

MICROSOFT VISUAL C++ 6.0 TUTORIAL

INTRODUCTION

Microsoft Visual C++ allows you to create many different types of applications. This guide addresses creating and using Console Applications. A console application is a program that runs inside a DOS window. This guide is divided into the following sections:

- Starting Visual C++
 - Creating and Executing a Single-File Console Application
 - Importing and Executing C++ Source Code Files
-

STARTING VISUAL C++

To start the Microsoft Visual C++ compiler, click the **Start** button on the task bar to open the start menu. Open the **Programs** menu (**All Programs** in Windows XP) and select **Microsoft Visual C++ 6.0**. If a Tip of the Day dialog box appears, click the **close** button to close the dialog box. If the Microsoft Visual C++ window is not maximized, click the **maximize** button in the upper right corner of the title bar.

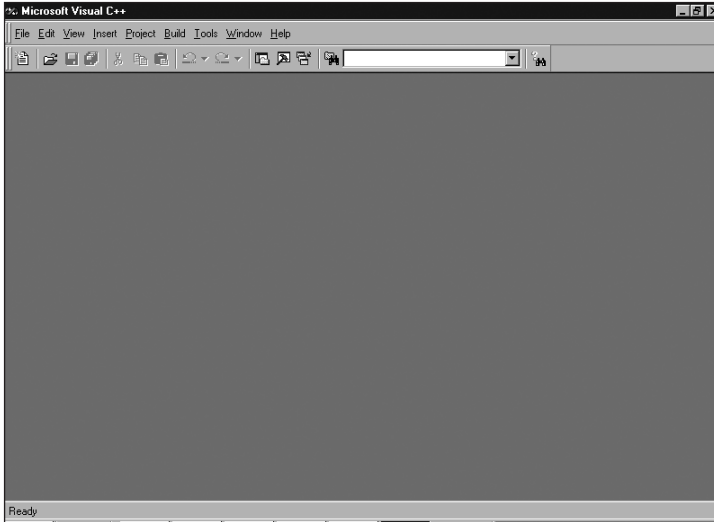


Figure 1-1 Microsoft Visual C++ 6.0

Figure 1-1 shows the initial application window. If your application window looks different, you may need to adjust the menu bar settings. This is accomplished by right-clicking the menu bar and modifying the settings until the Standard toolbar is the only item selected as shown in Figure 1-2.

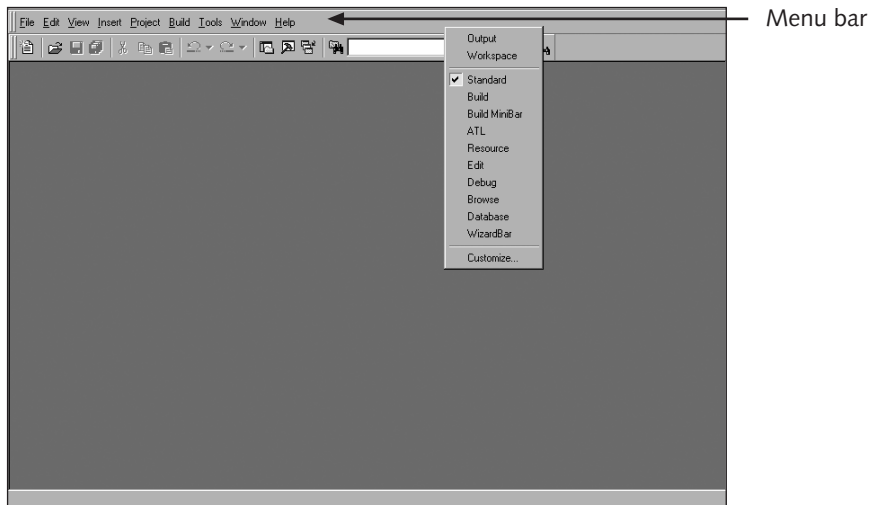


Figure 1-2 Initial Menu Settings

CREATING AND EXECUTING A SINGLE-FILE CONSOLE APPLICATION

One of the most common programs encountered in an introductory C++ programming class is a single source code file that contains the main function. To construct this type of application, you need to:

- Create a Win32 Console Application Project
- Add a source code file
- Write the program
- Execute the program
- Debug the program

Create a Win32 Console Application Project

Start Microsoft Visual C++. Select **New** from the File menu. (See Figure 1-3.)

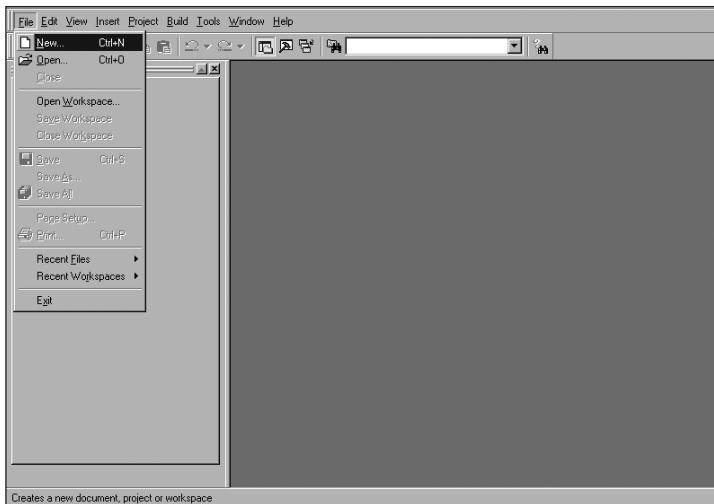


Figure 1-3 Create a New Project

When the New Project dialog box opens, enter the following information (see Figure 1-4):

- Select Win32 Console Application
- Enter the project name in the Project Name textbox
- Select the location for the project in the Location textbox

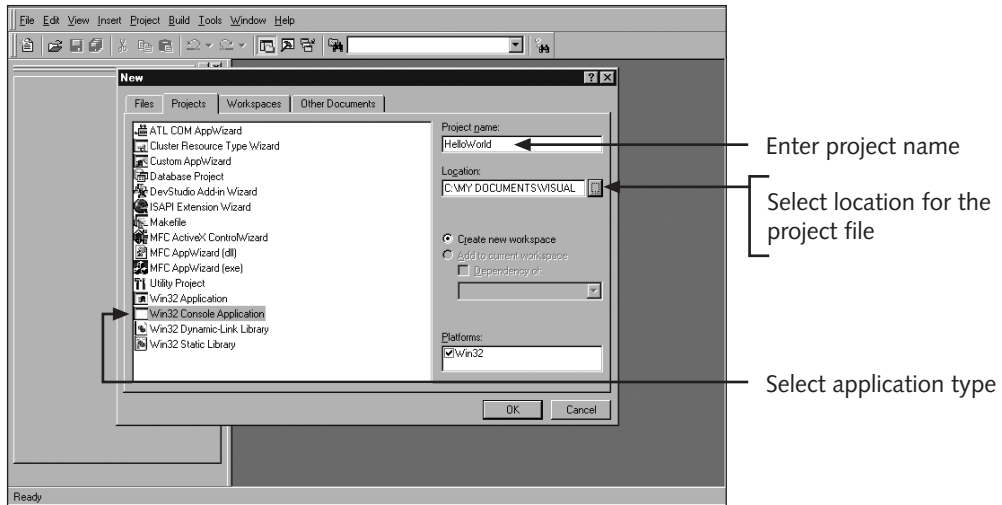


Figure 1-4 New Project Settings

The application provides a default location for saving projects or you can select your own location by pressing the button to the right of the location textbox to open the Choose Directory dialog box.

Press the **OK** button after entering the required information into the dialog box. When the Win32 Console Application wizard appears, select **An empty project.** (see Figure 1-5) and then press the **Finish** button. When the New Project Information dialog box appears, select **OK.**

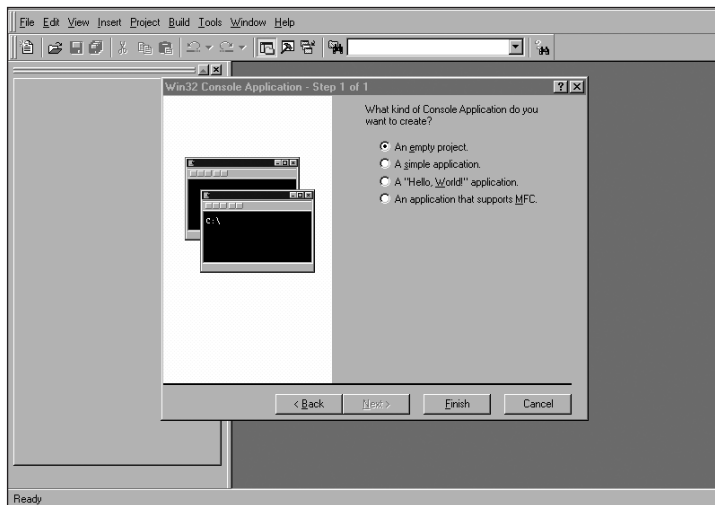


Figure 1-5 Win32 Console Application Wizard

Add a Source File

Add a blank source code file to your project by selecting **New** from the File menu. (See Figure 1-6.)

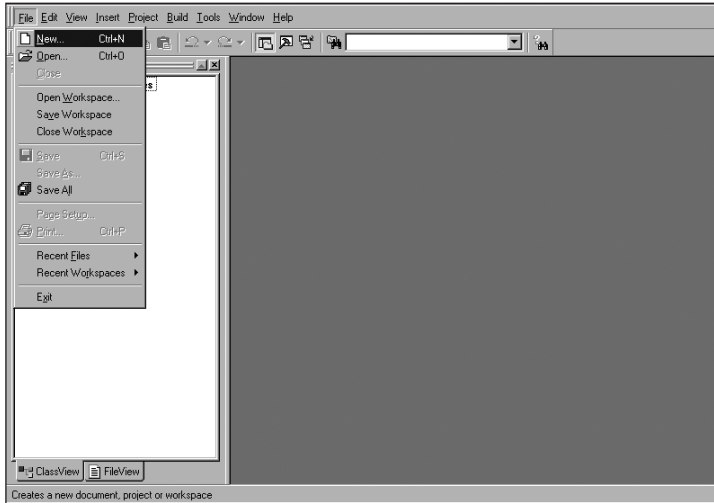


Figure 1-6 Adding Project Files

When the New file dialog box appears, select **C++ Source File**, type the file name into the File Name textbox (see Figure 1-7), and press the **OK** button.

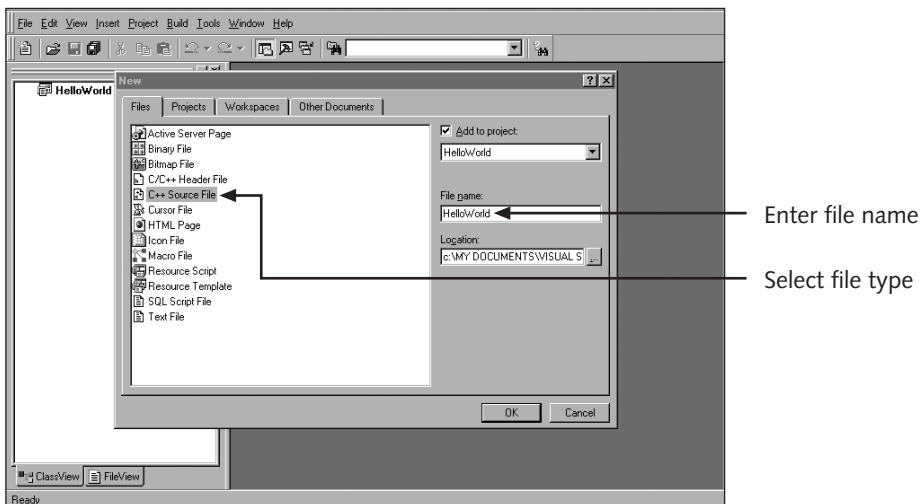


Figure 1-7 New File Settings

Write the Program

Type the source code for your program into the section located in the right side of the window. Figure 1-8 contains the source code for a simple C++ console program that displays the phrase Hello World. Note that the C++ editor automatically selects different colors for C++ reserved words and provides automatic indentation for blocks of code.

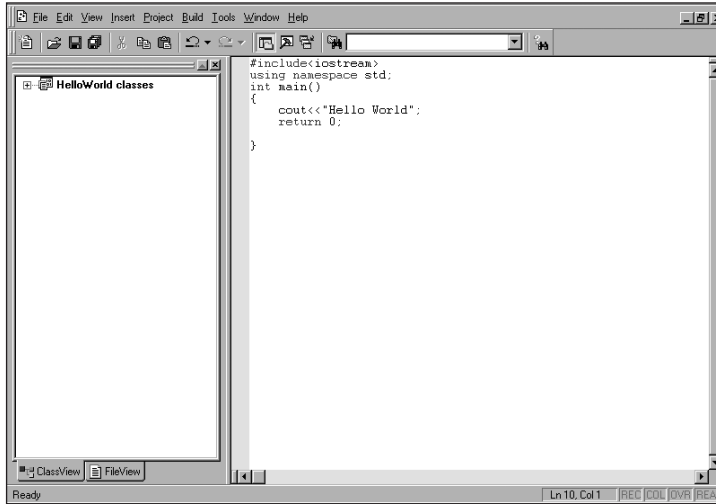


Figure 1-8 Hello World Source Code File

Executing the Program

Before you can execute the program you have written, you must save it and compile it. To save the file, select **Save** from the File menu. Once the program is saved, you compile it by selecting the **Compile** option from the Build menu. (See Figure 1-9.)

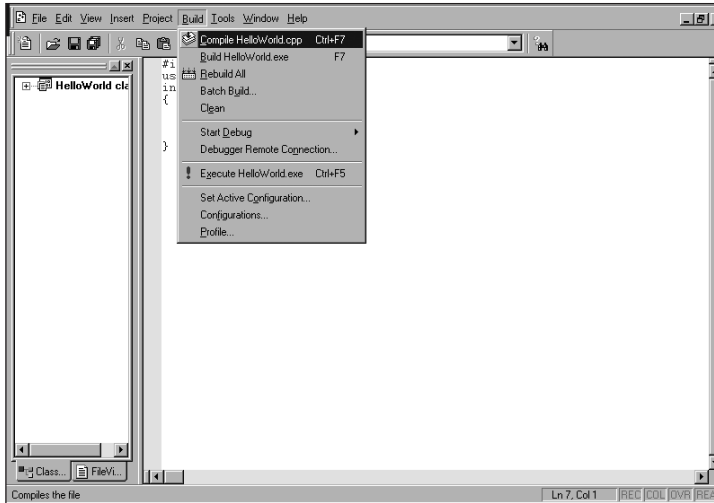


Figure 1-9 Compiling a Program

Once the program compiles, the results will be shown in the Output Window. (See Figure 1-10.) In this example, the program compiled with no errors and no warnings.

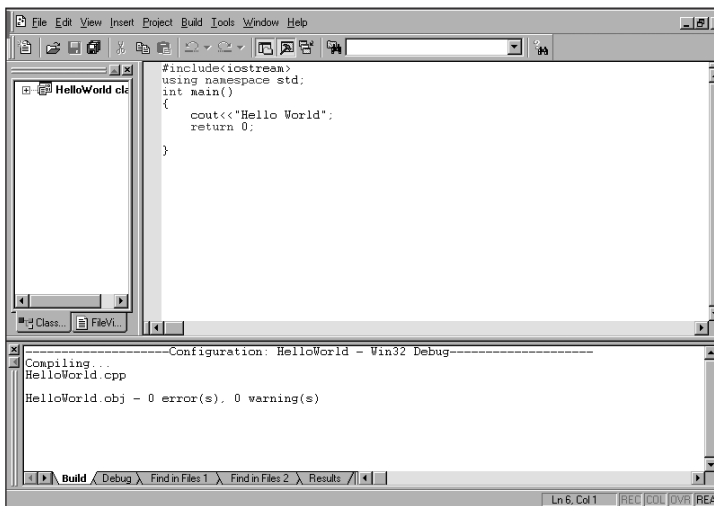


Figure 1-10 Results of Compiling the Program

After successfully compiling the program, the next step is to take the compiled object code and build the executable file necessary to run the program. In Microsoft Visual C++ 6.0, this is accomplished by selecting the **Build** option from the Build menu. (See Figure 1-11.)

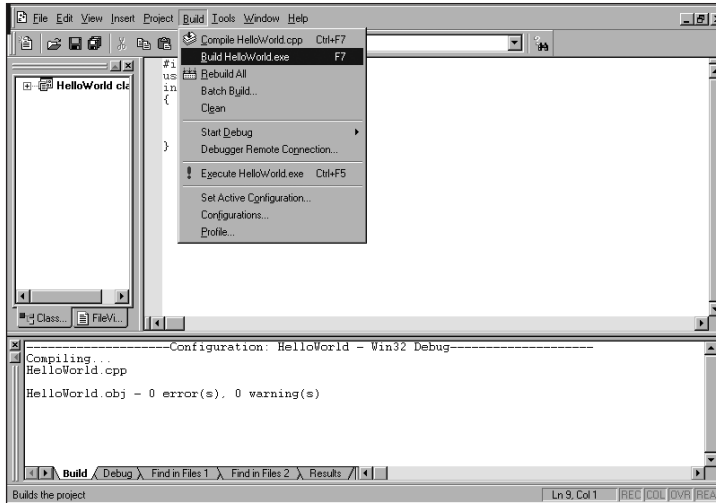


Figure 1-11 Initiating the Build Process

In this example, there were no errors or warnings generated by the build process. (See Figure 1-12.)

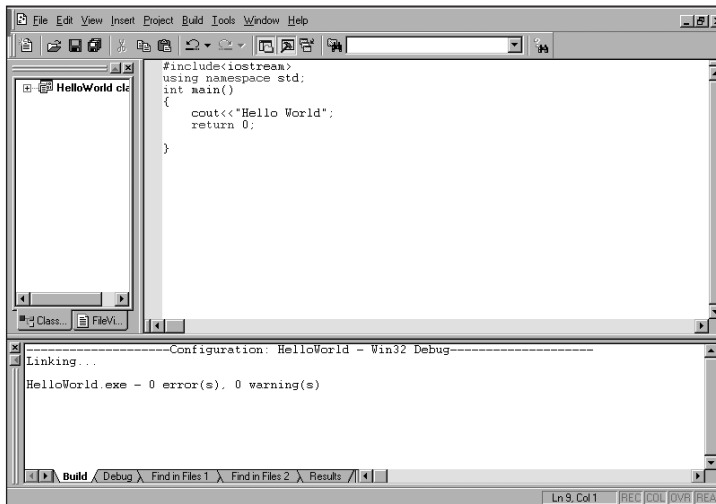


Figure 1-12 Results of the Build Process

After the build process has been successfully completed, you can now execute the program by selecting **Execute** from the Build menu. (See Figure 1-13.)

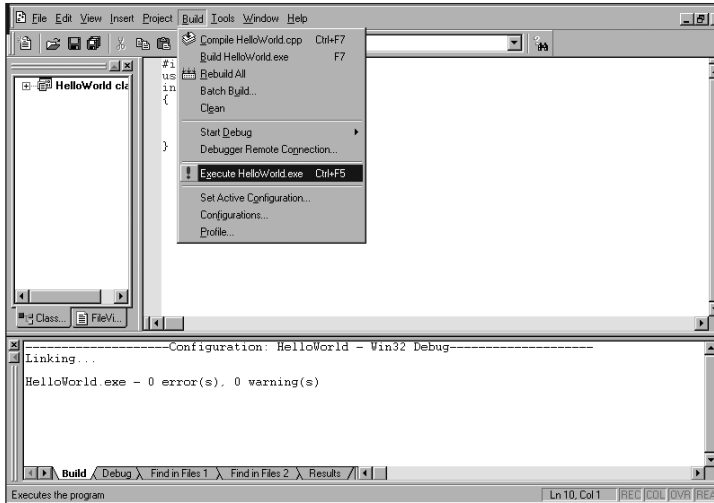


Figure 1-13 Initiating Program Execution

The program results will appear in a new DOS window. Notice that the phrase “Press any key to continue” has been added to the program output. This additional code was added to keep the DOS window open until you have had a chance to view the output and press a key on the keyboard. (See Figure 1-14.) Once a key on the keyboard is pressed, the program stops execution and the DOS window closes.

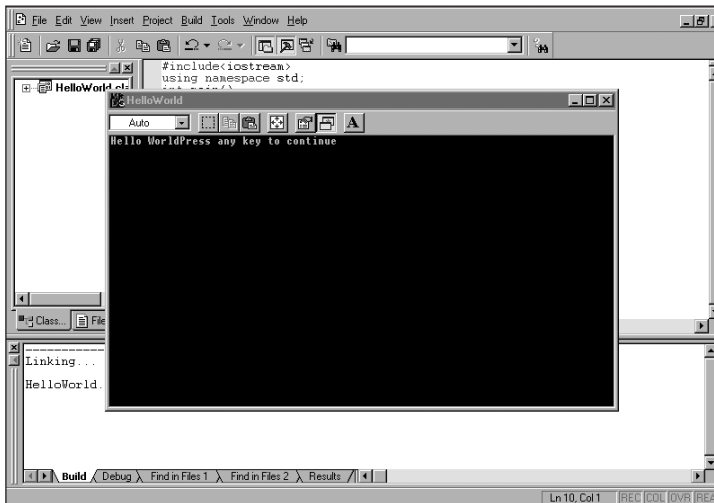


Figure 1-14 Program Results

Debug the Program

Often, the program will have errors when you attempt to compile, build, and execute it. The program that was successfully executed in the previous example has been modified to include an error—the semi-colon at the end of the return 0 statement has been removed. When the modified program is compiled, an error is displayed in the output window. (See Figure 1-15.)

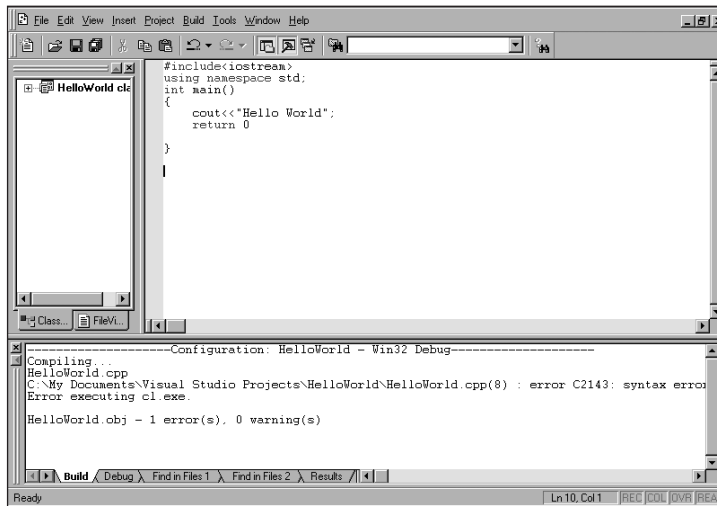


Figure 1-15 Compile Error

You can determine where the compiler found the error by double-clicking on the error message in the output window. This will cause a pointer to appear in the left margin of the source file where the error was encountered. (See Figure 1-16.)

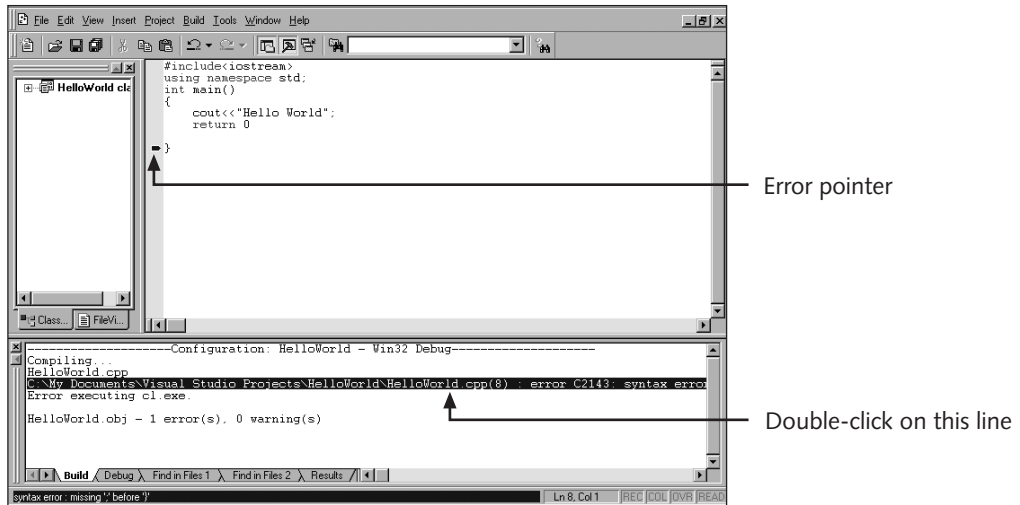


Figure 1-16 Locating a Compile Error

Notice in this case that the pointer is on the line after the line containing the actual error. This occurs when the error induces a compiler recognized fault on a subsequent line. While not always exact, the error pointer, in conjunction with the error description, can help locate errors in the source code. Once the error is identified and corrected, the program must be saved, re-compiled, built, and executed again.

IMPORTING AND EXECUTING C++ SOURCE CODE FILES

Often you will want to compile, build, and execute programs that you did not write. Consider for example the case where you want to execute a source file provided as part of a textbook's supplemental information. The steps are nearly identical to those described above. First you need to construct a Project file as described above. Next, instead of adding a new source file using the New command as describe above, you import the existing file into the project. This is accomplished by selecting the Add To Project option located under the Project menu. (See Figure 1-17.)

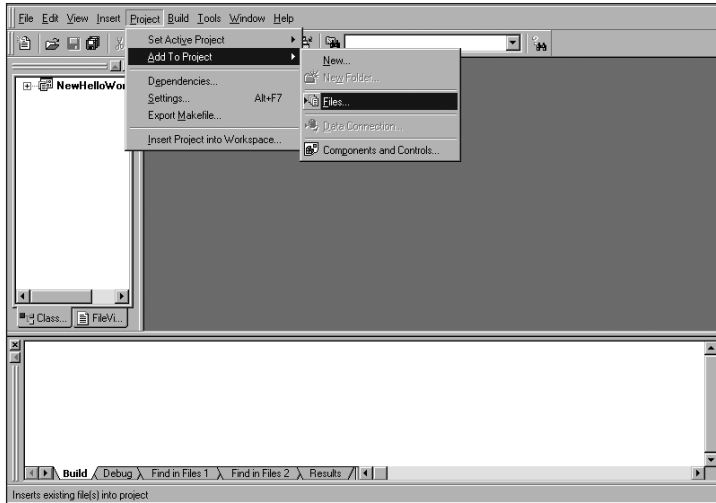


Figure 1-17 Importing a File into a Project

Once the desired file has been imported into the project, you compile, build, and execute the program as described above.

SUMMARY

Microsoft Visual C++ allows you to create many different types of applications. This guide addressed creating and using single source file Console Applications, but the basic operations are the same for more complex programs.

- Always start by creating a project file of the appropriate type
- Create blank files for writing your own programs or import existing files into the project
- Compile
- Build
- Execute
- Debug, if necessary