

Microsoft® Macro Assembler

Installing and Using the Professional Development System

Microsoft®

Microsoft® Macro Assembler

**Installing and Using
the Professional
Development System**

Version 6.0

For MS® OS/2 and MS-DOS® Operating Systems

Microsoft Corporation

Information in this document is subject to change without notice and does not represent a commitment on the part of Microsoft Corporation. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license or nondisclosure agreement. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of Microsoft.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of Commercial Computer Software—Restricted Rights at 48 CFR 52.227-19, as applicable. Contractor/Manufacturer is Microsoft Corporation, One Microsoft Way, Redmond, WA 98052-6399.

©Copyright Microsoft Corporation, 1991. All rights reserved.
Printed in the United States of America.

Microsoft, MS, MS-DOS, CodeView, and XENIX are registered trademarks and Windows, Windows/386, and *Making it all make sense* are trademarks of Microsoft Corporation.

U.S. Patent No. 4,955,066

386-Max is a trademark of Qualitas, Inc.

BRIEF is a registered trademark of UnderWare, Inc.

IBM is a registered trademark of International Business Machines Corporation.

UNIX is a registered trademark of American Telephone and Telegraph Company.

Contents

Chapter 1	The Professional Development System	1
	Before You Proceed	2
	Features New to Version 6.0	3
	Microsoft Programmer's WorkBench	4
	Key to Document Conventions	9
Chapter 2	Installing the Microsoft Macro Assembler	11
	Running SETUP	11
	Supplying SETUP Information	12
	Responding to the Prompts	13
	Configuring Your System	17
	Customizing the Programmer's WorkBench	19
	Configuring Extended Memory for the CodeView® Debugger	20
Chapter 3	Using the Programmer's WorkBench	23
	Starting PWB	24
	Elements of the PWB Environment	24
	Using the Editor	35
	Assembling and Linking	41
	Debugging Programs	49
	Getting More Details	57
Chapter 4	Using the Online Reference System	59
	Structure of the Microsoft Advisor	59
	Navigating through the Microsoft Advisor	60
	Using QuickHelp	68
Index		75

SUCCESS!

EXIT PROGRAMMER'S WORKBENCH YES

MASIN

the GAME

NO RUN SUCCESS?

OS/2

DOS

WINDOWS

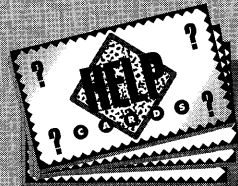
PICK AN ENVIRONMENT



DEBUG



LINK



PICK A CARD

BASIC



FORTRAN



PASCAL

C

PICK A HIGH-LEVEL LANGUAGE



PICK A CARD



WRITE

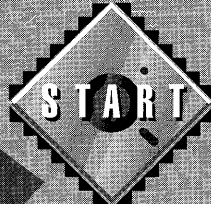


MAKE
GO DIRECTLY TO RUN



ASSEMBLE

PROGRAMMER'S WORKBENCH



Chapter 1

The Professional Development System

The Microsoft Macro Assembler Professional Development System is a collection of tools designed to increase DOS and OS/2 programming efficiency. The system offers

- The Microsoft Macro Assembler (MASM)
- A customizable editor that can accept user extensions
- A source-level, window-oriented debugger
- A project-management (“make”) utility
- A source-level browser
- A complete online reference system

The Programmer’s WorkBench (PWB) integrates these tools, creating an environment in which you can edit, assemble, link, and debug your programs.

This book is a general introduction to MASM. This chapter introduces some of the features of the Professional Development System. Chapter 2 describes how to install the Professional Development System. Chapter 3 introduces the Programmer’s WorkBench. Chapter 4 shows how to use the online reference system—a complete reference to the Professional Development System and the MASM language that is accessible both within PWB and at the OS/2 or DOS command line.

NOTE Microsoft documentation uses the term “OS/2” to refer to the OS/2 systems—Microsoft Operating System/2 (MS® OS/2) and IBM OS/2. Similarly, the term “DOS” refers to both the MS-DOS and IBM Personal Computer DOS operating systems. The name of a specific operating system is used when it is necessary to note features that are unique to that system.

If you'd like further information about a topic as you read through this book, you can consult

- The Microsoft Macro Assembler *Programmer's Guide*
- The Microsoft Macro Assembler *Reference*
- The online reference system (discussed in Chapter 4 of this book)

Before You Proceed

Before installing MASM, you should make sure that your computer meets minimum system requirements and that your package is complete.

System Requirements

MASM requires the following configuration:

- A personal computer running DOS version 3.0 or later or OS/2 version 1.1 or later.
- 640K (kilobytes) of available memory (RAM) for operating under DOS (1 megabyte of RAM is recommended).
- 3 megabytes of RAM for operating under OS/2 (4 megabytes are recommended).
- At least 384K of extended memory if you want to debug large DOS programs.
- A hard-disk drive with at least 4 megabytes of free space. (The actual space required depends on the options you select.)
- A floppy-disk drive.

Programs assembled with MASM version 6.0 run under DOS versions 2.1 and later.

Package Contents

Check your MASM package to see if everything is there. If any pieces are missing, contact the retailer from whom you purchased the package.

In the package, you should find the following items:

- **Registration card.** There are many advantages to being a registered owner of MASM 6.0, including notification of future releases and easier access to customer assistance. Please take the time to fill out and mail the registration card now. (If you are already a registered owner of an earlier version of MASM, a registration card is not included with the update.)
- **Disks.** The distribution disk labeled “Setup” contains the PACKING.LST file, which lists the location and description of all disk files in the MASM package. Disk files are compressed; the SETUP program decompresses them as they are installed.
- *Installing and Using the Professional Development System.* The book you’re reading. It explains how to install and start to use the Professional Development System.
- *Programmer’s Guide.* This book describes MASM features and explains how they work. It also shows how to write optimal MASM code.
- *Reference.* This book describes assembler directives, operators, and instructions, including timing and encoding data.
- **Product Assistance Request.** If you need to contact Microsoft Product Support, be sure to fill out this questionnaire (which is bound into the *Programmer’s Guide*) before calling.
- **Documentation Feedback Card.** To help Microsoft improve its documentation, this postage-paid survey mailer is included in the *Programmer’s Guide*. Please take the time to fill out the card with any comments or suggestions.

Features New to Version 6.0

If you’ve used an earlier version of MASM, you’ll find many new capabilities in version 6.0, and you’ll discover that you can perform familiar operations more quickly:

- Speed up development with the integrated Programmer’s WorkBench.
- Build programs more easily with the new “make” facility (NMAKE) and PWB’s powerful project capabilities.

- Track down program bugs and logic errors more quickly with the new Microsoft CodeView® debugger version 3.12. With an 80286 or 80386 processor and 1 megabyte or more of RAM, you can debug a program up to 640K in size in real mode.
- Access MASM language, assembler, linker, and utility documentation with the Microsoft Advisor and QuickHelp online reference systems.
- Cross-reference program procedures with the PWB Source Browser.
- Create OS/2 applications with increased support for dynamic-link libraries (DLLs), multiple threads, and improved debugging options.
- Edit source code with PWB's mouse- and window-oriented editor.

For a detailed list of new language features, see Appendix A, "Differences between MASM 6.0 and 5.1," in the *Programmer's Guide*.

Microsoft Programmer's WorkBench

The Programmer's WorkBench can cut development time and increase your productivity. Its integration allows you to edit, build, run, and debug a program without ever leaving PWB. Its multiwindow display lets you edit several files at once or access online help while still viewing your code. Drop-down menus and mouse support let you perform most functions by clicking the mouse on menu names or by pressing a few keys.

The PWB editor is easy to configure and completely extensible. It combines powerful features such as macros and regular-expression search capabilities with the simplicity of a window-based application (see Figure 1.1).

The Microsoft Advisor online reference system provides quick access to information about MASM, PWB, CodeView, the assembler, linker, utilities, and even the reference system itself. Figure 1.2 shows online information about MASM's **INC** instruction.

```

File Edit View Search Make Run Options Browse Help
D:\MASM60\MISCDEMO.ASM
: Preserve original screen and put up window
: Pass top, left, bottom, right, and attribute
INVOKE WinOpen, 0, 0, vconfig.rows, 79, 07h

D:\MASM60\MISCDEMO.LST
: Preserve original screen and put up w
: Pass top, left, bottom, right, and at
INVOKE WinOpen, 0, 0, vconfig.rows, 79
0034 33 C0 * xor ax, ax
0036 50 * push ax
0037 33 C0 * xor ax, ax
0039 50 * push ax
003A A0 0002 E * mov al, byte ptr DGROUP:vconfig+002h
003D 32 E4 * xor ah, ah
003F 50 * push ax
0040 BB 004F * mov ax, +00000004Fh
0043 50 * push ax
0044 BB 0007 * mov ax, +00000007h
0047 50 * push ax
0048 EB 0000 E * call WinOpen
<F1=Help> <Alt=Menu> <F6=WINDOW> macro 00206.053

```

Figure 1.1 Editor Window

```

File Edit View Search Make Run Options Browse Help
Help: INC Instruction
<Detail> <Key> <Example> <Up> <Contents> <Index> <Back>

Title: Increment Flags: O D I T S Z A P C
Syntax: INC dest ± ± ± ± ±
See also: DEC, SUB
Description:
Adds 1 to the destination operand. Because the operand is treated
as an unsigned integer, the INC instruction does not affect the
carry flag. If you need to test the carry flag, use the ADD
instruction.
-----
D:\MASM60\MISCDEMO.ASM
inc WinMsg[LEN3] ; Increment window number
mov bx, Row1
add bl, 2 ; Message row
<F1=Help> <Alt=Menu> <F6=WINDOW> macro 00621.012

```

Figure 1.2 Help Window

When you're ready to assemble the program, you don't need to remember the full range of command-line assembler options. All options are conveniently located in a PWB dialog box where they can be clicked on or off (see Figure 1.3). Setting options is simply a matter of using the mouse or keyboard to set case, warning level, debug, or release options.

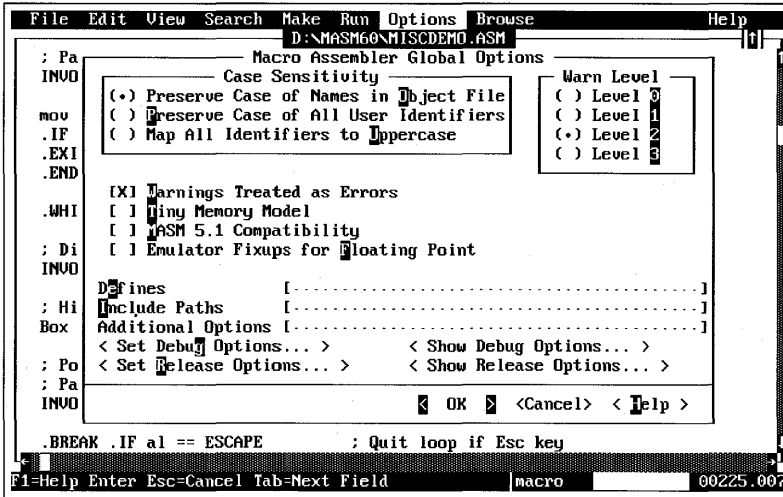


Figure 1.3 Macro Assembler Global Options Dialog Box

PWB uses the powerful NMAKE tool to speed up development. NMAKE examines each module of a program to determine whether changes have been made that require the module to be reassembled. NMAKE then automatically assembles the modules as necessary and links them. When you're ready to build your program, you just select a single menu command to start the process. PWB invokes NMAKE and builds an application, using the options you set.

If warnings or errors occur during assembly or linking, PWB opens a window with a full list of assembly or link errors (see Figure 1.4). You can then consult the on-line reference system for comprehensive information about each error.

```

File Edit View Search Make Run Options Browse Help
D:\MASM60\MISCDEMO.ASM
cmp vconfig.rows, 49 ; Current mode 50 lines?
je toggle25 ; Yes? Toggle UGA to 25-line
cmp vconfig.row, 42 ; Current mode 43 lines?
jne toggle43 ; No? Must be 25
cmp vconfig.adapter, EGA ; Yes? And is adapter EGA?
je toggle25 ; Yes? Then toggle to 25 line
mov al, 50 ; No? Toggle UGA to 50 line
jmp toggle25
toggle43:
mov al, 43 ; If currently 25 lines, make
; either EGA or UGA 43 lines
toggle25:
; Change line mode, pass lines
INVOKE SetLineMode, ax

```

```

Compile Results
MISCDEMO.asm(542): error A2006: undefined symbol : row

```

<F1=Help> <Alt=Menu> <F6=WINDOW> macro 00541.065

Figure 1.4 Error Window

Once a program has been successfully built, you can run it from within PWB to test it. If you need to make modifications, you can return directly to PWB to make the changes.

Another useful PWB feature is the Source Browser. With the Browser you can examine your code and display information about procedure and variable relationships (see Figure 1.5). This option is helpful for tracking program flow and isolating bugs.

For bugs that are more difficult to locate and understand, use PWB's Debug Build option, then invoke the CodeView debugger directly from PWB. You can step and trace through your code at the source level while examining variables, registers, and memory (see Figure 1.6).

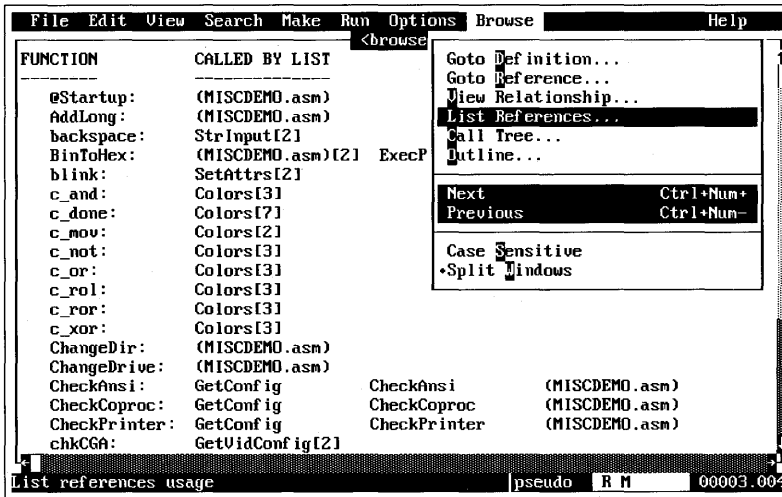


Figure 1.5 Browser Window

Once you locate the bug, you can exit CodeView and return directly to PWB, where you can fix the error. When the code is completely debugged, you can build a final release program using PWB's Release Build option.

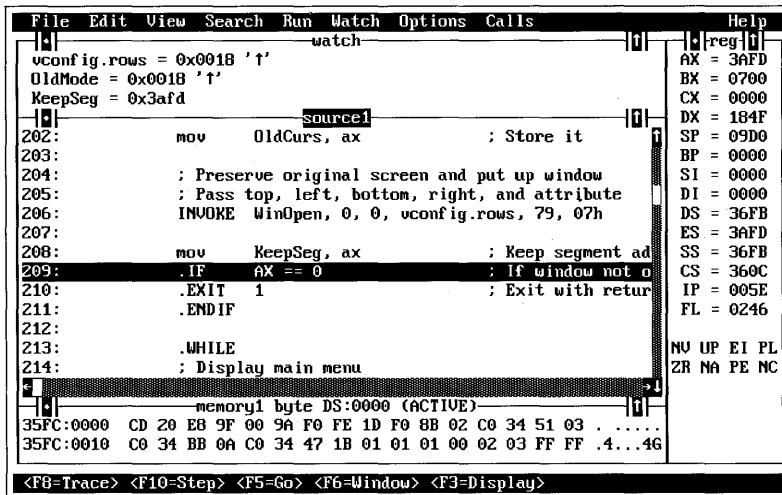


Figure 1.6 CodeView Window

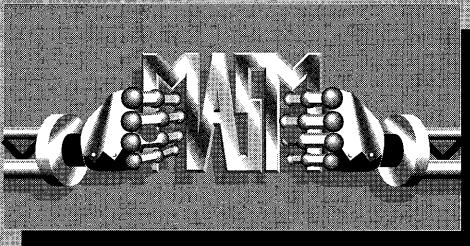
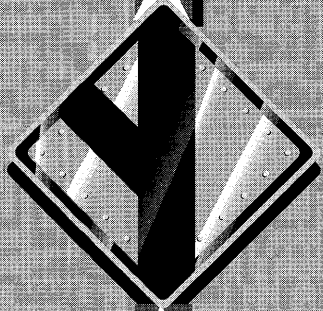
Key to Document Conventions

This book uses the following document conventions:

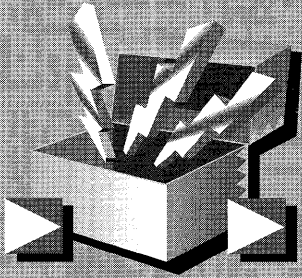
<u>Example</u>	<u>Description</u>
COPY TEST.ASM C:	Uppercase letters represent DOS commands and filenames.
INVOKE	Boldface letters indicate standard features of the MASM language: keywords, operators, and standard library functions.
<i>expression</i>	Words in italics indicate placeholders for information you must supply, such as a filename. Italics are also occasionally used for emphasis in the text.
ML /Zi HELLO.ASM	This typeface is used for example programs, program fragments, and the names of user-defined functions and variables. It also indicates user input and screen output.
SHIFT	Small capital letters denote names of keys on the keyboard. A plus sign (+) indicates a combination of keys. For example, SHIFT+F5 tells you to hold down the SHIFT key while pressing the F5 key.
“bookmark”	The first time a new term is defined, it is enclosed in quotation marks. Since some knowledge of programming is assumed, common terms such as memory or branch are not defined.
Programmer’s WorkBench (PWB)	The first time an acronym appears, it is spelled out.



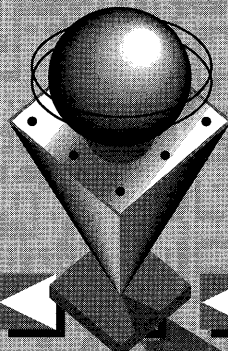
PICK A CARD



ASSEMBLE



UNMAKE
GO DIRECTLY TO RUN



Chapter 2

Installing the Microsoft Macro Assembler

This chapter explains how to install the Microsoft Macro Assembler (MASM), configure extended memory for use with the CodeView debugger, and customize components of the Programmer's WorkBench.

You must run the installation program (SETUP.EXE) to install MASM. The files on the distribution disks are compressed; SETUP both uncompresses the files and copies them to your hard disk. SETUP runs under both DOS and OS/2.

Before running SETUP, back up the distribution disks and make sure you have enough disk space (at least 4 megabytes, depending on the options you choose) to install the Professional Development System.

Running SETUP

Insert the disk with SETUP in the appropriate disk drive. Make that drive the current drive. Type `SETUP` and press `ENTER` to begin installation.

If you aren't sure how you want to set up MASM 6.0, the default responses to SETUP's questions are usually a good starting point. (You can run SETUP again at any time to install a different configuration.) Press `ENTER` to proceed.

From the Main Menu of SETUP (see Figure 2.1), you can

- Install the Microsoft Macro Assembler
- Install the Macro Assembler using defaults
- Run SETUP without installing any files
- View important documentation notes (README.DOC)
- View the packing list (PACKING.LST)
- Copy a file from the distribution disks
- Exit SETUP

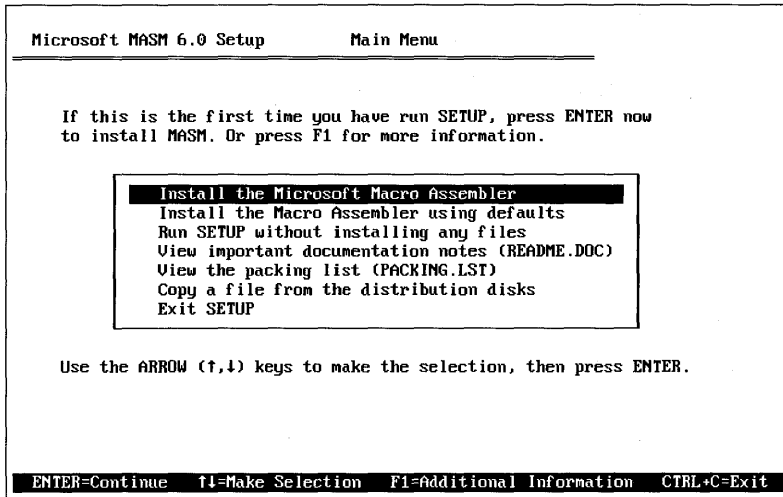


Figure 2.1 Main Menu of SETUP

“Install the Microsoft Macro Assembler” is highlighted as the default. If you want a different function, use the UP and DOWN ARROW keys to highlight the desired function; then press ENTER.

The first time you run SETUP, press ENTER to begin installation. If you want to become familiar with SETUP and its options, select the third menu item, “Run SETUP without installing any files.”

The fifth menu item, “Copy a file from the distribution disks,” allows you to transfer specific files without having to go through the full installation.

To exit SETUP at any time, press CTRL+C. SETUP never erases files from the distribution disks, so you can quit and start over as often as you need to.

Supplying SETUP Information

The SETUP screens following the Main Menu request information about your operating system, hardware configuration, and directory paths. After you have answered all the questions, SETUP begins transferring files to your hard disk.

Most questions (such as which operating system you’re working under or whether you use a mouse) have a limited number of specific responses. These are listed in a box, with the default response highlighted. The following sections provide information about individual SETUP questions.

If you want to accept the default response, press ENTER. If not, use the UP and DOWN ARROW keys to select a different response; then press ENTER. In either case, pressing ENTER advances you to the next screen.

The remaining questions prompt you for the directories in which the assembler, linker, debugger, include files, and other files are to be placed. SETUP suggests a default path, which you can accept by pressing ENTER.

If you don't want to use the default directory, use the BACKSPACE key to erase it; then type in the directory you want. (You can also change the hard-disk specification if you want to install MASM 6.0's components on different drives.) Press ENTER to accept the directory and advance to the next screen.

If you specify a directory that does not exist, SETUP automatically creates it for you. Therefore, be careful to enter the correct directory name.

Changing Your Response

You can change responses as often as you like before pressing ENTER and moving to the next screen. Don't worry if you make a mistake or change your mind. The last screen of SETUP displays all the options you have selected; you can change any of them before the installation actually begins.

Responding to the Prompts

The SETUP questions are largely self-explanatory. The following sections supply additional information that might be useful to you.

Choosing the Host Operating System

There are two versions of MASM; one runs under DOS and the other under OS/2. Select the versions for the operating systems you use. If you want to run MASM under both DOS and OS/2, you can install both.

MASM can create executable files for OS/2 or DOS, regardless of which operating system the assembler runs under. Programs must be debugged and run under the operating system they were written for.

Installing the Programmer's WorkBench (PWB)

PWB integrates editing, assembling, linking, and debugging to speed program development. You can customize the PWB editor to accommodate your own working style and needs.

Installing the Programmer's WorkBench is optional. You can call the assembler and linker from the command line, and run CodeView and the other tools separately.

Emulating BRIEF

If you install PWB, you can choose to configure the PWB editor so that most of its editing functions are assigned to the same keystrokes that the BRIEF editor uses. If you decide not to configure PWB this way during installation, you can add these (or other) customizations later.

Installing the New MASM.EXE Utility

MASM version 6.0 now includes ML.EXE, which replaces MASM.EXE. The ML.EXE utility both assembles and links, making the assembler's behavior identical to that of Microsoft compilers (such as CL.EXE for C and FL.EXE for FORTRAN). You can still assemble without linking by adding the /c option to the ML.EXE command line.

To simplify the transition from previous versions of MASM to MASM version 6.0, Microsoft has provided a special MASM.EXE utility. The new MASM.EXE is not an assembler. It simply translates the old-style command line into the new format and passes the command to ML.EXE.

Since MASM.EXE also sets certain options, you may want to look at Appendix A in the *Programmer's Guide*, "Differences between MASM 6.0 and 5.1," before using MASM.EXE. This appendix lists all the new and changed features in MASM 6.0 and gives recommendations for updating existing code.

Copying Documentation Files and Sample Programs

Having SETUP copy documentation files and sample programs to your hard disk makes them available at any time. You can load them into PWB or a text editor, or print them for convenient reading. You can also read the documentation files from

within SETUP by selecting the “View important documentation notes” option from the function screen.

Only the README.DOC and the PACKING.LST files can be copied directly from the distribution disks without SETUP. From the Main Menu in SETUP, you can also select the “Copy a file from the distribution disks” option. This decompresses and copies files to the current directory.

Copying the Microsoft Mouse Driver, MOUSE.COM

If you choose DOS as the host mode, you are asked if you want to copy the most recent version of the Microsoft mouse driver, MOUSE.COM. (The mouse interface is built into the OS/2 environment.) This driver is written for the Microsoft Mouse, but it will work with any 100%-compatible mouse or pointing device.

If you currently use a Microsoft Mouse, you already have a mouse driver on your hard disk. SETUP places the mouse driver in the directory for real-mode files that you specify later. This directory is probably not the location of your current mouse driver. If you want to use the new driver, be sure to copy it to the appropriate directory after you leave SETUP.

Choosing the Target Hard Disk

Specify the hard disk on which you want to install the Professional Development System. It becomes the default drive for the directories SETUP will suggest later on. However, if you want to install the Development System components on more than one drive or partition, you can change the drive designation in the suggested directories.

Choosing Bound, Real-Mode, and Protected-Mode Directories

Some of the software supplied with the Professional Development System is “bound,” which means it runs under DOS or OS/2. However, five system components come in two versions: one for DOS (or OS/2 real mode), the other for protected mode for OS/2. These components are ML, PWB, NMAKE, QH, and CodeView.

If you set up for only one mode, it doesn't matter where you place these components. However, ML, PWB, QH, and NMAKE use the same name for their real- and protected-mode versions. If you set up for both real and protected mode, you

cannot place these components in the same directory, because you cannot have two identically named files in the same directory.

Therefore, SETUP requires that you select different directories for the bound, real-mode, and protected-mode components. If you change SETUP default directories, be sure you don't choose the same directory for both the real- and protected-mode components.

Note that if you have chosen to install only for DOS or OS/2 real mode, you are not prompted for a protected-mode directory. Likewise, if you have chosen to install only for OS/2 protected mode, you are not prompted for a real-mode directory.

Selecting File Directories

SETUP wants to know where to place help, include, initialization, and library files (in OS/2). The corresponding environment variables in your AUTOEXEC.BAT or STARTUP.CMD file should point to the correct directories. This is explained in the section called "Configuring Your System," later in this chapter.

Reviewing and Changing Responses

The Status and Change screen is the last screen before SETUP actually begins installation. This screen displays all your responses. If you are satisfied, press ENTER to select the "No Changes" default, and installation will begin.

If you want to change a response, press the DOWN ARROW key to move the highlight to the appropriate line; then press ENTER to display the screen that controls this response. Change your response; then press ENTER again to return to the status and change screen. Your new response for that option is now displayed.

You can repeat this process as often as you want. When all options are the way you want them, select "No Changes" and press ENTER to begin installation.

Installing

SETUP checks to see if your system has adequate space available and then prompts you for each distribution disk it requires. The only likely error is inserting the wrong disk. If this occurs, SETUP prompts you for the correct disk.

Configuring Your System

At the end of installation, SETUP creates system-configuration files. These are listed in Table 2.1, along with SETUP's default directories:

Table 2.1 SETUP Configuration Files

Operating System	Configuration Files	Default Directory
DOS or OS/2 real mode	NEW-VARS.BAT, NEW-CONF.SYS	C:\MASM\BIN
	TOOLS.PRE	C:\MASM\INIT
OS/2 protected mode	NEW-VARS.CMD	C:\MASM\BINP
	TOOLS.PRE	C:\MASM\INIT

If you add the contents of these files to your AUTOEXEC.BAT (or STARTUP.CMD), CONFIG.SYS, and TOOLS.INI files, your system will be correctly configured for MASM each time you start your computer. You can also run NEW-VARS.BAT or NEW-VARS.CMD to set environment variables before you run MASM.

Modifying AUTOEXEC.BAT or STARTUP.CMD

The NEW-VARS.BAT and NEW-VARS.CMD files contain commands that set environment variables. Insert these commands in the AUTOEXEC.BAT or STARTUP.CMD file. All changes to environment variables are prefixed to the current settings. Table 2.2 explains these variables.

Table 2.2 Environment Variables

Variable	Description
ASMEX	Location of assembly-language example files
HELPPFILES	Location of help (.HLP) files
INCLUDE	Location of include (.INC) files
INIT	Location of initialization (.INI) files
MASM, ML	List of command-line options requested
PATH	Path to search for executable files
QH	Location of QuickHelp help (.HLP) files
TMP	Location of PWB/LINK temporary files

The online reference system searches for help files in a specific sequence. Therefore, do not modify the `HELPPFILES` variable unless you are familiar with the search sequence. Changing this path might make the online reference system inaccessible.

Modifying CONFIG.SYS

If the host operating mode is OS/2 real mode or DOS, check the values of `files` and `buffers` in `CONFIG.SYS`. The values in `NEW-CONF.SYS` are minimums; the `CONFIG.SYS` values should be at least as large.

Under OS/2, be sure the `LIBPATH` variable includes the directory of the help system's dynamic-link library `MSHELP.DLL`. OS/2 users who want to use CodeView must also include the statement `IOPL=YES`.

To use the extended memory features of CodeView under DOS, you must use the `HIMEM.SYS` driver supplied with this product. Add `DEVICE=HIMEM.SYS` to `CONFIG.SYS`. Be sure to specify `HIMEM`'s full path. See the section later in this chapter called "Configuring Extended Memory for CodeView" for additional information.

Modifying TOOLS.INI

The `TOOLS.PRE` file contains additional PWB configuration settings. Add the contents of `TOOLS.PRE` to your existing `TOOLS.INI` file. If you don't already have a `TOOLS.INI` file, rename `TOOLS.PRE` to `TOOLS.INI`.

PWB looks for `TOOLS.INI` in the directory containing PWB. If you move `TOOLS.INI` to another directory, specify its path in the `INIT` environment variable in your `AUTOEXEC.BAT` or `STARTUP.CMD` file.

The TOOLS.INI file controls many configuration options for MASM 6.0 utilities. These options let you customize the following:

- PWB
- CodeView debugger
- Microsoft Advisor online reference system
- NMAKE utility

You can modify the TOOLS.INI settings with the PWB editor or any word processor. These are some of the options you can change:

- Screen colors and the number of lines displayed by PWB
- Macros for PWB
- Key assignments for PWB
- Default key settings for PWB
- Location of help files used by the online reference system
- Options for NMAKE

TOOLS.INI is described in the online reference system. See Chapter 14 of the *Programmer's Guide* for more information about customizing PWB.

Customizing the Programmer's WorkBench

There are other ways to customize PWB besides editing the TOOLS.INI file. For example, you can control a variety of editor functions and options from within PWB with the Editor Settings command from the Options menu.

For more information about customizing the Programmer's WorkBench, see Chapter 3 in this manual, and Chapter 14 of the *Programmer's Guide*, "Customizing the Microsoft Programmer's WorkBench." For more information about customizing NMAKE and other utilities, see the online reference system or the *Programmer's Guide*.

Configuring Extended Memory for the CodeView Debugger

Extended memory is the memory above the first megabyte that an 80286, 80386, or 80486 can access in protected mode. (*Extended* memory is not the same as *expanded* memory. Expanded memory is bank-switched memory that overcomes the 640K RAM limit of machines with 8086/8088 processors.)

To use the extended memory features of the CodeView debugger under DOS, you must first add `DEVICE=HIMEM.SYS` (with HIMEM's full pathname) to the `CONFIG.SYS` file, then reboot. The `NEW-CONF.SYS` file created by `SETUP` contains a sample HIMEM entry.

NOTE HIMEM.SYS implements the XMS 2.x standard. The complete XMS source and specification are available from the Microsoft Information Center.

There are several additional factors to consider when using extended memory with CodeView under DOS or Windows:

- CodeView needs at least 384K of extended memory. If you add memory to your computer, be sure to configure at least 384K of it as extended memory.
- Older versions of HIMEM.SYS may not work properly with this product. Use the version of HIMEM.SYS on the MASM 6.0 distribution disks.
- Other memory managers such as 386-Max or QEMM may conflict with HIMEM.SYS. Don't install more than one memory manager.
- Don't use HIMEM.SYS (or any other memory manager) with Windows/386. Windows/386 allocates extended and expanded memory automatically.

- If you run Windows 3.0 in 286-Protected mode, create a PIF for CodeView and specify at least 384K of XMS (extended) memory.
- Many RAM-disk and disk-cache programs are incompatible with HIMEM.SYS. The RAMDRIVE and SMARTDRV programs supplied on the distribution disks are HIMEM.SYS-compatible.
- If you run TSRs, check their documentation to see if they are compatible with HIMEM/XMS 2.x. If there is any question about compatibility, manually install one TSR at a time.

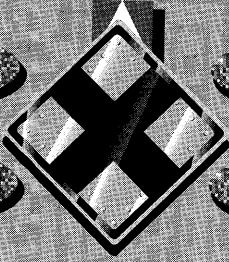
SETUP places the HIMEM, RAMDRIVE, and SMARTDRV drivers in the C:\MASM\BIN directory or in the directory you have specified for real-mode executable files.

BASIC

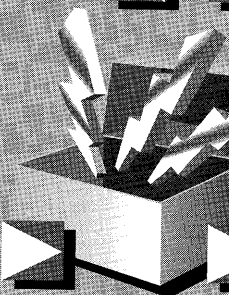
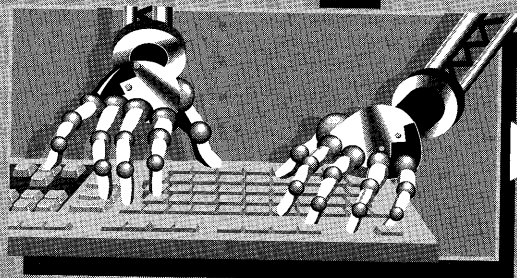
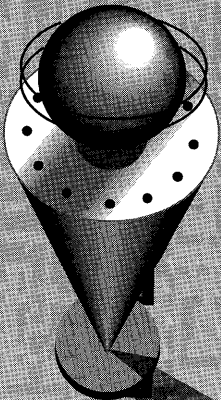
PASCAL

FORTRAN

C

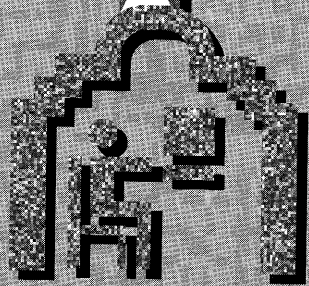


PICK A HIGH-LEVEL LANGUAGE

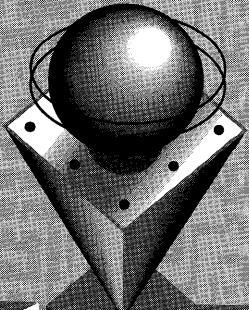
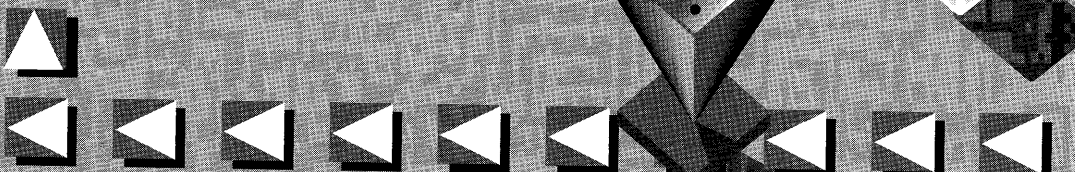


WRITE

**NMAKE
GO DIRECTLY TO RUN**



PROGRAMMER'S WORKBENCH



Chapter 3

Using the Programmer's WorkBench

The Programmer's WorkBench (PWB) is a window-oriented programming environment that incorporates a text editor, an assembler, a linker, a debugger, a make utility, a source-code browser, and an online reference system. It is an alternative to switching between command-line-based programs: you can edit, assemble, link, and debug without leaving this integrated environment. To demonstrate PWB's features, this chapter includes a sample program that you can assemble, link, and debug.

This chapter explains how to start PWB and then introduces the following PWB elements:

- **Windows and menus.** You can quickly find the command you need using the menus. The section "Using Windows and Menus" explains how to open and close windows and how to navigate through the menus.
- **Programmer's editor.** Features such as "bookmarks," macros, customized key commands, and enhanced search capabilities decrease the amount of time you spend writing code.
- **Integrated assembler and linker.** You can assemble, link, and run a program without leaving the editor. PWB's integrated environment and project-management facilities save you hours of development time.
- **Integrated browser and debugger.** Using the Source Browser, you can quickly find a data declaration or procedure definition in your source code, as well as references to all procedures and variables in the program. With the CodeView debugger, you can set breakpoints, examine variables and machine registers, and step through execution of your program one line at a time.

Starting PWB

To run PWB, type

```
PWB
```

at the command line. You can immediately begin entering source code in the untitled window that appears. PWB maintains a history of the files you have opened. The next time you run PWB, it opens the last file you worked on.

Specifying a Source File

You can open an existing source file by specifying its name or pathname after the PWB command. If PWB can't find the specified file, it asks if you want to create a new file with that name. You can also open an existing source file with the Open command from the File menu.

Command-Line Options

PWB offers a number of command-line options that can configure PWB, position the file at a bookmark, or perform user-selected functions automatically. These are documented in the *Reference* and in the "Starting PWB" topic in online help.

Elements of the PWB Environment

This section introduces the PWB environment and describes how to control windows and choose commands from the menus. It also describes elements within the PWB environment.

You can enter PWB commands from the keyboard or with a Microsoft (or fully compatible) Mouse. Unless the right mouse button is specifically mentioned, "clicking" means to press and release the left mouse button once.

Windows and Other Screen Elements

Figure 3.1 shows the components of a typical PWB screen. Some elements provide information only. For example, if CAPS LOCK is on, the letter C appears in the lower-right corner of the screen. Other elements perform actions triggered by a specific keystroke or mouse action. For example, if you click the button in the upper-left corner of a window with the mouse, the window closes.

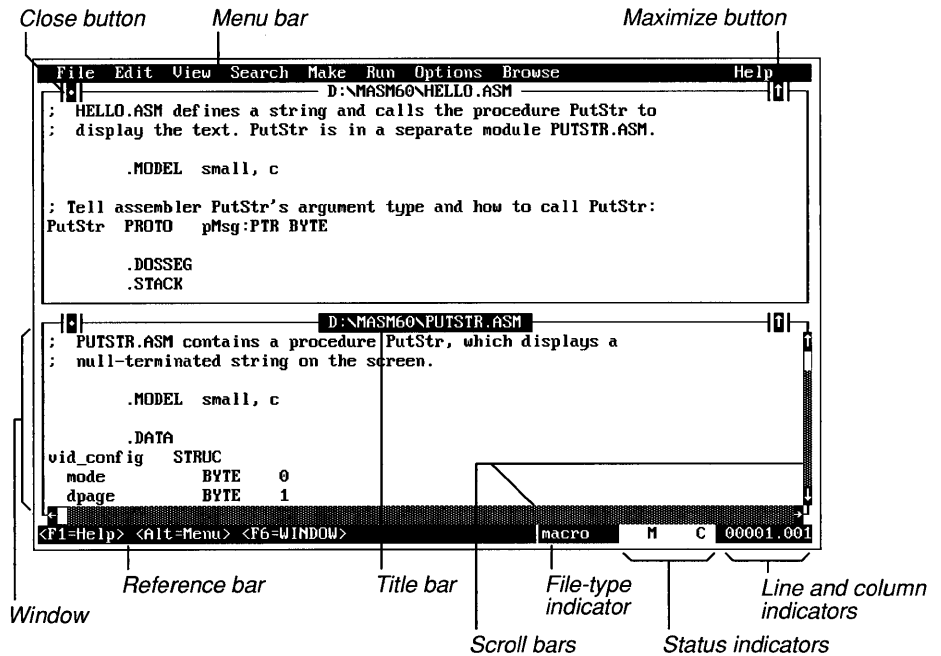


Figure 3.1 Parts of a Screen

The parts of a PWB screen and their uses are listed in Table 3.1.

Table 3.1 Parts of a PWB Screen

Name	Use
Menu bar	Lists names of available menus.
Close button	Closes window (appears only if more than one window is displayed).
Title bar	Shows name of file currently being edited.
Windows	Contain source code or display information associated with on-line help.
Maximize button	Enlarges or restores window to its original size.
Scroll bars	Indicate cursor position in the current file and allow cursor movement.
Reference bar	Lists shortcut keystrokes (keyboard users) and direct PWB commands (mouse users); summarizes menu contents and displays other information.

Table 3.1 (continued)

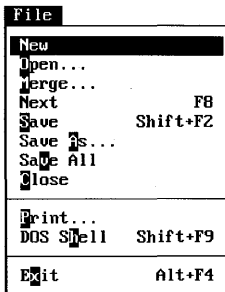
Name	Use
File-type indicator	Tells type of file. ASM: MASM source code; text: any other user-created file; pseudo: file-like means of displaying data.
Line and column indicators	Show current line and column of text cursor.
Status indicators	A : Meta prefix is set. B : Background assembly in OS/2. C : CAPS LOCK is on. L : Carriage return isn't used to terminate a line. M : File has been modified. N : NUM LOCK is on. O : Overtyping is on. R : File is set to Read Only status. T : File is temporary. X : A macro is being recorded.

Menus

PWB commands are organized into menus whose names appear in the menu bar. A brief description of the selected menu appears in the reference bar. To get more information about a menu, select the menu and press F1, or point the mouse cursor at the menu name and click the right mouse button.

PWB has the following menus:

File Menu



New	Clears the Source window to start a new file
Open	Loads an existing source file
Merge	Merges one or more files with the current file
Next	Displays the next file in the file list specified at start-up
Save	Saves the current file
Save As	Saves the current file under a different name
Save All	Saves all modified files
Close	Closes the current file

Print	Prints a selection or the current file
DOS Shell	Temporarily exits to DOS or OS/2
Exit	Leaves PWB

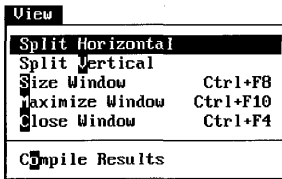
At the bottom of the File menu, PWB displays a list of recently opened files. This helps you access the files you've been working with.

Edit Menu

Edit	
Undo	Alt+Bksp
Redo	
Repeat	
Cut	Shift+Del
Copy	Ctrl+Ins
Paste	Shift+Ins
Clear	Del
Set Anchor	
Select To Anchor	
Box Mode	
Read Only	
Set Record...	
Record On	Shift+Ctrl+R
Edit Macro	

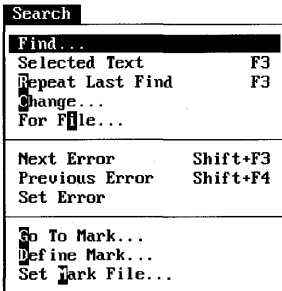
Undo	Reverses the effect of your recent edits
Redo	Reverses the effect of the last Undo command
Repeat	Repeats your last edit
Cut	Deletes a selected block of text from the active window and copies it to the clipboard
Paste	Inserts a selected block of text from the clipboard into the active window
Clear	Deletes selected text without copying it to the clipboard
Set Anchor	Saves the current cursor position as a reference point for text selection
Select to Anchor	Highlights text from the previously set anchor to the current cursor position
Box Mode	Toggles text selection mode between box, line, and stream modes
Read Only	Makes all files read-only (to protect from accidental modification)
Set Record	Defines a macro name and its shortcut key
Record On	Records keystrokes for a macro
Edit Macro	Edits a recorded macro

View Menu



- Split Horizontal Divides the active window horizontally.
- Split Vertical Divides the active window vertically.
- Size Window Enlarges or shrinks the active window.
- Maximize Window Toggles between Maximize Window and Restore Window. (Maximize Window enlarges the active window, while Restore Window restores it to its original size.)
- Close Window Closes the active window.
- Compile Results Shows the result of the last compilation.

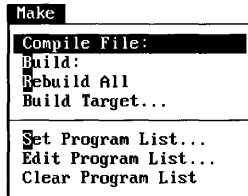
Search Menu



- Find Searches for the next occurrence of a text string in the active window
- Selected Text Searches for the next occurrence of the current selection
- Repeat Last Find Repeats the search without retyping the text string
- Change Searches for one text string and replaces it with another
- For File Searches for a file on disk
- Next Error Moves to the next build error in source code
- Previous Error Moves to the previous build error in source code
- Set Error Selects the error at the cursor as the current error (i.e., synchronizes active window and error window)
- Go to Mark Moves the cursor to a "bookmark" (a previously specified position in a file)

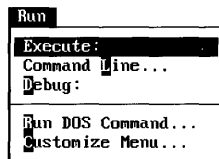
- Define Mark Creates a bookmark by associating a name with a position in a file
- Set Mark File Creates a file in which you can save bookmarks or open an existing bookmark file

Make Menu



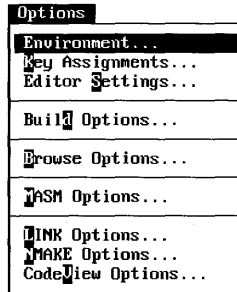
- Compile File Compiles the current source file
- Build Compiles and links all modified files in a multimodule program
- Rebuild All Compiles and links all files (even those that have not been modified)
- Build Target Builds a single target from the program list or non-PWB makefile
- Set Program List Creates or opens a program list file, which defines the components of a multimodule program
- Edit Program List Changes the contents of a program list
- Clear Program List Removes the current program list from memory, but does not change its contents

Run Menu



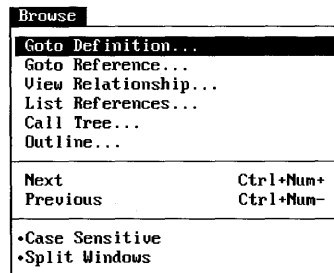
- Execute Runs the current program
- Command Line Specifies commands that will be supplied to your program when you choose Execute or Debug from the Run menu
- Debug Invokes the Microsoft CodeView debugger
- Run DOS Command Performs any single DOS or OS/2 task without exiting the environment
- Customize Menu Adds commands to the Run menu

Options Menu



Environment	Specifies the paths to search for include files, libraries, and online help
Key Assignments	Assigns keystrokes that invoke commands, macros, and extension functions
Editor Settings	Changes the setting of any editor switch
Build Options	Sets the main language, determines what type of program is built, specifies whether the program is built as a debug or release version, and saves current build options
Browse Options	Defines the way the PWB Source Browser database is built
MASM Options	Sets compiler options for your project
LINK Options	Sets linker options for your project
NMAKE Options	Sets the command line for NMAKE
CodeView Options	Controls how CodeView uses memory, displays information, and handles display of output

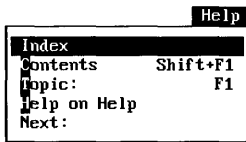
Browse Menu



Goto Definition	Locates the definition of any program symbol in your source code
Goto Reference	Locates the references to any name in the source database
View Relationship	Queries the PWB Source Browser database
List References	Displays a list of functions that call each function and use each variable, type, and macro
Call Tree	Views which functions call other functions
Outline	Displays a program outline

Next	Finds the next definition or next reference
Previous	Finds the previous definition or reference
Case Sensitive	Defines whether or not your searches will be case sensitive
Split Windows	Determines how the Browser appears

Help Menu



Index	Displays a list of indexes for the online reference
Contents	Displays a table of contents for subjects covered by the online reference
Topic	Displays information about the item or keyword on which the cursor is positioned
Help on Help	Explains how to use the help system
Next	Displays the next piece of information in the help system on the topic you looked at

Choosing Menu Commands

Keyboard ► To choose a command:

1. Press the ALT key to activate the menu bar.
2. Press the highlighted character in the menu name (such as F for File), or use the RIGHT and LEFT ARROW keys to select a menu. Press ENTER.
3. Press the highlighted character in the command name (such as S for Save in the File menu), or use the UP and DOWN ARROW keys to highlight the command and then press ENTER.

Mouse ► To choose a command:

1. Open the menu by clicking the menu name.
2. Click the command.

If you open a menu and then decide you don't want to issue a command, there are three ways to close the menu:

- Press the ESC key.
- Click the mouse outside the menu.
- Press ALT twice.

Using Shortcut Keys

Some menu items are followed by the names of keys or key combinations. You can press these "shortcut keys" to execute these commands immediately, instead of selecting them from a menu.

The reference bar at the bottom of the screen also displays commonly used shortcut keys. You can click any of these buttons to execute the indicated command. Many other shortcut keys have been assigned than are listed on menus. Assignments are also redefinable. See the Key Assignments option in the Options menu.

Shaded Commands

When a menu command is gray (rather than black), it is currently unavailable for use. For example, when PWB is first run and no programs have been assembled or linked, the Next Error and Previous Error commands in the Search menu are both disabled and are therefore shaded gray.

Ellipses and Dialog Boxes

A command followed by an ellipsis (...) needs more information before the command is executed. You enter this information in a dialog box that appears when you select the command. (A command not followed by an ellipsis executes immediately.) Figure 3.2 shows the dialog box opened by the Find command from the Search menu.

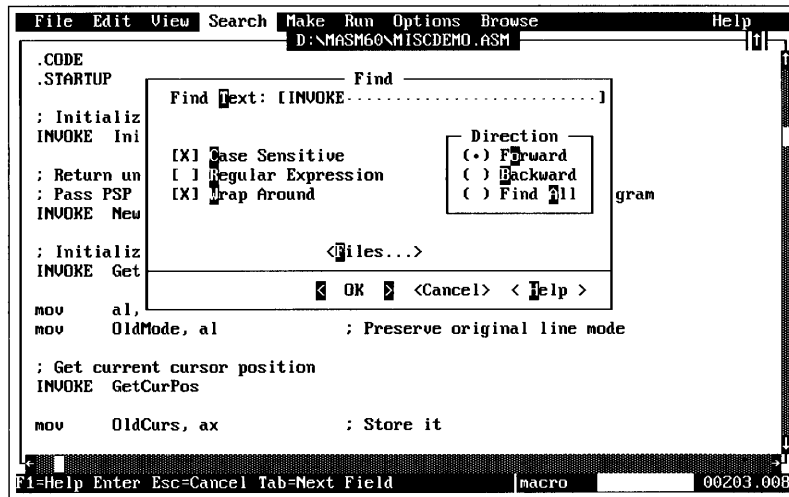


Figure 3.2 Dialog Box for the Find Command

Dialog boxes can contain any or all of the items in the following list. Press ALT and the item’s highlighted letter, or the TAB or SHIFT+TAB keys to move among items in a dialog box. You can set dialog items from the keyboard or with the mouse.

<u>Item</u>	<u>Description</u>
Option Buttons (●)	Offer a list of choices; only one option can be chosen. Use the ARROW keys to move between the choices. In Figure 3.2, three option buttons set the direction of the search (Forward, Backward, Find All).
Check Box [X]	A yes/no switch. If the box is empty, the option is turned off. If it contains the letter X, the feature is on. Press the SPACEBAR or use the UP and DOWN ARROW keys to turn a check box on or off.
Text Box [.....]	Accepts text that you type. In Figure 3.2, Find Text requires you to type the text to search for.
List Boxes	Certain dialog boxes display lists of information (such as the contents of the current disk directory) within a list box. If the number of items exceeds the list box space, press the ARROW keys, PGUP/PGDN, or click the scroll bar to move around the list.

<u>Item</u>	<u>Description</u>
Command Buttons < OK >	Confirm the settings you have entered and which are now enclosed by angle brackets. The OK button uses the current settings. The Cancel button exits the dialog box and does not change the current settings. If one of the command buttons is highlighted, press ENTER to execute that command. Clicking a command button also executes the command. If a button contains an ellipsis, it indicates that another dialog box will appear when the command is selected.

Dialog boxes usually contain shortcut keys. Shortcut keys are identified by highlighted letters.

Getting Help

PWB uses the Microsoft Advisor online reference system to provide information about PWB, the assembler, the linker, the CodeView debugger, other utilities, and topics relating to the MASM language. Information can be displayed at any time and at any point in PWB or CodeView. For a complete discussion of the online reference system, see Chapter 4.

Menu Help

To get information about a PWB menu command, choose the command with the ARROW keys, then press F1; or point the mouse cursor at the menu command and click the right mouse button. A help window appears with information about the command.

Dialog-Box Help

Many dialog boxes have a help button that provides additional information about the dialog box and its contents. To display the information, click the help button with the mouse or press F1.

The Help Menu

The commands in the Help menu can display a table of contents, an index of all help topics, and help on using the online reference system itself.

The Contents command displays the contents of the entire reference system. Information is organized by topic, such as PWB, CodeView, the MASM language, and specific utilities. Click a topic to move to that topic's table of contents.

The Index command displays every available item in the reference system, organized alphabetically. Select the first letter of the topic you want from the alphabet bar at the top of the screen. Then select the specific topic from the list that appears. Use PGUP, PGDN, and the ARROW keys to scroll through the topics.

The Help on Help command provides a brief overview of how to navigate through the help system and invoke help on any topic.

Using the Editor

The program editor is an important part of the PWB environment. This section provides a brief overview of its major functions and assumes you are using the default key assignments. For a complete list of editor commands and functions, refer to the following online reference topics, available on the PWB Contents screen:

- “Function Quick Reference”
- “Using PWB Functions”

Moving Around in a Text File

Most of the cursor-control keys position the cursor and modify the display just as you would expect them to. The HOME, END, PGUP, PGDN, and ARROW keys behave as they do in most other text editors.

If you click in the shaded area on either side of the scroll box in the scroll bar, you move the cursor one full window at a time. If you click the scroll-bar arrows, you move the cursor one row or column in the corresponding direction. If you click and drag the scroll box, you move the cursor to the corresponding relative position within the file.

Defining a Block of Text

One of the most common editing tasks is defining a block of text, usually before deleting, moving, or copying it.

Keyboard ► **To define a block of text:**

1. Move the cursor to the beginning of the block.
2. Hold down the **SHIFT** key and use any cursor-movement key (**ARROW**, **HOME**, **PGDN**, and so on) to move to the end of the block.

Mouse ► **To define a block of text:**

1. Click the first character at the beginning of the block. Continue pressing the left mouse button.
2. Drag the mouse to select the rest of the desired text. The block remains selected until you press a cursor movement key or click the mouse again.

NOTE As long as a block is defined, anything you type deletes the defined block and inserts the typed characters.

As you move the cursor or mouse to define a block, you usually select all characters from the starting cursor position to the ending cursor position. This is called “stream mode” and is the default. The Edit menu offers three text-selection modes:

- Stream mode selects all text from the starting cursor position to the ending cursor position (the default).
- Box mode selects text from a rectangle whose opposite corners are the starting and ending cursor positions.
- Line mode selects full lines of text from the line with the starting cursor position to the line with the ending cursor position.

You can select a different mode from the Edit menu. The mode displayed in the menu is the next available text-selection mode (not the current mode). You can also move through the selection modes during text selection by clicking the right mouse button while you hold down the left mouse button.

Once a block is defined, press **DEL** to erase it. You can also place the block in the clipboard buffer for later use. Press **CTRL+INS** to copy the block to the clipboard.

Press **SHIFT+DEL** to copy the block to the clipboard and erase it from the screen.
Press **SHIFT+INS** to insert the current contents of the clipboard at the cursor.

Setting Bookmarks

A “bookmark” is a permanent marker that lets you quickly return to a specific section of text. A bookmark stays with the text it marks, even if the text is moved.

To set a bookmark, first position the cursor at the location you want to mark. Then select the **Define Mark** command from the **Search** menu.

In the dialog box that appears, type the name you want to give the bookmark. (The source filename and the row and column are already entered.) Press **ENTER** to save the bookmark for the current session. To save a bookmark for later editing sessions, use the **Add To Mark File** option. The **Set Mark File** command from the **Search** menu reloads previously saved bookmarks.

To move the cursor to a bookmark, choose the **Go To Mark** command from the **Search** menu. A list of all bookmarks is displayed. Select the bookmark from the list using the **ARROW** keys, or click the selected bookmark with the left mouse button. Press **ENTER** to go to the bookmark.

Setting Anchors

An “anchor” temporarily marks the beginning of selected text. It is most often used to define a block that extends beyond the current screen.

To set an anchor, move the cursor to the desired location and choose the **Set Anchor** command from the **Edit** menu. Use the **Select To Anchor** command from the **Edit** menu to select all text between the current cursor position and the anchor. Only one anchor can be set at a time.

Searching for and Changing Text

You can search for a string of text using the **Find** command in the **Search** menu. Type the string you want to look for, then use the check boxes to set options such as wrap-around search and case sensitivity. The **Files** button allows you to specify a list of files to search through for the designated string.

Once a string has been located, use the **Repeat Last Find** command in the **Search** menu (or press the shortcut key, **F3**) to move to the next occurrence of the string. If

you specified several files to search, press SHIFT+F3 to move to the next file that contains that string.

To modify several instances of a string in the source file, use the Change command in the Search menu. Enter the target string, the replacement string, and the search options.

PWB also supports searching for and replacing text patterns using regular-expressions. You can use either the UNIX or Microsoft regular-expression syntax. For further information, see the online reference system.

PWB can also search for selected text in files not currently loaded. Select the Files command in the Find dialog box. From the File List box, select the name of a file that you want to search. Then click the Add/Delete button to place it in the File(s) Selected box. Repeat this step for all the files you want to search.

You can also enter a wildcard pattern in the File Name text box, and then use the Add Pattern button to place it in the File(s) Selected box. Press ENTER or click the OK button to begin the search. Use the Next Error and Previous Error commands to scan any text that is found.

Creating Macros

You can save blocks of text or frequently performed editing tasks as macros. Each macro is assigned a key combination that “plays back” the text or task.

For example, if several programmers are working on a file and need to keep track of who made what changes, a macro can be used to insert a comment with the programmer's name.

To create such a macro, use the Set Record command from the Edit menu to enter a macro name and its associated key combination. For this example, call the macro `modname` and assign it to the key combination CTRL+J.

Now choose the Record On command from the Edit menu. A bullet appears to the left of the menu command and an X appears in the status line, indicating that all keystrokes and menu commands are being recorded. Then type the following:

```
; j. courtney coded this
```

To stop the recording, choose Record On again, which causes the bullet to disappear. This action associates all recorded actions with the previously selected macro name. Now, each time you press CTRL+J, the comment you typed is inserted at the cursor position.

Macros are saved in the TOOLS.INI file when you save the <record> pseudofile. You can change or delete them by modifying the file. See the section “Customizing PWB” in the online reference system for additional information about macros.

Customizing the Editor

You can modify editor settings, assign keyboard commands, and emulate command sets from other editors to make the editor work the way you like. See the online reference system and Chapter 14, “Customizing the Microsoft Programmer’s WorkBench,” in the *Programmer’s Guide* for complete details.

Changing Editor Settings

The PWB editor has a variety of settings you can customize, such as word wrap, color, and width of tab stops.

View or modify the settings by choosing the Editor Settings command from the Options menu. The editor settings appear in a new window labeled “Current Assignments and Settings.” The format is *setting:value*, where *setting* is the name of the setting, and *value* is a Boolean, numeric, or text value. You can learn about a setting by positioning the cursor on it and pressing F1.

To change a setting, simply overtype the old value with a new one. The change does not take effect until you move the cursor to a different line. To save the changed assignments, press SHIFT+F2 (if the default keyboard assignments are in effect) or choose the Save command in the File menu. If you do not save, the assignments are temporary and last only for the duration of the PWB session.

To return to your source file, press F2.

Modifying Keyboard Assignments

Most editor functions (such as **delete**, **home**, and **copy**) are already assigned a keystroke combination for direct execution. You can reassign these functions to different keystrokes by choosing the Key Assignments command in the Options menu.

This command displays the current assignments on a screen labeled “Current Assignments and Settings.” Function assignments appear in the format *function:keyname*. You can learn about a PWB function by positioning the cursor on the function name and pressing F1. Press PGDN to view a list of unassigned keys following the assignment list.

To assign a new key to a function, replace *keyname* with the name of the new key. The change does not take effect until you move the cursor to a different line.

When you move the cursor, the line with the new assignment is highlighted. To make the assignment permanent, save the file by pressing SHIFT+F2 or by choosing the Save command in the File menu. This updates the TOOLS.INI file, where changes to key assignments are stored.

To remove a key assignment, assign the **unassigned** function to the key.

Using Advanced Editor Features

Most of the standard editing features in PWB are intuitive and easy to use. However, there are also many advanced editing options that allow you to customize and control the editor beyond simple menu commands.

The PWB editor incorporates all of the powerful features found in the Microsoft Editor. You can set a variety of functions and switches to further customize the PWB editor and enhance its performance. A complete list of editor functions and settings is available in the online reference system.

You can add new or altered functions and settings to the TOOLS.INI file so they are loaded when PWB starts, or you can enter them interactively while you are working in PWB.

For example, to assign a key command interactively to the editor's **curdate** function, which enters the current date at the cursor, first press ALT+A. (The display Arg[1] appears at the left end of the reference bar to show that you have pressed ALT+A once.) Type the following into the argument dialog box (note that the dialog box does not appear until you have typed the first character):

```
Curdate:SHIFT+CTRL+T
```

Press ALT+= to assign the key to the function.

Now when you press SHIFT+CTRL+T, the current date is inserted at the editor's cursor position.

For detailed information about advanced PWB features, refer to the *Programmer's Guide* and the online reference system.

Reconfiguring the Editor

PWB includes the BRIEF.INI file with key assignments that emulate the BRIEF editor. To use these key assignments, add the contents of this file to TOOLS.INI. You can also simulate command sets from other editors using the customization techniques described earlier.

Running Other Programs

One of PWB's most powerful features is its ability to run other programs without exiting to the command line. The Customize Menu command in the Run menu lets you add any program to the Run menu, where you can execute it simply by clicking its name. You can also specify any keystroke to run it directly.

After selecting Customize Menu, click the Add button in the Customize Menu dialog box that appears. The dialog box lets you type all the information PWB needs to successfully execute the program: its name, directory, command-line arguments, and so on. Note that you can give the program any name you like on the command line, and you can specify that the program run in the background.

Assembling and Linking

PWB can handle all the details of assembling and linking (building) a program for you. You first tell PWB which source files constitute the program. PWB then creates a "makefile," which specifies how these files are supposed to be assembled, compiled, and linked to create the final executable file. PWB already knows that .ASM files are to be assembled, .C files are to be compiled, and the resulting .OBJ files are to be linked, so it can handle simple projects automatically. Complex projects may require you to write your own makefile.

Once the makefile is created, you build your project by selecting the Build command. The makefile also speeds the build process with multimodule programs, since only those source files that have changed from the last build are reassembled or recompiled.

Building Programs in PWB

This section shows how to build programs in the PWB environment. If you don't have a program of your own, you might want to use the following program:

```
; HELLO.ASM defines a string and calls the procedure PutStr to
; display the text. PutStr is in a separate module PUTSTR.ASM.

        .MODEL    small, c

; Tell assembler PutStr's argument type and how to call PutStr:
PutStr  PROTO    pMsg:PTR BYTE

        .DOSSEG
        .STACK

        .DATA
msg     BYTE     "Hello, world.", 13, 10, 0 ; Null-terminated string

        .CODE
        .STARTUP                ; Initialize data and stack segments

        INVOKE   PutStr, ADDR msg ; Call external procedure

        .EXIT    0                ; Exit program

        END
```

Note that there are new directives (such as **.DOSSEG**) and new constructs (such as **INVOKE** and **PROTO**) in this and the following procedure. MASM version 6.0 has a number of new features to make assembly-language programming more like programming in a high-level language, while maintaining the compact, fast-executing code assembly language offers. These features are explained in detail in the online reference system and in the *Programmer's Guide*.

Saving a Program

To save a source file, select the File menu and choose either Save or Save As. If the file is new or you choose Save As, a dialog box prompts you for a filename (see Figure 3.3). If you choose Save with an existing file, PWB automatically saves the file under the name listed in the title bar of the Source window. To save a copy of a file under a different name, choose Save As. (The older version of the file, with its original name, is not deleted or modified.)

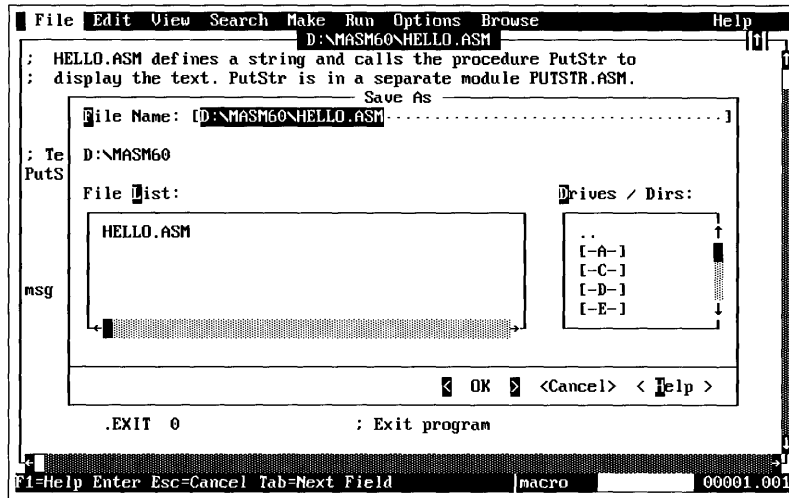


Figure 3.3 Dialog Box for the Set Program List Command

Type in the name of the source file, `HELLO.ASM`. Press `ENTER` or click the `OK` button to save the file with this name.

Writing and Saving a Procedure

`HELLO.ASM` calls the `PutStr` routine to display a message (in this case, `Hello, world`) on the screen. You could add this routine to the `HELLO.ASM` file; but because it's a procedure that could be used in other programs, you should put it in a separate file.

Use the `New` command in the `File` menu to clear the screen. Type in the `PutStr` routine that follows and save it in a file named `PUTSTR.ASM`:


```

; PUTSTR.ASM contains a procedure PutStr, which displays a
; null-terminated string on the screen.

.MODEL small, c

.CODE
PutStr PROC pMsg:PTR BYTE ; Address of string (near or far
; depending on model) is passed on
; the stack
mov ah, 02h ; Display character function

mov di, pMsg ; Load address in DI
mov dl, [di] ; Load each character through [DI]

.WHILE (dl)
int 21h ; DOS displays character
inc di ; Point to next character
mov dl, [di] ; Load each character through [DI]
.ENDW

ret

PutStr ENDP

END

```

Setting the Main Language and Build Options

Before assembling the HELLO program, you must first specify the type of program you are building. Once you choose a main language and an initial build option, you can change its settings using the MASM Options command from the Options menu.

► To specify a main language and an initial build option:

1. Choose the Build Options command from the Options menu
2. Choose the Set Main Language button
3. Use the mouse or the ARROW keys to select "Assembler" from the list, and then press ENTER
4. Choose the Set Initial Build Options button
5. Use the mouse or the ARROW keys to select a build option from the list, and then press ENTER

For the HELLO program, select the DOS EXE build option.

Setting and Clearing the Program List

Once you set the main language and initial build options, you need to set the program list. The Set Program List command in the Make menu tells PWB which makefile to use in building a program. It clears the current makefile setting, instructs PWB to use the makefile you have specified, and specifies which Browser (.BSC) file is associated with the current project.

Choose the Set Program List command from the Make menu. In the dialog box that appears, type the main program's filename (HELLO), then press ENTER (see Figure 3.4). PWB automatically loads the .MAK file if it exists or asks you if you want to create it.

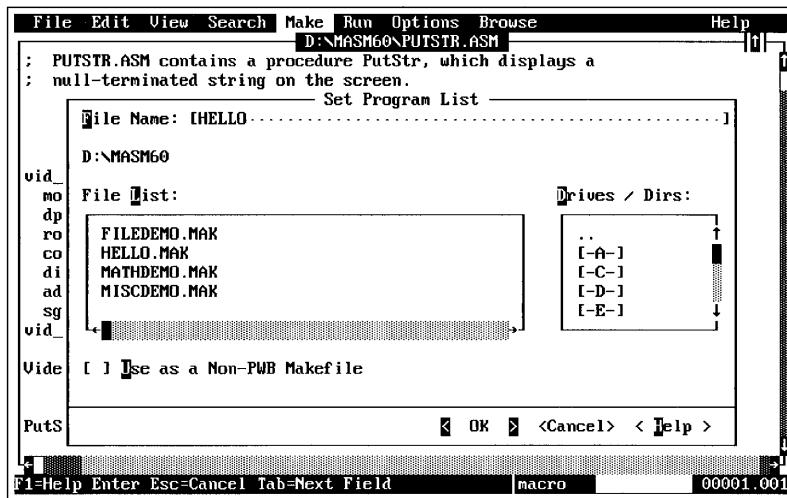


Figure 3.4 Dialog Box for the Set Program List Command

If the makefile does not exist, PWB asks if you want to create a new makefile; answer Yes. The Edit Program List dialog box appears, listing all files in the current directory. Select the files you want to include in the project (in this example, HELLO.ASM and PUTSTR.ASM). It doesn't matter in which order the files appear.

Keyboard ► To select the files:

1. Press TAB until the cursor is within the list box containing the directory of files (another box lists the directory names).
2. Use the ARROW keys to move to the HELLO.ASM file. Press ENTER to add the filename to the program list.
3. Repeat step 2 for PUTSTR.ASM.
4. Once both filenames appear in the program list, press TAB until the Save List command button is highlighted, and then press ENTER.

Mouse ► To select the files:

1. Click HELLO.ASM once; then click the Add/Delete button (or just double-click HELLO.ASM).
2. Repeat step 1 for PUTSTR.ASM.
3. Once both filenames appear in the program list, click the Save List command button to save the makefile.

PWB automatically creates a makefile appropriate for the files you selected. You can edit the .MAK text file to change or add additional commands.

If you want to create a new program or instruct PWB to “forget” the current program, use the Clear Program List command in the Make menu. This action clears the current program-list setting. You can then specify a new program to build using the Set Program List command. You should always clear the program list before starting work on a different project.

Assembling and Linking HELLO.ASM

The MASM Options command in the Options menu lets you control a variety of assembly options (see Figure 3.5).

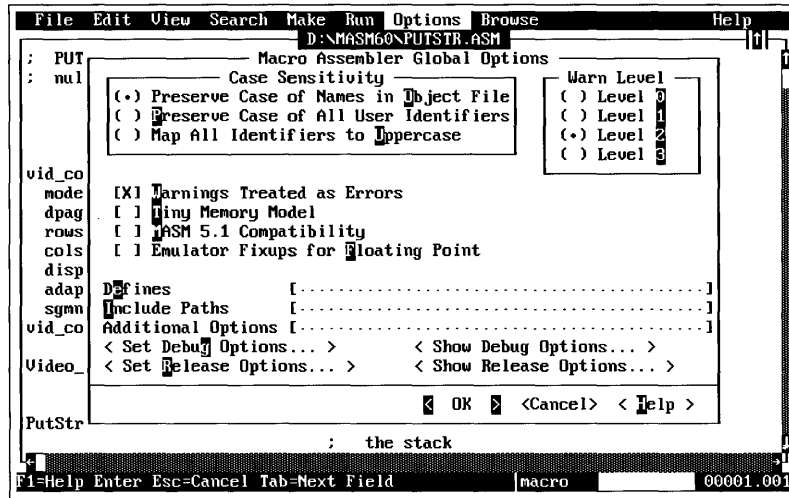


Figure 3.5 Macro Assembler Global Options Dialog Box

From within the Macro Assembler Global Options dialog box, you can use the Set Debug Options and Set Release Options buttons to easily create debug and release versions of your program.

The Set Debug Options button displays a dialog box where you can set parameters for activities like generating list files and debugging. These settings also apply to programs built with the Debug Build option, which you select from the dialog box displayed by the Build Options command. Use debug settings when you are developing and actively debugging your program. When the Release Build option is checked in the Build Options dialog box, the parameters specified in the Set Release Options dialog box apply.

Similar debug and release options are also available for the linker with the LINK Options command in the Options menu (see Figure 3.6). You can display the options in both the Macro Assembler Global Options and LINK Options dialog boxes using Show Debug Options and Show Release Options buttons.

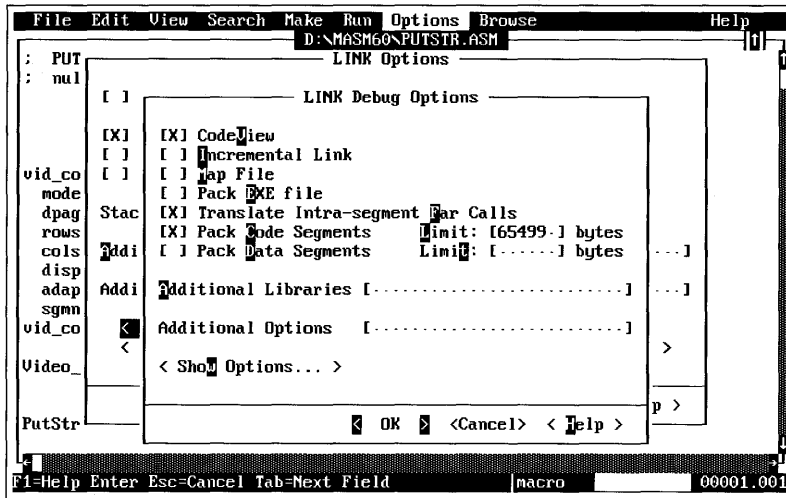


Figure 3.6 LINK Options Dialog Box

The Set Release Options button in the Macro Assembler Global Options dialog box offers the same choices as Set Debug Options, but Set Release applies them to a final version of the program. Use these settings when your program is completely debugged and ready for release.

Before building a program, make sure its name appears after the Compile File and Build commands in the Make menu. If the name does not appear after the Build command, choose the Set Program List command in the Make menu to set the makefile as the current project.

To build the program, choose the Build command in the Make menu. If an assembly or link error occurs, the build terminates and the error appears in the Compile Results window. You can get information about unknown errors from the online reference system by selecting the error and pressing F1. In OS/2, you are notified of errors and asked if you want to see the assembly log. (If you want to view the assembly log when running PWB under DOS, you must select the Compile Results command from the View menu before starting assembly.)

Running the Program

After the program is built, PWB adds the name of the executable file to the Execute command in the Run menu. Choose Execute to run the program. When the program is complete, press any key to return to PWB.

Debugging Programs

PWB has a variety of resources available for debugging programs. They range from simple assemble-and-link error messages to sophisticated source browsing and source- or assembly-level debugging.

When assembly errors occur, the cursor moves to the line that caused the first error; a brief description of the error appears in the reference bar. Use the Next Error and Previous Error commands in the Search menu to move the cursor to lines where other errors occurred.

To see a complete list of errors, choose the Compile Results command from the View menu. Move the cursor into the Compile Results window and use the Set Error command in the Search menu to select the error under the cursor as the current error. This command synchronizes the source and error windows so the source-code line containing the error appears in the active window.

Use the Microsoft Advisor to display online help about the error. In the Compile Results window, move the cursor to the error number and press F1, or click the number with either mouse key.

PWB incorporates a source-code browser and interfaces with the CodeView debugger. The Browser displays information about function and data relationships. CodeView is a window-oriented symbolic debugger that lets you step through your code and examine variables and data as the program executes.

Using the Browser

The Browser can search a database of selected files to get a list of references and definitions, to build a call tree and outline, or to create a list of relationships among program symbols. You can examine this material to see how your program is structured. Figure 3.7 shows how HELLO.ASM calls the PutStr procedure.

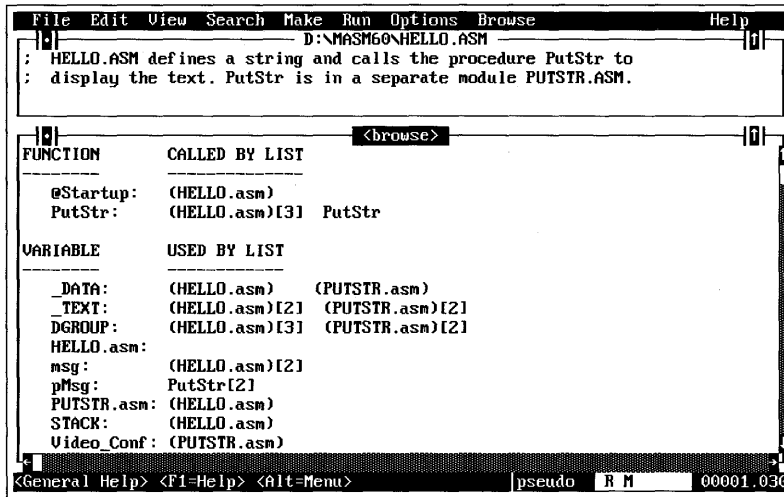


Figure 3.7 Browser Information

The commands on the Browse menu are referred to collectively as the PWB Source Browser. Before using these commands, you must define the database the Source Browser is going to search.

► **To define a database:**

1. Generate a program list using the Set Program List command from the Make menu. The files contained in the program list make up the database.
2. Choose Browse Options in the Options menu. You must select the Generate Browse Information field in the dialog box to create the Source Browser database. The rest of the fields in this dialog box offer you choices of how to build the database; they are optional.
3. Build your program using the commands from the Make menu. When you build the program, a file with the same program-list root name and a .BSC (Browser Source Cache) extension is created. This file contains reference information used by the Browser.

Once you have completed these steps, the following Browse menu commands are ready for you to use:

- The Goto Definition command provides a list of symbols and variables that occur within include files and modules. You can immediately open the file at the line number where the definition of a symbol or variable appears by selecting the OK button.
- The Goto Reference command displays a list of files and line numbers where symbols and variables have been referenced in include files and modules.
- The View Relationships command provides detailed information about various portions of your program. You can examine functions, macros, types, and variables by their use, definition, or inclusion in include files and modules.
- The List References command displays references to functions, macros, types, and variables. You can move the cursor to a reference and then use the Goto Definition or Goto Reference command to show occurrences of the reference.
- The Call Tree command displays a tree structure of function calls within the program.

For additional information about the Browser, refer to the online reference system.

NOTE You can create a Browser information file with the ML (command-line) assembler using the /FR or /Fr option. This creates a .SBR file, which you must convert to a Browser-readable .BSC file with the PWBRMAKE.EXE utility.

Debugging with the CodeView Debugger

Once a program has been built, you can use the Microsoft CodeView debugger to step through the source code, displaying output, registers, and variables as each line is executed (see Figure 3.8). You can also set breakpoints and watchpoints to cause execution to pause at critical sections of code. From within CodeView, the Microsoft Advisor can provide online help about CodeView operation, options, and MASM keywords in the source code. Although you can call CodeView from the command line, PWB provides access to CodeView without exiting. See Chapter 15 in the *Programmer's Guide* for more information about using CodeView.

NOTE CodeView version 3.12 is compatible with releases of OS/2 later than version 1.0. If you have problems exiting CodeView, you have either an older version of OS/2 or the wrong version of the DOSCALLS.DLL file.

If you use CodeView from the OS/2 DOS compatibility box, be sure the /S (Swap Screen in Buffers) option is set in the CodeView Options dialog box. If the option is not set, the mouse cursor is not displayed while you are debugging.


```

File Edit View Search Run Watch Options Calls Help
+EBP+00041 unsigned char pMsg = 0x61EF:0x000E "Hello, world.Jp"
-----local-----
source1 CS:IP PUTSTR.asm (ACTIVE)
14:      sgmnt      WORD      10
15:      vid_config ENDS
16:
17:      Video_Conf vid_config <0, 1, 25, 80, 3, 4, 10>
18:
19:      .CODE
20:      PutStr PROC  pMsg:PTR BYTE ; Address of string (near or far
21:                ; depending on model) is passed on
22:                ; the stack
23:                mov     ah, 02h ; Display character function
24:                mov     di, pMsg ; Load address in DI
25:                mov     dl, [di] ; Load each character through [DI]
26:
27:                .WHILE (dl) ; DOS displays character
28:                int     21h ; Point to next character
29:                inc     di ; Load each character through [DI]
30:                mov     dl, [di]
31:                .ENDW
<F8=Trace> <F10=Step> <F5=Go> <F6=Window> <F3=Display>

```

Figure 3.8 The CodeView Window

Preparing a Debug Build

It's easy to create a debug version of a program so it can be examined with CodeView.

Choose MASM Options in the Options menu; then select the Set Debug Options button. This dialog box contains a variety of settings for a debug build. Although most of the options can be set to your preference, you must check the CodeView option under the Debug Information heading.

The CodeView Options command in the PWB Options menu provides a variety of choices for configuring CodeView. These include two-monitor debugging, memory allocation, and selecting the number of lines on the screen. For now, use the default settings initially displayed.

For the assembler to build a program containing the necessary CodeView information, you must check the Debug option, which is set in the dialog box displayed by the Build Options command in the Options menu. You must also select the CodeView option (which is on by default) in the Set Debug Options dialog box accessed from the MASM Options command in the Options menu.

To demonstrate how CodeView works, set the debug options as listed above, then rebuild the HELLO.ASM program using the Rebuild All command in the Make menu.

Examining the Program

After your program has been built with the appropriate debug options, choose the Debug command from the Run menu. PWB immediately transfers control to CodeView, loading the current project as indicated in the program list.

When you switch from PWB to CodeView, the program appears in the Source window. The Source command in the Options menu displays your program in one of the following ways:

- As source code from your MASM source-code file. This is the default display.
- As assembly language generated by CodeView disassembling the program's executable file.
- As a mix of these two, with the disassembled code following the corresponding source code.

The mixed-code option is especially useful when your program contains macros or directives that generate code; you can see how the assembler converted them into machine instructions.

In addition to showing source code, commands in the View menu show windows displaying memory, registers, and local variables. All windows are similar to PWB windows; they can be scrolled, sized, and closed. You can have multiple windows on one screen, displaying a variety of information. A list of shortcut keys and buttons appears at the bottom of the screen.

Setting Watch Expressions and Breakpoints

A “watch expression” is a program variable that CodeView continually monitors. Watch expressions are displayed using the Watch command from the View menu. As the value of a variable changes, the value also changes in the watch window.

To add a variable to the watch expression list, position the cursor anywhere within the variable and choose the Add Watch command from the Watch menu. You can also type the variable name in the dialog box. Remove watch expressions using the Delete Watch command from the Watch menu, or position the cursor on that expression's line in the Watch window and press CTRL+Y.

To set a watch variable in the HELLO.ASM file, use the cursor to highlight the `pMsg` variable in the CodeView Source window. Then choose the Add Watch command from the Watch menu. The `pMsg` variable appears in the Expression text box. Press `END` to position the cursor at the end of the name, and then type `,s` (so that CodeView displays the data at this address as a string). Press `ENTER` or click the OK button to add this variable to the Watch window. The Watch window gives a "Watch Expression Not in Context" error until you trace into the `PutStr` procedure.

CodeView also incorporates a temporary watch option called Quick Watch. Highlight a variable in the Source window with the mouse or cursor. Choose the Quick Watch command from the Watch menu. A dialog box appears giving the current value of the variable. You can add any Quick Watch variable to the watch window using the Add Watch dialog box option.

You can expand any structures or arrays that are set as watch expressions (identified with a `+` in the Watch window) to show individual elements. Position the cursor on the variable name and either double-click the left mouse button or press `ENTER` (see Figure 3.9).

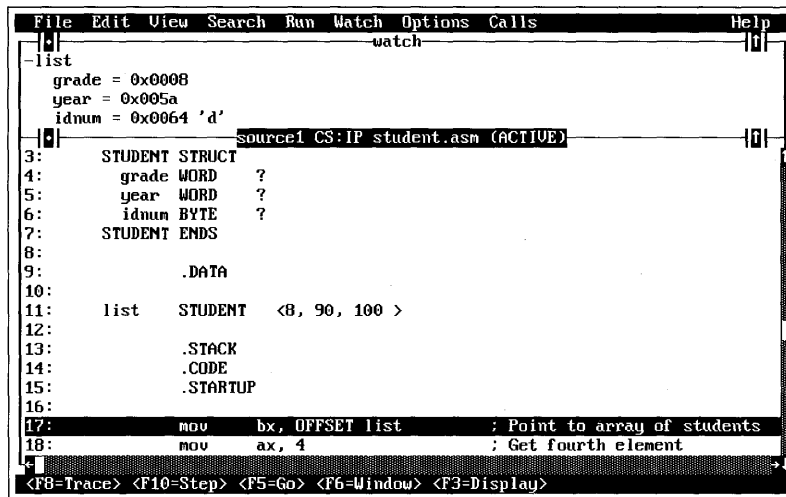


Figure 3.9 The Watch Window

A "breakpoint" is an instruction that tells CodeView to interrupt execution at a certain location or when an expression becomes true.

Use breakpoints to cause execution of the program to pause before an error occurs and to move quickly to a part of the program you want to trace. When CodeView stops at a breakpoint, you can single-step or trace through instructions until you find the problem.

Set a breakpoint by positioning the cursor on the line at which you want the program to stop and then pressing F9. You can also choose the Set Breakpoint command from the Watch menu. A dialog box presents a series of options for setting types of breakpoints. You can modify or delete breakpoints using the Edit Breakpoints command from the Watch menu.

Set a breakpoint in the HELLO.ASM file by moving the cursor in the CodeView Source window to the line that reads

```
INVOKE  PutStr, ADDR msg
```

From the Watch menu, choose the Set Breakpoint command. The number of the line the cursor appears on is displayed in the dialog box. Press ENTER or click the OK button to store the breakpoint. After the breakpoint has been set on a line, that line is highlighted.

Running the Program Within CodeView

CodeView offers several ways to run a program. You can let the program execute at full speed or view each line as it executes. Table 3.2 lists the CodeView run commands.

Table 3.2 CodeView Run Commands

Name	Use	Key
Go	Executes the program to the next breakpoint, error, or end of the program	F5
Trace	Executes a single line	F8
Step	Executes a single line, but doesn't trace into functions	F10

Enter run commands by pressing the appropriate key or clicking the commands at the bottom of the screen. The Trace and Step commands highlight each line as it executes. The Go command simply executes (without highlighting) until the program stops at an error, a breakpoint, or the end. If you choose the Animate option for the Go command, CodeView highlights each line as it executes. To force execution to start at the beginning of the program, use the Restart command in the Run menu.

You can either single-step through a program or use the Animate command in the Run menu to step through the program automatically, highlighting each line as it proceeds.

If a program requires arguments to be passed from the command line, these arguments can be entered with the Set Runtime Arguments command from the Run menu.

Running HELLO.EXE

To look at some of the CodeView commands while running HELLO.EXE, make the Source window active by pressing F6 or by clicking its title bar. Then choose the Restart command from the Run menu. Restart clears memory and prepares the program to start execution at .STARTUP.

Press F5 or choose the Go button at the bottom of the screen to start the program. The program executes until it reaches the breakpoint you set at the INVOKE directive.

If you press F8 or the Trace button, CodeView executes and displays each statement of `PutStr`. If you use F10 or the Step button, CodeView executes `PutStr` without showing each step.

The variable `pMsg` is actually a pointer, so no value is displayed until you execute the `PutStr` procedure. As you step through `PutStr`, the display switches to the output screen each time a character is written to the screen.

Because `pMsg` is also a local variable, it appears in the Local window with all variables local to `PutStr`. You can expand local data in the Local and Watch windows by double-clicking it.

Press F10 to finish stepping through the program. A dialog box informs you that the program has completed execution normally. Press ENTER to clear the dialog box and complete your debugging session.

Debugging HELLO.EXE

To see how CodeView deals with a bug, add these statements at the end of the source code for this program:

```
mov ax, 0FFh
int 21h
```

Set the appropriate debug options, rebuild the program, and return to CodeView.

Use F8 (Trace command) to execute the program. An execution error occurs when the INT instruction is executed, because there is no interrupt function corresponding to 0FFh.

Getting More Details

PWB, the assembler, the linker, NMAKE, and CodeView have additional features and options beyond the scope of this introduction.

Refer to the *Programmer's Guide* for information about PWB, NMAKE, CodeView, and other tools. Refer to the *Reference* for a quick reference to all commands and options for the applications included with MASM.

You can access additional information about any component of the Professional Development System through the Microsoft Advisor online reference system. For a description of the online reference system, see Chapter 4 of this manual.

SUCCESS!

EXIT PROGRAMMER'S
WORKBENCH

YES

NO

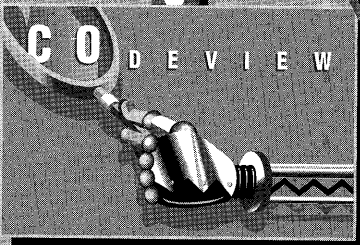
RUN SUCCESS?

OS/2

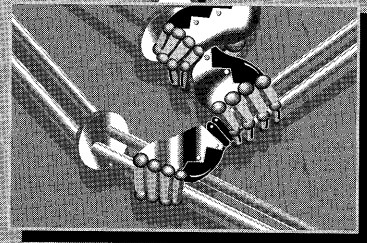
DOS

CALL
You called a procedure that was
never properly debugged.
Jump to CodeView

PICK AN ENVIRONMENT



DEBUG



LINK

Chapter 4

Using the Online Reference System

The Microsoft Macro Assembler (MASM) offers two types of online reference systems:

- The Microsoft Advisor, accessible from within the Programmer's WorkBench (PWB) and CodeView environments
- The QuickHelp program, accessed from the OS/2 or DOS command line

Both reference systems provide the same information about important features of the Professional Development System, including language-specific information and information about utilities like the CodeView debugger and PWB.

The first section of this chapter, "Structure of the Microsoft Advisor," outlines the structure and contents of the online reference database. The second section, "Navigating the Microsoft Advisor," takes you on a quick tour of the system. The third section, "Using QuickHelp," explains how to use the QuickHelp program and how it differs from the Microsoft Advisor.

Structure of the Microsoft Advisor

The Microsoft Advisor displays a table of contents of all online reference files available with the Professional Development System. Choose the Contents command from the Help menu to display the table of contents. From the table of contents you can select any topic you want to know more about. Figure 4.1 shows the table of contents screen.

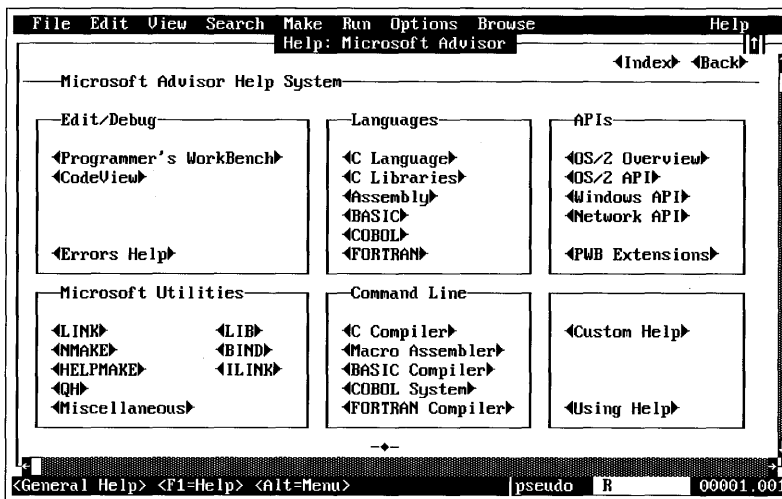


Figure 4.1 Table of Contents of the Online Reference

The Microsoft Advisor also displays indexes for each online reference file. Choose the Index command from the Help menu to display the help topics, and then select the topic whose index you want to search. The alphabet bar at the top of the index screen allows direct access to a list of the index topics starting with the letter you select.

Navigating through the Microsoft Advisor

You request information about a topic in a window by positioning the cursor over it and pressing F1 or clicking the right mouse button once. The help system then searches through the help files for information about the topic. If it finds the topic, the help system displays information in the Help window. If help cannot be found for a particular word or symbol, a message informs you that no information is associated with the topic. Sometimes a topic with the same name occurs in several help files. In such cases, a dialog box is displayed in which you can select the context of the topic.

Using the Help Menu

The Help menu provides full access to the online reference system. All these commands display information in the Help window:

<u>Command</u>	<u>Description</u>
Contents	Displays a global contents screen of the topics in the help files.
Index	Displays a global index list and provides access to indexes for all components of the online reference system. (See Figure 4.2.)
Topic:	Provides information about the topic the cursor is currently on. If information about the topic is available, the topic's name is appended to the Topic: command. Otherwise, this menu option is grayed.
Help on Help	Displays information about using the online reference system itself.
Next	Locates the next occurrence of a topic name and displays the information associated with it. Use this command to get additional information about a topic that appears in more than one help file.

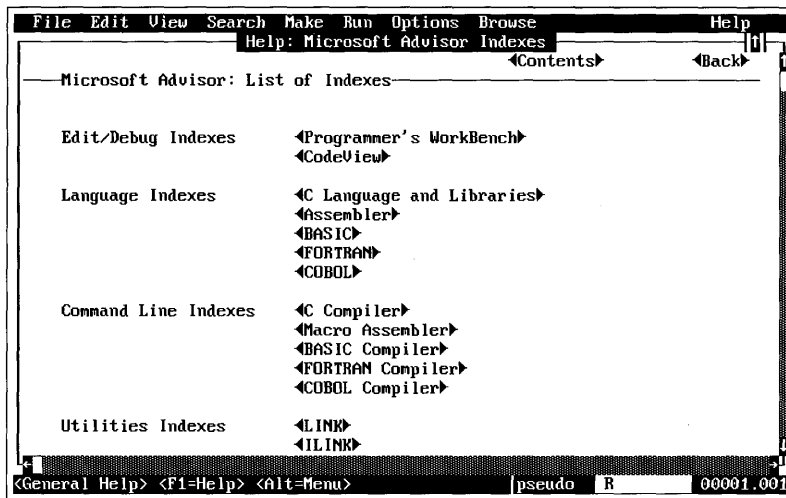


Figure 4.2 Global Index of the Online Reference

Using the F1 Key and the Mouse

The F1 key displays information about the MASM keyword or operator the cursor is positioned on. You can also use the F1 key to directly access information about

the menu command currently highlighted or the PWB dialog box currently displayed.

The mouse can perform the same function. Position the mouse cursor on the command, dialog box item, or keyword, and click the right mouse button.

If the help system cannot find help for a particular word or symbol, a message tells you so. Sometimes a topic with the same name occurs in several help files. In such cases, a dialog box appears in which you can select the context of the topic.

Figure 4.3 shows the help information for the Cut command from the Edit menu.

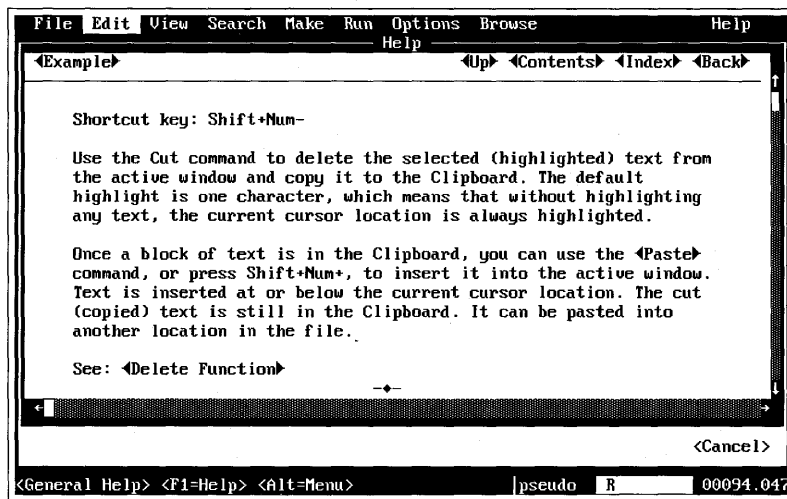


Figure 4.3 Help on the Cut Command

Using Hyperlinks

Hyperlinks are cross-references that connect related topic information. Hyperlinks can be marked explicitly, or they can be implicit.

Hyperlinks marked with the ◀ and ▶ characters are called "buttons." You can move through the online reference system using these buttons. Press TAB to move the cursor to the next hyperlink button within the current Help window. SHIFT+TAB moves the cursor back to the previous button. Typing any letter moves the cursor to the next button that begins with the letter. Holding SHIFT down and typing a letter moves the cursor backward.

The Microsoft Advisor also recognizes MASM keywords, constants defined by MASM, and similar identifiers as hyperlinks, but they are unmarked. These unmarked hyperlinks are recognized wherever they appear in the help text or in your source code. You cannot move to an unmarked hyperlink with the TAB key.

You can access a hyperlink from the keyboard or with the mouse by doing one of the following:

- Move the text cursor to the hyperlink and press F1
- Position the mouse cursor over the hyperlink; then click the right mouse button or double-click the left mouse button

You can also execute highlighted button hyperlinks by pressing ENTER or the SPACEBAR.

Any of these actions displays information about the topic the cursor is positioned on. If the topic isn't a hyperlink, a message informs you that no information on the topic could be found.

NOTE CodeView uses the right mouse button differently in the Source window. Clicking the right button in the Source window executes the program to the line the mouse was clicked on. However, once the Help window is displayed, clicking the right mouse button selects hyperlinks.

Using Help Windows and Dialog Boxes

The Microsoft Advisor displays information in windows or dialog boxes. Help windows and dialog boxes function the same as other windows and dialog boxes found in PWB and CodeView. For a description of windows and dialog boxes, refer to Chapter 3, "Using the Programmer's WorkBench."

Using the Help Window

The Help window displays tables of contents, indexes, and information about selected topics. Some screens of information are larger than the Help window; you can view information beyond the window borders with the scroll bars or the cursor-movement keys. The **—◆—** symbol marks the end of information in the Help window.

Navigating with Hyperlink Buttons At the top of the Help window is a row of buttons that is useful for moving through the reference system:

<u>Button</u>	<u>Description</u>
◀Up▶	Moves upward in the hierarchy of help screens. Since information is ordered from the general to the specific, this button takes you to more general information about the subject.
◀Contents▶	Displays the table of contents. This button always returns you to a known point in the online reference system.
◀Index▶	Displays the index list. Selecting an item from the list displays the index for that category.
◀Back▶	Moves to the last online reference screen you viewed.

Frequently, screens about a particular topic are grouped together in a help file. Press CTRL+F1 to display information about the next topic in the help file.

Viewing the Previous Help Screen The Microsoft Advisor remembers the last 20 help screens you've accessed. To return to a previous screen, use the ◀Back▶ button, or hold down the ALT key and press F1 as many times as necessary to return to the screen you want to see. The help screen that appears is active; you can ask for help on any of its hyperlinks or topics.

You can always return to the global contents by pressing SHIFT+F1.

Copying and Pasting Information You can easily copy any text that appears in the Help window to another window. For example, to test an example program in the Help window, just copy it to the Source window and assemble it.

Keyboard ▶ **To copy and paste, follow these steps:**

1. Move the cursor to the beginning of the text you want to select. Hold down the SHIFT key and move the cursor to the end of the desired text. The text is now highlighted.
2. Execute the Copy command: press ALT to activate the menus, E (Edit), then C (Copy). Note that the menu lists the shortcut command (CTRL+INS), which you can substitute for ALT+E, then C.
3. Press F6 to switch to another window. Position the cursor where you want to insert the text and execute the Paste command: ALT+E, then P; or use the SHIFT+INS shortcut. The text from the Help window is inserted at the current cursor position.

Mouse ► **To copy and paste, follow these steps:**

1. Click the left mouse button and drag the cursor (hold down the button while moving the mouse) to select the text you want to copy.
2. Choose Copy from the Edit menu (or press CTRL+INS).
3. Move the mouse cursor to the location where you want to insert the text, and click once. Then choose Paste from the Edit menu. The text from the Help window is inserted at the current cursor position.

Closing the Help Window To close the active Help window and return to the Source window, do one of the following:

- Press ESC
- Click the Close button in the upper-left corner of the window with the left mouse button

Using Help Dialog Boxes

Help dialog boxes contain information about menu commands and their dialog boxes. A help dialog box differs from a Help window in that it is displayed over the Source window and any other open windows. You must close a help dialog box before you can execute menu commands. Clicking the Cancel button in the lower-right portion of the dialog box closes the help dialog box.

To view information about any PWB dialog box, do one of the following:

- Press F1
- Click the help button in the dialog box with the left mouse button

Either of these commands causes a help dialog box to appear. To close a help dialog box, do one of the following:

- Press ESC
- Click the Cancel button in the lower-right portion of the dialog box with the left mouse button

Accessing Different Types of Information

This section presents some ways to access the different types of information available within the online reference system.

Keyword Information

To get information about any keyword, operator, or symbolic constant, first display the appropriate index for the keyword. Then select the keyword's first letter from the index, scroll down the list of entries, and select the topic hyperlink. If you know the exact name of a keyword, it may be quicker to type it in the Source window and press F1.

You can also view information about keywords using the **arg** command. Type the keyword in the **arg** command dialog box. Then press F1. If you have the **arg** function assigned to `alt+a`, the following displays help about the ADC (add with carry) command:

```
alt+a ADC F1
```

Figure 4.4 shows the information screen that then appears.

When information about a keyword is shown in the Help window, additional hyperlink buttons may be displayed:

- ◀Detail▶ provides a detailed explanation of the keyword. When the information is displayed, the button changes to ▶Summary◀. Click this button to return to the summarized information about the keyword.
- ▶Example▶ displays MASM source code that provides an example of how the keyword is used.
- ▶Key▶ explains symbols used in instruction encodings.

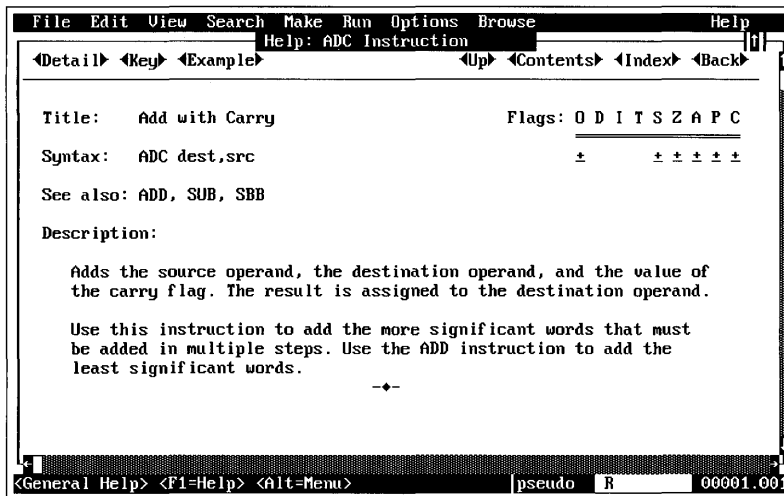


Figure 4.4 Help on the ADC Command

Topical Information

The online reference system is useful when you want an overview of available reference topics or when you have only a general idea of what information you need. Start with the global contents screen and select any hyperlinks that relate to the topic. Then move through the hyperlinks until you locate the information you need.

Menu and Dialog-Box Information

You can get information about any menu command or dialog box by pressing F1 when the menu command is highlighted or when the dialog box is displayed.

Error Information

The Microsoft Advisor provides information about assembler and linker error messages.

Keyboard ► **To find the meaning of an error message:**

1. Press F6 to move the cursor to the error window
2. Position the cursor on the error number and press F1

Mouse ► **To find the meaning of an error message:**

1. Position the mouse cursor on the error number
2. Click the right mouse button

To get help about error messages directly from the PWB Source window, type the **arg** command with the error number and its alphabetic prefix, and then press F1.

Specifying Temporary Help Files

You can instruct the Microsoft Advisor to search help files that you specify before searching the help files specified in the HELPFILES environment variable. For example, you may want to access the C help files first when writing a mixed-language program.

Select the Environment command in the PWB Options menu. Type the path for the help files you want searched first in the text box that appears. These files are searched first only during your current session.

Using QuickHelp

QuickHelp is a separate utility that works like the Microsoft Advisor. It displays information in the same way and can access any Microsoft Advisor help file. Use QuickHelp if you use an editor other than PWB, or if you prefer to use command-line utilities. QuickHelp also supplies convenient access to help when you aren't running PWB.

The following sections explain how to call QuickHelp from the command line. You can use these same command-line calls to access QuickHelp from within any program that lets you pass commands to DOS or OS/2.

Entering and Exiting QuickHelp

If you installed documentation files during SETUP, you can run QuickHelp by typing QH at the DOS or OS/2 command line. You can also specify the name of an application, such as

```
QH LIB.EXE
```

To get QuickHelp about the **mov** command, type

```
QH mov
```

If your editor supports macros, you can write a macro that reads the desired help topic from the editor's command line or from the text-entry window, then appends it to QH and passes the result to the operating system.

QuickHelp can also display information about run-time errors. Type the error number with its alphabetic prefix after the QH command at the DOS or OS/2 command line.

To exit QuickHelp, choose the Exit command from the File menu.

Using /HELP

You can get immediate information about major components of MASM by using /HELP.

For example, to learn about the LIB utility, type

```
LIB /HELP
```

QuickHelp immediately displays information about LIB.

Specifying Help Files

QuickHelp searches for .HLP files according to a few simple rules. Understanding these rules helps you quickly access only the .HLP files you want. The more specific the search, the faster QuickHelp starts.

You tell QuickHelp which .HLP files to open by specifying files or directories on the command line or in environment variables. To save time, most programmers specify commonly used .HLP files in the HELPFILES or QH environment variables.

You can specify either an entire directory or an individual file. If you specify a directory, QuickHelp opens every .HLP file in that directory.

QuickHelp searches for .HLP files in the following order:

1. If you specify a .HLP file or directory on the command line, QuickHelp first searches that file or directory.
2. QuickHelp next searches for files and directories specified in the HELPPFILES and QH environment variables.
3. If you do not specify any .HLP files on the command line, and QuickHelp cannot find any .HLP files in the HELPPFILES and QH directories, QuickHelp searches the following directories and environment variables in the order shown:

```
c:\qh
DPATH
PATH
BOOKSHELF (OS/2 only)
```

Note that QuickHelp does not search the current directory unless you specify that directory in one of the environment search paths.

The List Database command from the File menu displays a list of currently open help files. To open additional help files

- Choose the Open Database command from the File menu.
- Type in the fully qualified pathname of the help file to be opened in the dialog box that appears. You can specify all help files in a directory by typing *.HLP.
- Press ENTER or click the OK button with the left mouse button.

To close an open help file

- Choose the Close Database command from the File menu.
- Use the mouse or keyboard to select the help file you want to close from the list of help files displayed in the menu.

Displaying a Topic

You can view information about a topic by using the Search command from the View menu. When topic information is displayed, it is shown in the same format as information presented by the Microsoft Advisor.

► **To display a topic from any of the open help files:**

1. Choose the Search command from the View menu.
2. Type in the topic you want information on in the Search dialog box (the search is not case-sensitive).
3. Press ENTER or click the OK button with the left mouse button.

QuickHelp begins searching for the topic in the open help files. If the topic cannot be found, a dialog box informs you that the search failed. If the search is successful, information about the topic is displayed in the QuickHelp window.

Navigating through the Topics

A series of commands in the View menu allows you to display topics selectively. These commands include the following:

<u>Command</u>	<u>Description</u>
View History	Displays a list of all of the topics that have recently been displayed. To view a topic in the list, select it and click the right mouse button, or press ENTER.
View Last	Displays the last topic you looked at.
View Back	Moves backward one topic in the help file.
View Next	Displays the next topic in the help file.

Using the QuickHelp Window

The QuickHelp window (see Figure 4.5) is similar in function to the Microsoft Advisor Help window, described earlier in this chapter. Information that doesn't fully fit in a window can be scrolled, and hyperlinks are used to display additional information.

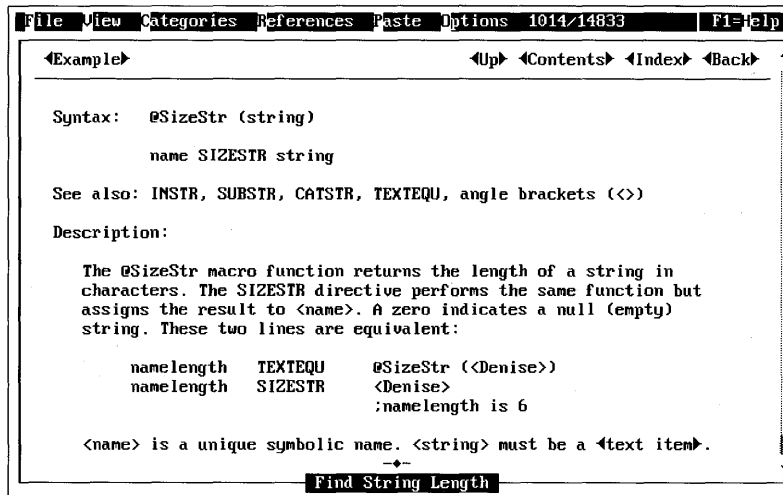


Figure 4.5 QuickHelp Window

Copying and Pasting in QuickHelp

In order to transfer information, you must first specify a new file with the Rename Paste File command from the File menu. Once the file is specified, choose either the Current Window or the Current Topic command from the Paste menu to transfer the text to that file. Be sure to specify a new file when you paste, since QuickHelp overwrites text in an existing file.

More About QuickHelp

In addition to the features described above, QuickHelp has a variety of other options, such as changing the appearance of the Help window, searching for text within topics, and controlling the function of the right mouse button.

To learn more about QuickHelp features, refer to the *Reference* for a list of QuickHelp options. For online information, make sure the QH.HLP file is loaded, then do one of the following:

- Press F1 to view QuickHelp's online reference
- Choose a menu command and press F1 or click the right mouse button to get information about the command

Index

A

- Anchors in text files, 37
- ASMEX environment variable, 17
- Assembling and linking
 - Browser command-line options, 51
 - build options, 44
 - building programs, 41–42
 - debug build, 52
 - debug options, 47–48
 - program list
 - clearing, 45
 - setting, 45
 - release options, 47–48
 - running programs, 46
- AUTOEXEC.BAT, modifying, 17

B

- Bookmarks in text files, 37
- Bound software, 15
- Breakpoints, setting, 54
- BRIEF.INI, 41
- Browse menu, 50
- Browser. *See* Source Browser
- Buffers parameter in CONFIG.SYS, 18
- Build options, setting, 44
- Buttons
 - command, 34
 - hyperlink, 62
 - option, 33

C

- Check boxes, 33
- CodeView
 - breakpoints, setting, 54
 - configuring extended memory, 20
 - debug build, 52
 - DOS compatibility box, 51
 - examining programs in, 53
 - executing programs in, 55
 - generating information for, 51

- CodeView (*continued*)
 - mouse buttons, use of, 63
 - OS/2, 51
 - Quick Watch option, 54
 - run commands, 55
 - Source window, 53
 - using, 51, 56
 - watch expressions, 53
- Command buttons, 34
- Compiling Results command, 49
- CONFIG.SYS, modifying, 18
- Configuration files, 17

D

- Debug options, 47
- Debugging, 49, 51–56
- Dialog boxes
 - described, 32
 - online reference, 65
- Directories
 - bound-mode, 15
 - protected-mode, 15
 - real-mode, 15

E

- Editor, PWB
 - advanced features, 40
 - anchors, setting, 37
 - blocks, defining, 36
 - bookmarks, setting, 37
 - changing settings, 39
 - cursor, moving in, 35–36
 - customizing, 39, 41
 - key assignments, changing, 39
 - macros, creating, 38
 - reconfiguring, 41
 - saving files, 42
 - searching and replacing, 37
 - text-selection modes, 36
 - writing and saving procedures, 43
- Environment variables, 16–17
- Errors, getting help on, 67

F

F1 key, 61, 67
Features, new, 3
Files parameter in CONFIG.SYS, 18

H

Hard disk, choosing, 15
Help. *See* Online reference system
Help files, specifying, 69
Help information
 copying and pasting in QuickHelp, 72
 copying and pasting with Microsoft Advisor, 64
Help menu
 commands, 60
 using, 35, 60
HELPPFILES environment variable, 17, 68
HIMEM.SYS, 18, 20
Hyperlinks, 62–63

I

INCLUDE environment variable, 17
INIT environment variable, 17
Installing MASM
 See also SETUP.EXE
 mouse driver, copying, 15
 operating system, choosing, 13
 Programmer's WorkBench, 14

K

Keyboard, changing assignments, 39
Keys, shortcut, 32

L

LIBPATH environment variable, 18
Linking, 41
List boxes, 33

M

Macros, creating, 38
Make menu, 48, 50
Makefile, 41
MASM.EXE, installing, 14

Memory

 extended, configuring, 20–21
 MASM, required for, 2

Menus

 Browse, 50
 choosing commands, 31
 closing, 32
 Help, 60
 Make, 46, 48, 50
 Options, 44, 47, 50, 52
 shortcut keys, 32
 View, 49, 53
 Watch, 53, 55
Microsoft Advisor. *See* Online reference system
ML.EXE, 14, 51
Mouse support, 15
MSHELP.DLL, 18

N

New features, 3
NEW-CONF.SYS, 17
NEW-VARS.BAT, 17
NEW-VARS.CMD, 17
Next Error command, 49

O

Online reference system
 See also QuickHelp
 accessing, 60
 copying and pasting text, 64
 dialog boxes, 34, 65, 67
 error information, 67
 Help menu, 34
 Help window, closing, 65
 hyperlinks
 Back button, 64
 Contents, 64
 defined, 62
 Index button, 60, 64
 keyword information, 66
 Up button, 64
 keyword information, 66
 menu information, 34, 60, 67
 previous help screen, viewing, 64
 temporary file, specifying, 68
 topical information, 67
 using mouse in, 61

Operating systems

choosing, 13

DOS

defined, 1

selecting during SETUP, 13

setting up directories in, 15

OS/2

defined, 1

selecting during SETUP, 13

setting up directories in, 15

Option buttons, 33

Options menu, 44, 47, 50, 52

P

PATH environment variable, 17

Previous Error command, 49

Procedures, writing and saving, 43

Program list

clearing, 45

setting, 45

Programmer's WorkBench (PWB)

assembling programs in, 41

Browser, accessing from, 49

building programs in, 41–42

customizing, 19, 39, 41

debugging in, 49

described, 4, 6–7

Editor settings, 19, 39

features, 23

help, getting, 34, 63

installing, 14

macros, writing, 38

menu commands, choosing, 31

menus, 26, 31

options, setting, 19

screen, illustrated, 25

source files, specifying, 24

startup options, 24

text, selecting, 36

using, 23

Protected-mode directories, 15

PWB. *See* Programmer's WorkBenchPWB Windows. *See* Windows**Q**

QH environment variable, 17

QuickHelp

See also Online reference system

copying and pasting, 72

described, 59

QuickHelp (*continued*)

entering and exiting, 70

error information, 69

/HELP, 69

.HLP files, specifying, 69

navigating through topics, 71

topics, searching for, 71

using, 68

window, 72

R

RAMDRIVE program, 21

Real-mode directories, 15

Release options, 47

Run menu, 41

S

Saving files, 42

Searching and replacing, 37

Set Error command, 49

SETUP.EXE

changing your response, 13, 16

described, 11

file compression, 11

location on disks, 11

prompts, responding to, 13

running, 11

system-configuration files, 17

Shortcut keys, 32

SMARTDRV program, 21

Source Browser

commands, 51

defining databases, 50

generating information for, 50–51

PWBRMAKE.EXE, 51

using, 49

STARTUP.CMD, modifying, 17

System configuration, 17

System requirements, 2

T

TEMP environment variable, 17

Text boxes, 33

Text selection, 36

TOOLS.INI

macros saved in, 39–40

modifying, 19, 40–41

TOOLS.PRE, 17–18

V

View menu, 49, 53

W

Watch menu, 53, 55

Windows, PWB

 components, 25

 described, 24

 Help

 closing, 65

 using, 63

 QuickHelp, using, 72

Writing and saving procedures, 43

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Microsoft[®]
Making it all make sense[™]

0291 Part No. 06558