Contents

For help, click a topic below---or press TAB to move to the topic, and then press ENTER:

Welcome to Setup

If you select Continue, the Setup program:

- Prompts you for information
- Confirms your Setup choices
- Copies files to your hard disk
- Builds requested libraries
- Asks you to confirm changes to system files

The *Getting Started* manual contains information on:

- System requirements
- Product components
- Location of topics in the documentation
- CONFIG.SYS and AUTOEXEC.BAT files
- SYSTEM.INI and TOOLS.INI files
- Memory configuration for optimal MS C/C++ performance

After MS C/C++ has been installed, you can run Setup again at any time to:

- Copy additional file groups to your hard disk
- Build additional run-time libraries

For more information on any screen, either:

- Press F1, or
- Choose the Help button.

If you have used an earlier version of Microsoft C, you will find many new features in Microsoft C/C++ version 7.0:

- Support for C++ programming
- Precompiled header files for faster compilation
- Support for packed code (p-code) to reduce program size
- Support for inlining functions so the compiler can automatically substitute function code for function calls when the substitution will optimize your code
- Function allocation using based addressing
- Adjustable warning levels and message display control
- QuickWin library for converting MS-DOS applications to applications for Windows (TM)
- Remote debugging with CodeView
- VESA Super VGA Graphics Support

See Appendix C in the *C Language Reference*, "Differences between C Versions 6.0 and 7.0," for more information about these new features.

If you have a previous MS C/C++ beta version of the compiler on your computer, you must delete it before continuing with Setup.

If you do not want to install MS C/C++ now:

- Select Exit on the previous screen

Environment

Setup makes several changes to your SYSTEM.INI file, but one DEVICE statement in SYSTEM.INI must be different depending on the version of Windows you use. If you configure MS C/C++ for Windows 3.0, you will not be able to run the compiler under Windows 3.1 without changing your SYSTEM.INI file. Your SYSTEM.INI must contain this statement for running under Windows 3.0:

```
DEVICE=VMCPD.386
```

When you use the C/C++ compiler under Windows 3.1, you must remove this statement from your SYSTEM.INI file and replace it with

```
DEVICE=*VMCPD
```

If you have both Windows 3.0 and Windows 3.1 on your system or upgrade from Windows 3.0 to Windows 3.1, you must edit your SYSTEM.INI file to contain the appropriate DEVICE statement.

Base Directory

The base directory should contain both drive and path. This will be the root directory for all files installed for MS C/C++.

Before file copy begins, Setup verifies that your hard disk has adequate space.

Default or Custom Installation

If you select Default Installation, Setup displays the README.TXT file and begins copying files and building libraries.

The following file groups and libraries are installed if you select Default Installation:

- C/C++ Compiler
- Run-Time Libraries for MS-DOS and Windows Operating Systems
  - Small and medium memory model
  - Emulation for floating-point math
  - Windows .EXE files
  - MS-DOS graphics library is not installed
- Microsoft Foundation Classes
- Programmer's WorkBench
- CodeView Debugger
- Sample Source Code
- Online Help

If you select Custom Installation, you will be able to select the file groups, tools, and libraries that you want Setup to install. Then Setup displays the README.TXT file while copying files.

If you have limited disk space, you can choose not to install particular file groups by selecting Custom Installation.

NOTE: The Build Additional Libraries option adds additional libraries. It does not install the compiler. Select this only if you have already installed the compiler, but want to add new libraries such as those for different memory models, environments (such as QuickWin), floating-point options, or the Microsoft Foundation Classes.

To install file groups and libraries at a later time, use the same base directory, select Custom Installation, and select the file groups you wish to add.

Custom C/C++ 7.0 Install

C/C++ 7.0 Compiler

New to Microsoft's C compiler is support for C++. See Chapter 1 of the *Getting Started* manual for a description of product components and new features.

Run-Time Libraries

Use the Libraries button on the previous screen to select the memory model, floating-point math options, and types of executables you will use when developing applications. Setup builds run-time libraries to support the options you select. By default Setup builds the small and medium memory model libraries.

The byte sizes of the components shown on the previous screen are rough estimates, not exact byte counts.

If you select Run-Time Libraries, the C startup source code is copied automatically to your \C700\SOURCE\STARTUP directory. The C startup source code is the code in the run-time libraries that is linked into your executables to give startup instructions for program execution. Since this code is distributed with C/C++, you can modify this code if necessary.

Microsoft Foundation Classes

The Microsoft Foundation Class library is a C++ class library designed for writing Microsoft Windows applications in C++. The Microsoft Foundation Class library offers more than 60 reusable C++ classes that include all the Windows Graphical User Interface (GUI) features as well as general-purpose classes that support the nongraphical portion of an application. Sample code demonstrating the Microsoft Foundation Classes will also be installed if you select this option. The *Class Library Reference* manual and the *Class Library User's Guide* manual contain complete information for using this library.

Programmer's WorkBench (PWB)

PWB, a character-mode integrated development environment, includes an editor, a source code browser, and a complete help system in a menu-driven interface. For complete reference information and a tutorial on using PWB, see the *Environment and Tools* manual.

When you have Setup install PWB, Setup also copies a file named TOOLS.C70 to the \C700\INIT directory. This file contains environment settings and macros for use with PWB. Append the contents of this file to your existing TOOLS.INI or rename TOOLS.C70 to TOOLS.INI.

CodeView Debugger

The CodeView debugger is installed if you select this option. CodeView is now composed of multiple DLLs. This allows you to configure CodeView for the type of applications you develop. You can also access CodeView from the PWB environment. CodeView also requires settings in the TOOLS.C70 file be added to your TOOLS.INI file if you want to debug p-code or do remote debugging. For complete reference information, see the *Environment and Tools* manual or the Microsoft Advisor Help system.

Sample Source Code

Source code of sample C and C++ programs is copied to your \C700\SAMPLES directory. If you request MS-DOS targeting, Sample Source Code, and the MS-DOS Graphics Libraries, Setup copies the samples that demonstrate graphics library functions. Some of these sample programs are designed to accompany tutorials in the following MS C/C++ books: *Class Libraries User's Guide*, *C++ Tutorial*, and *Environment and Tools*.

Online Help

Help for the Microsoft C and C++ language and libraries, debugger, PWB, Microsoft Foundation

Classes, and all tools and utilities will be copied to \C700\HELP.

MS-DOS Graphics Libraries

The MS-DOS graphics libraries provide a set of MS-DOS graphics and full-screen text functions. These functions include coordinate conversion, figure drawing and filling, text and font manipulation, image manipulation, cursor positioning, screen clearing, color-text display, and presentation graphics. Setup copies the GRAPHICS.LIB and PGCHART.LIB libraries if you choose this file group. If you request MS-DOS targeting, Sample Source Code, and the MS-DOS Graphics Libraries, Setup copies the samples that demonstrate graphics library functions.

NOTE: The functions in the graphics library are for MS-DOS only. They are not for use with applications for Windows.

Custom Libraries

Memory Models

Memory model refers to the size requirements for the code and data segments in a program. This determines the default pointer sizes a program uses when accessing executable code and data. Pointers can be near (16-bit) or far (32-bit). Far pointers must access data or code by specifying both a segment and an offset.

Model	CODE	DATA
Small	Near	Near
Medium	Far	Near
Compact	Near	Far
Large	Far	Far

If you are concerned about disk space, select only the memory models you need. You can run Setup at any time to create new libraries.

Math Support

The floating-point math support that your programs need depends on the existence of a math coprocessor chip on the host or target computer.

If your computer has a math coprocessor, both the emulator and 80x87 chip libraries are the default.

The emulator uses a math coprocessor if one is present; otherwise it emulates the coprocessor with software. The results may differ slightly from the results obtained if a coprocessor is used.

The 80x87 chip library uses a math coprocessor if one is present, but if a coprocessor chip is not available on the target computer at run time, an error results.

The alternate math library never uses the coprocessor. The results of floating-math operations are the same on any computer, but calculations are slower on systems that do have a coprocessor. In addition, calculations do not have the same degree of accuracy as emulator and coprocessor libraries.

NOTE: Floating-point support in Windows does not require separate libraries. The Windows libraries use built-in emulation software.

See Chapter 4, "Managing Memory in C," Chapter 5, "Managing Memory in C++," and Chapter 7, "Controlling Floating-Point Math Options," of the *Programming Techniques* manual for more information.

Windows Targets

The library support you need depends on the type of executables and libraries you intend to build with MS C/C++.

Select Windows .EXE Files for developing stand-alone executable programs for Windows.

Select Windows .DLL Files if you will be developing dynamic-link libraries for applications for Windows.

Select QuickWin .EXE Files if you want to develop MS-DOS applications that use standard input and output to run in Windows using a simple Windows interface. This is the quickest way to port MS-DOS programs to Windows, but there are limitations on what the converted programs can do. For more information, see Chapter 8 of the *Environment and Tools* manual.

Setup automatically builds the appropriate library for each memory model you select.

If you selected MS-DOS or MS-DOS and Windows as your target environment, the MS-DOS libraries are automatically built.

Change Installation Directory

If you have found that you do not have enough hard disk space on the previously selected base directory, you may now change to another drive and path.

Target Environment

If you select MS-DOS only, you will not be able to install any Windows-specific tools or libraries.

If you want to produce executables for both MS-DOS and Windows, select targeting for Both MS-DOS and Windows.

Setup recommends that you select both MS-DOS and Windows unless you have limited hard disk space.

Begin Copying Files

If you want to view the README.TXT file later, it is copied to your hard disk as C:\C700\README.TXT.

The README.TXT file contains important information that you should read before running Microsoft C/C++.

System Files

You have three options for how Setup modifies your AUTOEXEC.BAT, CONFIG.SYS, and SYSTEM.INI files:

- Let Setup modify your system files for you.

  Setup saves current copies in the same directory, with .OLD as the file extension, and automatically updates all three files.
- View and edit the changes now.

  Setup saves current copies in the same directory, with .OLD as the file extension, and automatically updates all three files. Then Setup displays a split-screen viewer that allows you to edit the updated versions of CONFIG.SYS and AUTOEXEC.BAT. Clicking the Cancel Edits button cancels edits you have made but keeps the updates Setup made automatically.
- Save the changes to edit later.

  Setup does not modify your system files. Setup copies recommended updates named CONFIG.C70, AUTOEXEC.C70, and SYSTEM.C70 to your C:\C700\INIT subdirectory.

NOTE: Depending on your existing system configuration, some of these changes are necessary in order for C/C++ to run on your system.

For additional information on configuring your computer for MS C/C++, see Chapter 3, "Configuring Your System," in the *Getting Started* manual.

View and Edit Files

Setup has written a backup copy of your system file to a file with the .OLD extension. The viewer displays your modified system file in the top screen and your old system file in the bottom screen. You may edit in either window, but only the edits made to the file in the top window are saved.

NOTE: Pressing ENTER in the viewer is the same as pressing the Continue button. You must press CTRL+ENTER in the top screen for a carriage return.

If you press the Cancel Edit button in the viewer, Setup cancels edits you have made but keeps the updates Setup made automatically.

Edit System Files Later

Setup has not modified your system files. The changes have been saved to files with the extension .C70 in the \INIT subdirectory.

NOTE: Depending on your existing system configuration, some of these changes are necessary in order for C/C++ to run on your system.

Where to Get More Information

If C:\C700 is the base directory, the file structure Setup creates look like this:

```
C:\C700
    \BIN
    \HELP
    \INCLUDE
          \SYS
    \INIT
    \LIB
    \SAMPLES
          \CPPTUTOR
          \GRAPHICS
          \IOSTUTOR
          \PWBTUTOR
          \SORTDEMO
          \PWBEXTEN
    \SOURCE
          \MOVE
          \STARTUP
    \MFC
          \SAMPLES
          \INCLUDE
          \LIB
          \DOC
          \SRC
```

What to Do Next

If Setup has automatically updated your system files, reboot now to enable these changes in your system files.

If you have saved the system file changes to edit later, you may exit Windows now, edit the changes in MS-DOS, and then reboot your computer to enable the changes to your system.

For additional information on configuring your computer for MS C/C++, see Chapter 3, "Configuring Your System," in the *Getting Started* manual.

Once you have correctly configured your system, explore the C/C++ samples, tutorials, user guides, and reference manuals.

Need More Disk Space

The size requirements specified by Setup are the number of bytes that need to be available for Setup to successfully install the file groups you requested.

The error message you received may say you need to free disk space on a drive other than the one you specified as the base drive for C++ installation. Setup installs dynamic-link libraries in your Windows directory so Setup may need additional space on the drive where Windows is installed.

To free disk space, use the Windows File Manager to delete files, or use the DEL command in a DOS session.

There are several reasons that Setup might incorrectly determine the amount of available disk space on your hard disk or generate this "Not Enough Disk Space" error message:

- You have a utility that compresses files so the file sizes known by Setup may actually be less once the files are copied to your hard disk.
- You are installing C/C++ on a computer that uses a nonstandard disk controller.
- You actually have enough disk space, but Setup needs space on your hard disk for additional temporary files used during library building.

If you think that Setup is generating this "Not Enough Disk Space" error for one of these reasons, return to the Custom Installation screen and choose fewer options. Once you have installed those components, run Setup again to install the remaining components you need.

If you select Override and your hard disk does not have room for all requested files and libraries, Setup will continue until no more disk space is available.