Microsoft® BASIC Compiler Version 6.0

For Personal Computers Running the MS® OS/2 or MS-DOS® Operating System

It's here—a BASIC compiler that lets you develop programs for MS OS/2 as well as MS-DOS! Plus you get the exceptional debugging power of Microsoft CodeView®, the revolutionary window-oriented source-level debugger. Microsoft BASIC compiler 6.0: it's the most advanced BASIC compiler available for the professional programmer.

Product Overview

Microsoft BASIC compiler version 6.0 supports the MS OS/2 and MS-DOS operating systems, letting you create both protected-mode and real-mode programs. This broad operating-system support, together with the debugging power of Microsoft CodeView, significant new language enhancements, and a host of new utilities, makes Microsoft BASIC compiler the most advanced BASIC compiler available for the personal computer environment—and the total BASIC solution for the professional.

Take advantage of protected-mode features.

Under MS OS/2, there's no longer a 640K limit to addressable memory, so your programs can use up to 16 MB of real memory or 128 MB of virtual memory! Microsoft BASIC compiler version 6.0 also supports other MS OS/2-specific capabilities, including multitasking and direct calls to the operating system.

Debug faster with Microsoft CodeView.

Microsoft BASIC compiler version 6.0 includes Microsoft CodeView, a powerful window-oriented source-level debugger. You can use Microsoft CodeView under both MS OS/2 and MS-DOS, debugging programs as large as 128 MB under MS OS/2. And its support for expanded memory lets you debug large MS-DOS programs, too.

Multiple windows show you everything that's going on—both inside the program and inside the CPU—while the program is running. And Microsoft CodeView features a visual interface, plain-language commands, and on-line help that make it easy to learn and use.

New power for professional programmers.

Microsoft BASIC compiler version 6.0 includes a host of new features designed specifically for the professional programmer. For example, you now have more control over which portions of the run-time library are linked to your program, so you can create smaller standalone executables. And if you use the separate run-time module, you can customize it by adding your own routines—whether you're working in protected mode or real mode.

There's also a new alternate fast math library for programs that don't use a math coprocessor, so your programs will run faster.

And if you're interested in writing instrumentation and manufacturing control programs, you'll appreciate the compiler's ability to trap and respond to hardware- or software-based user-defined interrupts.

We've also added advanced language features such as user-defined types, recursion, and huge arrays. And new structured program constructs shorten your development and maintenance time and simplify program logic.

Support for interlanguage calling lets your BASIC programs include modules written in several Microsoft languages, including C 5.1, QuickC™, FORTRAN 4.1, Macro Assembler, and Pascal 4.0.

Finally, you get Microsoft QuickBASIC version 4.0 with Microsoft BASIC compiler 6.0, providing an instant development environment for your MS-DOS programs.

Microsoft®

New support for MS OS/2.

Microsoft BASIC compiler version 6.0 gives you all the tools you need to create protected-mode programs for the MS OS/2 environment.

A flexible compiler, Microsoft BASIC compiler version 6.0 lets you choose whether you want to compile and link your program to run in protected mode under MS OS/2 or in real mode under MS-DOS.

Among the special MS OS/2 features supported by Microsoft BASIC compiler 6.0 are the SHELL function and the OPEN PIPE: statement, which let your program communicate asynchronously with other programs. Using SHELL, your program can start a process running and then continue executing, so time-consuming tasks such as printing won't slow it down. And OPEN PIPE: lets it transfer information to and from other MS OS/2 programs during execution. Other new MS OS/2 features are ON SIGNAL and SLEEP.

Powerful debugging.

Microsoft BASIC compiler now comes with Microsoft CodeView, the powerful window-oriented source-level debugger. Microsoft CodeView gives you total control over your program's execution, significantly shortening the debugging process. And now you can use CodeView for both protected- and real-mode programs.

Watch your code execute.

The window-oriented visual interface of Microsoft CodeView helps you catch problems quickly by letting you view source code, disassembly code, variables, registers, and stack—all at the same time. You can set and watch expressions and variables, animate or single-step through the program's execution, trace the contents of the stack, and set conditional breakpoints to stop execution whenever a variable or expression reaches a critical value. The screen even updates automatically as the program executes, so you always see the current values in all the windows.

You're in control.

With Microsoft CodeView, you can debug in your original source code or the resulting assembly language—or both at once. You can view assembly code with no symbols or full symbols, use language-specific expressions, and change any symbol by name, including common, shared, and local variables. For graphics-oriented applications, there's even a screen-swapping function that lets you switch back and forth easily between your code and the resulting output.

Debug in mixed languages.

Microsoft CodeView lets you debug mixed-language programs. In addition to BASIC, CodeView supports Microsoft C, QuickC, FORTRAN, Macro Assembler, and Pascal.

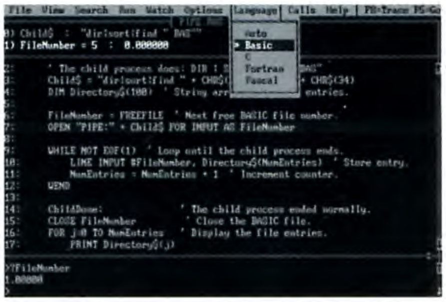
Choice of three math libraries.

With Microsoft BASIC compiler version 6.0, you have a choice of three floating-point math libraries. The first, an 8087/80287/80387 library, offers the fastest execution as well as high precision and requires the presence of a math coprocessor. The second, a floating-point emulator, provides the same 80-bit precision without a coprocessor. Or if a coprocessor is available at run time, the emulator switches to the coprocessor library instead. The third choice, an alternate math package, offers added speed with 64-bit precision for machines without a coprocessor.

New language enhancements.

We've significantly extended the capabilities of the BASIC language with a broad range of enhancements. These include:

- *New! Improved error handling* that lets you use a common error-handling routine for multiple-module programs.
- *New! User-defined events* that let you write applications for test instrumentation and industrial control.
- *User-defined data types and records* that let you organize data for easy access, simplifying file I/O.
- *Recursion,* so your subprograms and FUNCTION procedures can call themselves—a feature that's useful for tasks such as sorting.
- *Arrays as large as available memory* (up to 16 MB of RAM or 128 MB of virtual memory under MS OS/2).
- *Long integers* that eliminate rounding errors and increase speed in calculations involving very large whole numbers.
- *Fixed-length strings* that can be used with huge arrays to manipulate string data as large as available memory or to hold string data in records.


File View Search Run Watch Options Language Calls Help | F8=Trace F5=Go
PIPE.BAS
Auto
▸ Basic
C
Fortran
Pascal

```
0) Child$  :  "dir|sort|find " BAS""
1) FileNumber = 5  :  0.000000

      ' The child process does: DIR | S                  BAS"
      Child$ = "dir|sort|find " + CHR$(                + CHR$(34)
      DIM Directory$(100)  ' String arr                entries.

      FileNumber = FREEFILE  ' Next free BASIC file number.
      OPEN "PIPE:" + Child$ FOR INPUT AS FileNumber

      WHILE NOT EOF(1)  ' Loop until the child process ends.
          LINE INPUT #FileNumber, Directory$(NumEntries)  ' Store entry.
          NumEntries = NumEntries + 1  ' Increment counter.
      WEND

      ChildDone:                  ' The child process ended normally.
      CLOSE FileNumber             ' Close the BASIC file.
      FOR j=0 TO NumEntries       ' Display the file entries.
          PRINT Directory$(j)

>?FileNumber
1.00000
>
```

Microsoft CodeView enhances your debugging productivity for both MS OS/2 and MS-DOS executables. Multiple windows let you see everything that's going on—both inside the program and inside the CPU—while the program is running.

Support for modular and structured programming.

By supporting modular and structured programming, Microsoft BASIC reduces your development time, simplifies program logic, and makes your programs easier to maintain. Language features that enhance this support include programming constructs such as SELECT CASE, DO...LOOP, FUNCTION, and block IF/THEN/ ELSE/END IF statements.

Customized run-time modules.

You can compile your BASIC programs either as standalone executable files or as files that require a Microsoft BASIC compiler 6.0 run-time module to execute. If you choose the latter, you can extend the standard BASIC run-time module with routines you write yourself.

This option, available for both protected-mode and real-mode programs, can result in considerably reduced program size on disk. By embedding often-used routines in the run-time module, where they can be instantly accessed by any application you create, you streamline both the development process and your executable files.

The first protected-mode editor.

Microsoft Editor is a "smart" editor. It lets you run programs from within itself and reports compilation errors directly to the source code.

It's also programmable, so you can customize it to meet your needs. For example, you can change keyboard functions, create macros that perform complex editing functions with a single keystroke, or write your own editing functions in C.

You can even make Microsoft Editor look like your favorite text editor. To make the process simpler, it comes with built-in keystroke mapping to emulate BRIEF®, WordStar®, Epsilon™, Microsoft QuickC, and Microsoft QuickBASIC. And since Microsoft Editor works in real mode as well as protected mode, you can use it to edit both MS-DOS and MS OS/2 programs.

Microsoft QuickBASIC version 4.0 included.

To help you create MS-DOS programs faster than ever, we've included Microsoft QuickBASIC 4.0 with Microsoft BASIC compiler 6.0. With its in-memory editing and debugging, Microsoft QuickBASIC provides an instant programming environment for MS-DOS programs, letting you move from one function to another without compiling. You can run, edit and debug, then continue to run your program. There's no time-consuming compile step. And whenever you edit your code, Microsoft QuickBASIC 4.0 incorporates your changes so quickly—usually at 150,000 lines per minute*—that it seems instantaneous!

Microsoft QuickBASIC comes with a multi-window, multi-file, full-screen editor that automatically checks your code for syntax errors as you enter each line. And if you aren't sure how to correct an error, context-sensitive help is available on line.

Microsoft QuickBASIC also includes a built-in code outliner that lets you view and manipulate all modules in memory. And you can edit any two subprograms or functions from a module simultaneously.

Time-saving utilities.

We've added several new utilities that provide you with additional flexibility while also reducing your development and maintenance time. These include:

- LIB, a library manager that lets you store modules written in any Microsoft language and call them directly from your Microsoft BASIC programs.
- ILINK, a high-speed incremental linker for MS OS/2 programs that works up to 20 times faster than a full link because it only relinks the modules that have been changed since the last link.
- MAKE, a maintenance utility that keeps your program components up to date by automatically recompiling and relinking them as you make source code changes.
- EXEPACK, a utility that removes null characters from EXE files and optimizes the relocation table, resulting in smaller files and faster loading.
- EXEMOD, a utility that lets you fine-tune file header information such as memory allocation values and the initial stack pointer, normally set by default.

Language Details

Program statements

- CALL
- CALL ABSOLUTE
- CALLS
- CALL INT86OLD
- CALL INTERRUPT
- CHAIN
- COMMON
- CONST
- DECLARE
- DEF FN
- DEFINT
- DEFSNG
- DEFDBL
- DEFLNG
- DEFSTR
- DIM
- DO...LOOP
- END {DEF • IF • SUB}
- ERASE
- ERR
- ERL
- ERROR
- EXIT {DO • FOR • DEF • SUB}
- FOR...NEXT
- FUNCTION
- GOSUB...RETURN
- GOTO
- IF...THEN...ELSE
- INP
- LET
- ON COM GOSUB
- ON ERROR GOTO
- ON...GOSUB
- ON...GOTO
- ON KEY GOSUB
- ON PLAY GOSUB
- ON SIGNAL
- ON TIMER
- ON UEVENT
- OPTION BASE
- PEEK
- POKE
- REDIM
- REM
- RESET
- RESUME
- RETURN
- RUN
- SELECT CASE
- SHARED
- SHELL
- SIGNAL ON • OFF • STOP
- SLEEP
- STATIC
- STOP
- SUB...END SUB
- SWAP
- SYSTEM
- TYPE
- UEVENT ON • OFF • STOP
- WAIT
- WHILE...WEND

Input/output statements & functions

- CLOSE
- CLS
- COM
- CSRLIN
- CVD
- CVI
- CVL
- CVS
- DATA
- EOF
- FIELD
- FILEATTR
- FREEFILE
- GET
- INKEY$
- INPUT
- INPUT#
- INPUT$
- KEY
- KILL
- LINE INPUT
- LINE INPUT#
- LOC
- LOCATE
- LOCK
- LOF
- LPOS
- LPRINT
- LPRINT USING
- MKD$
- MKI$
- MKS$
- NAME
- OPEN
- OPEN COM...
- OPEN PIPE:
- OUT
- PCOPY
- POS
- PRINT
- PRINT #
- PRINT USING
- PUT
- READ
- RESET
- RESTORE
- SEEK
- SPC
- TAB
- UNLOCK
- WIDTH
- WIDTH LPRINT
- WRITE
- WRITE #

Arithmetic functions

- ABS
- ATN
- CDBL
- CINT
- CLNG
- COS
- CSNG
- EXP
- FIX
- INT
- LOG
- SGN
- SIN
- SQR
- TAN

String functions

- ASC
- CHR$
- CVI
- CVD
- CVDMBF
- CVS
- CVSMBF
- HEX$
- INSTR
- LCASE$
- LEFT$
- LEN
- LSET
- LTRIM$
- MID$
- MKD$
- MKDMBF$
- MKI$
- MKL$
- MKS$
- MKSMBF$
- OCT$
- RIGHT$
- RSET
- RTRIM$
- SADD
- SPACE$
- STR$
- STRING$
- UCASE$
- VAL

Language Details (cont.)

Graphics statements & functions

- CIRCLE
- COLOR
- DRAW
- GET
- LINE
- PAINT
- PALETTE
- PALETTE USING
- PEN
- PMAP
- POINT
- PRESET
- PSET
- PUT
- SCREEN
- VIEW
- VIEW PRINT
- WINDOW

Sound & music

- BEEP
- PLAY
- PLAY ON • OFF • STOP
- SOUND

Special-purpose statements & functions

- BLOAD
- BSAVE
- CHDIR
- CLEAR
- COMMAND$
- DATE$
- ENVIRON$
- ENVIRON
- ERDEV
- ERDEV$
- FILES
- FRE
- IOCTL
- IOCTL$
- LBOUND
- MKDIR
- RANDOMIZE
- RMDIR
- RND
- SETMEM
- TIME$
- TIMER
- TIMER ON • OFF • STOP
- UBOUND
- VARPTR
- VARSEG
- VARPTR$

Operators

- =
- ^
- <
- >
- -
- *
- <=
- +
- <>
- /
- >=
- AND
- EQV
- IMP
- MOD
- NOT
- OR
- XOR

Metacommands

- $DYNAMIC
- $INCLUDE
- $STATIC

Specifications

System requirements

- MS OS/2 1.0 or higher or MS-DOS 2.1 or higher (for MS-DOS, IBM® Personal Computer or compatible required)
- 320K available user memory
- One double-sided disk drive and a hard disk

NOTE: Use of the Microsoft Mouse is optional.

Documentation

The manuals supplied with Microsoft BASIC compiler version 6.0 come in sturdy three-ring binders and are comprehensive, well organized, and full of illustrative examples.

User's Guide
This manual describes the new features of Microsoft BASIC compiler version 6.0.

Learning and Using Microsoft QuickBASIC
This user's manual describes the programming environment and features of Microsoft QuickBASIC version 4.0, including the compiler options available for Microsoft BASIC compiler version 6.0.

Programming in BASIC: Selected Topics
This tutorial helps familiarize both professionals and beginners with powerful techniques in a variety of areas, including graphics, error handling, and file I/O. Complete examples are included on disk.

BASIC Language Reference
This reference manual describes the syntax and use of all BASIC commands, statements, and functions.

Microsoft CodeView and Utilities, Microsoft Editor
This comprehensive manual is a complete guide to the capabilities and operation of the window-oriented debugger, text editor, and other utilities included with the compiler.

Shipping media

- 5¼" disks in standard MS-DOS and MS OS/2 format.
- 3½" disks available at no charge; details provided in package.

* On an IBM PC/AT® running at 8 MHz.

This data sheet is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.

Microsoft Corporation ▪ 16011 NE 36th Way ▪ Box 97017 ▪ Redmond, WA 98073-9717
1287 Part No. 098-007-229

Microsoft®