elements of the project workspace from the views in the Project Workspace window.
 - Selecting any item and pressing ALT+ENTER opens the property page for that item.
 - Double-clicking any item in a pane displays that item in an appropriate way: source files in a text editor, dialog boxes in the dialog editor, information topics in the topic window, and so on.

1.1 New Project

When you start a software development task with Microsoft Developer Studio, you create a project workspace and an initial project in the workspace. The initial project has Debug and Release configurations for each platform that you choose. Before you create your project workspace, you should determine which of the basic scenarios for project workspace organization suits your needs best.

With Visual C++, there are essentially two ways to create a new project workspace and the initial new project in the workspace:

- C Choose an AppWizard, OLE ControlWizard, or Custom AppWizard project type.

These choices automatically create starter files with the appropriate classes using the Microsoft Foundation Class Library (MFC).
- C Choose another project type.

In this case, all the files must be created, and the files must be selected and added to the project.

When a new project workspace is created for Visual C++, Microsoft Developer Studio always creates the following two files:

Makefile	This file has the extension .MAK. It contains all commands, macro definitions, options, and so forth to specify how to build all the configurations for all projects in the project workspace.
Workspace configuration file	This file has the extension .MDP. It contains environment settings for Developer Studio, such as window sizes and positions, insertion point locations, state of project breakpoints, contents of the Watch window, and so on.

These files can not be modified directly.

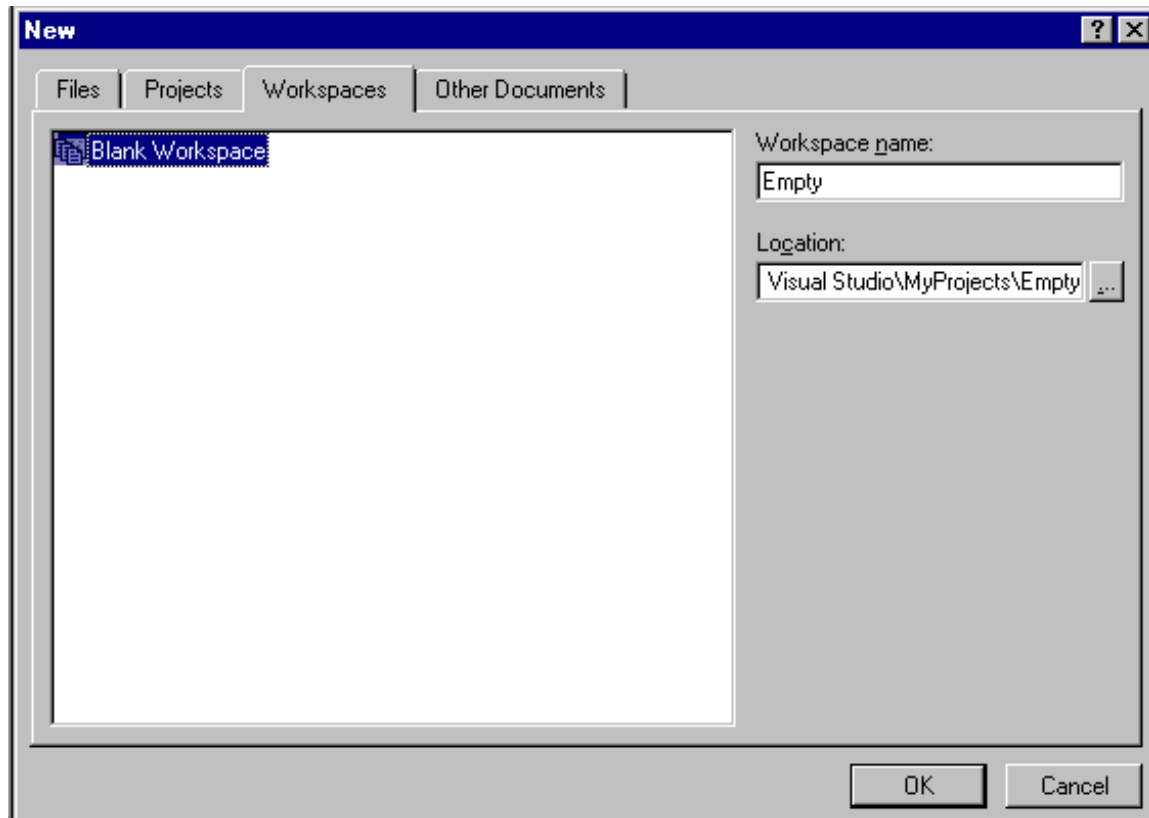
When creating a project workspace, select a root directory in which to create the project workspace directory. By default, Developer Studio selects the PROJECTS directory under your installation directory. You can, however, choose another directory. If you choose another directory, Developer Studio uses that choice for all subsequent project workspaces that you create.

To create a project workspace:

1. From the File menu, choose New.

The multi-tabbed New dialog box appears.
2. Select the Workspaces tab.
3. In the Name text box, type the name for the project workspace. This name is also used for the initial project in the Project Workspace window.

Developer Studio automatically creates a new subdirectory with this name for your project workspace and for the files for the initial project.



4. If you want, type a different location for the root directory for this project workspace in the Location text box, or choose the Browse button and select a location.
5. Click the OK button.

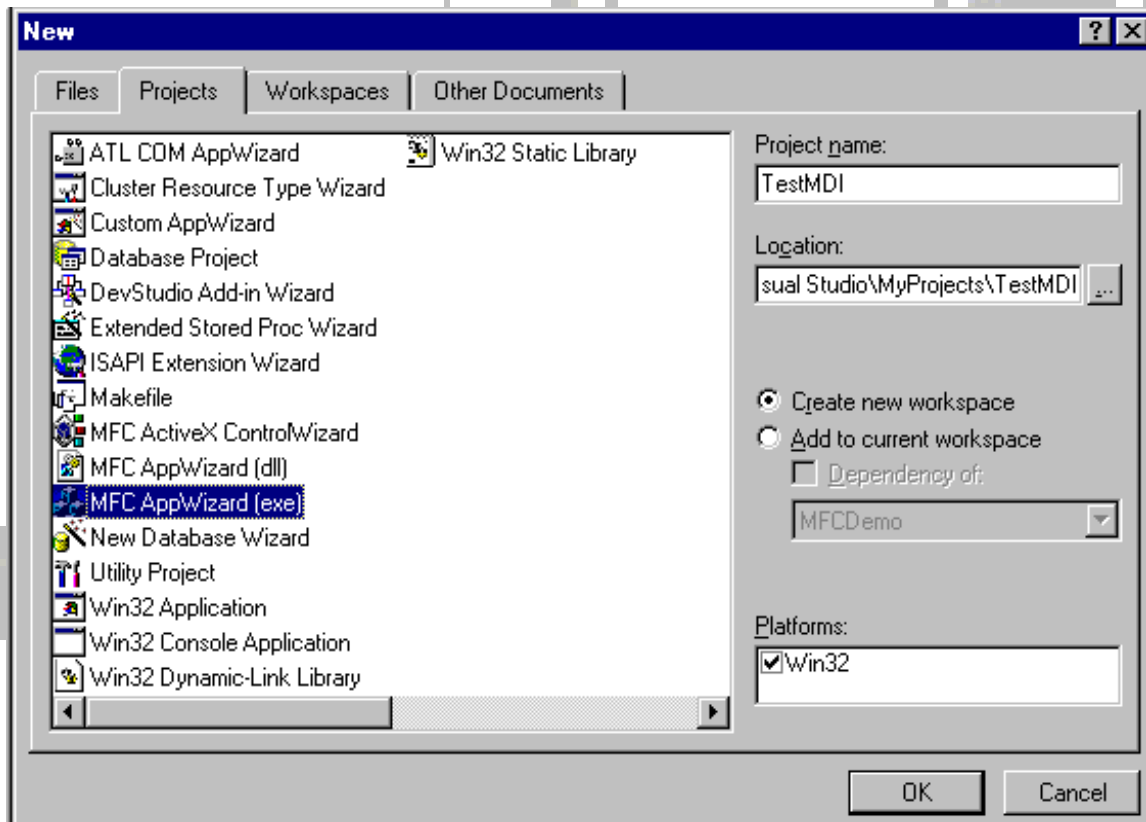
1.2 Top Level Project

This organization is suitable for the development of a single application without any dependencies on any other applications.

Choose this organization if it is necessary to develop, for instance, a single application generated by AppWizard, a single console application, or a static library.

To create a top-level project:

1. From the File menu, choose New.
2. From the New dialog, choose Projects tab.



3. Select the project type from the left side pane.
4. In the Name text box, type a name for the project workspace.
This name is also the name of the initial top-level project.
5. In Platforms list, Win32 is checked by default.
6. In the Location text box, type another directory name in which you want to create this project workspace subdirectory if you do not want to use the default directory.
7. Click the OK button.

1.3 Project Type

Each project has a project type, which is chosen when the project is created.

The project type specifies what to generate and specifies some default settings required in order to build that output type.

Project Type	Description
ATL COM AppWizard	With ATL, you can create efficient, flexible, lightweight controls.
Cluster Resource Type Wizard	Creates two projects that you can deploy on a Microsoft Cluster Server to manage and monitor applications on the cluster.
Custom AppWizard	Using the standard MFC AppWizard as a reference, you can create Custom AppWizards that will create programs appropriate for your work environment.
Database Project (Enterprise Edition only)	You can use direct database debugging to test SQL stored procedures without creating an ODBC application to call the stored procedure. Direct database debugging bypasses the need for an ODBC application.
DevStudio Add-in Wizard	With add-ins or VBScript macros, you can automate routine tasks in the Visual C++ Developer Studio environment.
Extended Stored Procedure (Enterprise Edition Only)	Using the Extended Stored Procedure Wizard to create an extended stored procedure is supported only in Visual C++ Enterprise Editions.
ISAPI Extension (Internet Server API) Wizard	Creates an ISAPI Extension or Filter .DLL using the ISAPI Extension Wizard.
Makefile	Any type of command-line program or any makefile created by an application other than the current version of Developer Studio.

Project Type	Description
MFC ActiveX ControlWizard	When you first create an ActiveX (formerly OLE) Control Program, you use the MFC ActiveX ControlWizard.
MFC AppWizard (dll version)	Function libraries developed with MFC.
MFC AppWizard (exe version)	Applications with a full graphical interface, developed with MFC.
Utility Project	The utility project does not generate any predetermined output files, such as a .LIB, .DLL or .EXE. A utility project can be used as a container for files you can build without a link step
Win32 Application	Applications with a full graphical interface, developed with Windows NT Win32 API functions or with MFC.
Win32 Console Application	Applications developed with Console API functions, which provide character-mode support in console windows.
Win32 Dynamic-Link Library	Function libraries developed with Windows NT Win32 API functions that are called dynamically at run time by 32-bit Windows-based programs.
Win32 Static Library	Standard libraries created directly by the build, using the object files and other library files belonging to the project.

2 Saving and Opening Files in Project Workspace

To save all files that have been modified in Microsoft Developer Studio, whether or not they are included in a project, choose Save All from the File menu.

To open an existing project workspace:

1. From the File menu, choose Open Workspace.

The Open Workspace dialog box appears.

The default selection in the List Files Of Type drop-down list is Workspaces (.dsp, mdp).

2. Select the drive and directory containing the project workspace that you want to open.
3. Select the .dsp or msp file for the project workspace from the File Name list and choose OK.

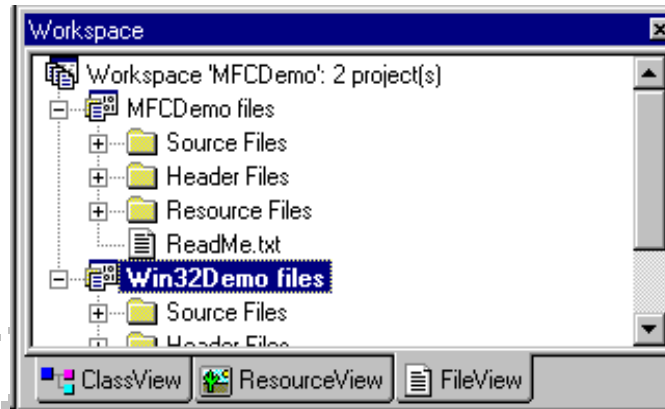
OR

Double-click the filename in the list.

3 Adding Files to a Project

To add files to a project:

1. From the Workspaces window, select the project to which you want to add files.



2. From the Project menu, choose Add to Project...Files.
The Insert Files Into Project dialog box appears.
3. Select the file type to display.
4. If necessary, select the drive and directory to view.
5. Select one or more files from the File Name list. You can use the SHIFT or CTRL key in conjunction with the mouse to make multiple selections.
6. Choose OK.

To add an open source file to a project:

1. With the mouse pointer in the source file, click the right mouse button.
2. From the pop-up menu, choose Add To Project, and select the project name from the cascading menu.

To remove files from a project:

- C. Select the file in FileView, and from the Edit menu, choose Delete.
- OR
- C. Press the DEL key.

You can hold down the CTRL or SHIFT keys and use the mouse to select multiple files in the Project Workspace window.

To move or copy files from one project workspace to another:

1. In the FileView pane of the Project Workspace window, select the files that you want to move or copy.

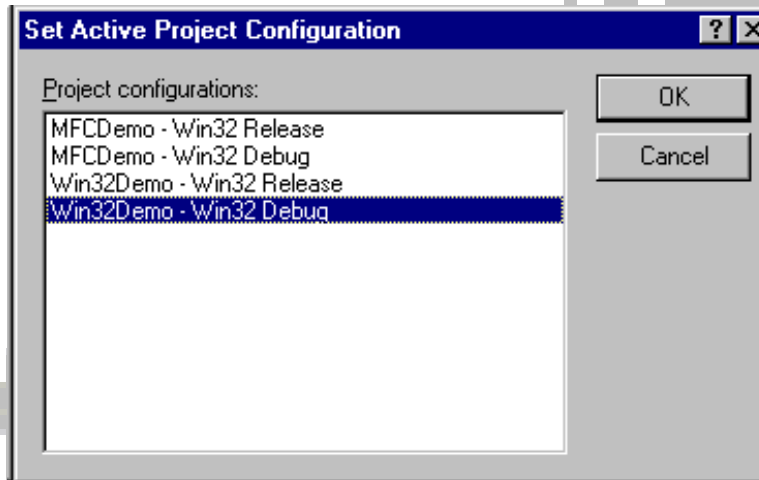
You can hold down the CTRL or SHIFT keys to select multiple files in the Project Workspace window.
2. From the Edit menu, choose Cut if you want to move the files, or Copy if you want to copy the files.
3. Close the current project workspace.
4. Open the destination project workspace.
5. Select the project to receive the files.
6. From the Edit menu, choose Paste.

4 Project Configuration

To select a project configuration:

1. From the Build menu, choose Set Active Configuration.

The Active Project Configuration dialog box appears.



2. From the Project Configurations list, select the default project configuration.
3. Choose OK.

To build the default project configuration:

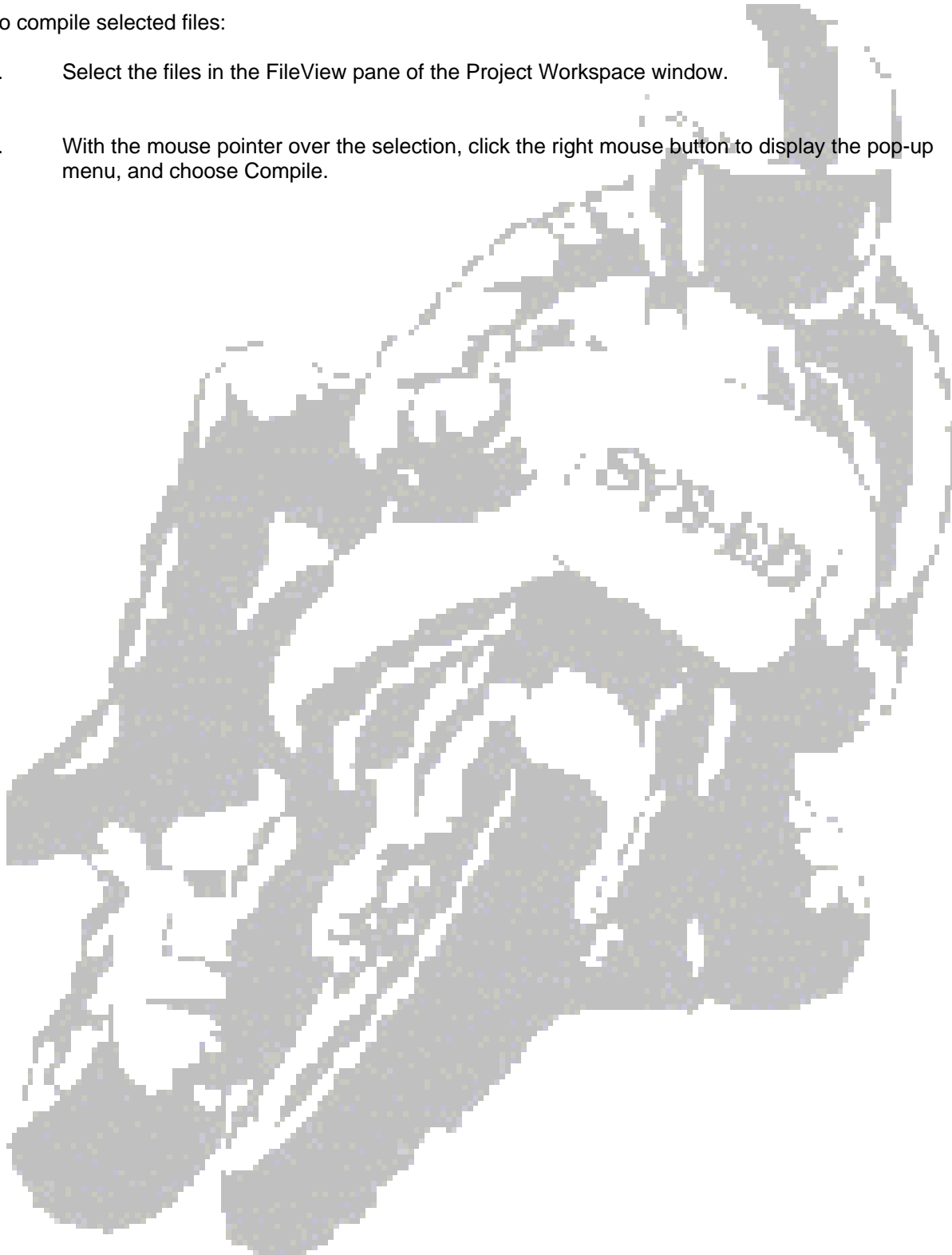
- C From the Build menu, choose Build *project*, where *project* represents the program or library defined by the project configuration.

If you want to ensure that all files associated with a project configuration get built, whether or not they are out of date, you can choose the Rebuild All command.

5 Compiling Files

To compile selected files:

1. Select the files in the FileView pane of the Project Workspace window.
2. With the mouse pointer over the selection, click the right mouse button to display the pop-up menu, and choose Compile.



6 Executing Programs

To run an executable program:

- C From the Build menu, choose Execute *project*, where *project* represents the program defined by the project configuration.

To run an application in the integrated debugger:

- C From the Build menu, choose Debug, and from the cascading menu, choose Go, Step Into, or if you have a source file open and it has the focus, Run To Cursor.

7 Recording Keystrokes

The text editor, can be used to automate repetitive keyboard tasks by recording and playing back keystrokes.

The playback feature is available until you record a new set of keystrokes or end the editing session.

To record keystrokes:

1. Move the mouse pointer to where you want to begin typing.

2. From the Tools menu, choose Record Quick Macro.

The Record toolbar appears.

3. Record the keystrokes that you want.

During recording, all mouse-driven selections are disabled. Keystrokes are entered at the location you have selected.

4. From the Tools menu, choose Stop Recording when you have finished recording your keystrokes.

To play back keystrokes:

1. Move the mouse pointer to where you want to play back the recorded keystrokes.

2. From the Tools menu, choose Play Quick Macro.

The recorded keystrokes will be played back into the active editor window at the location you have selected.