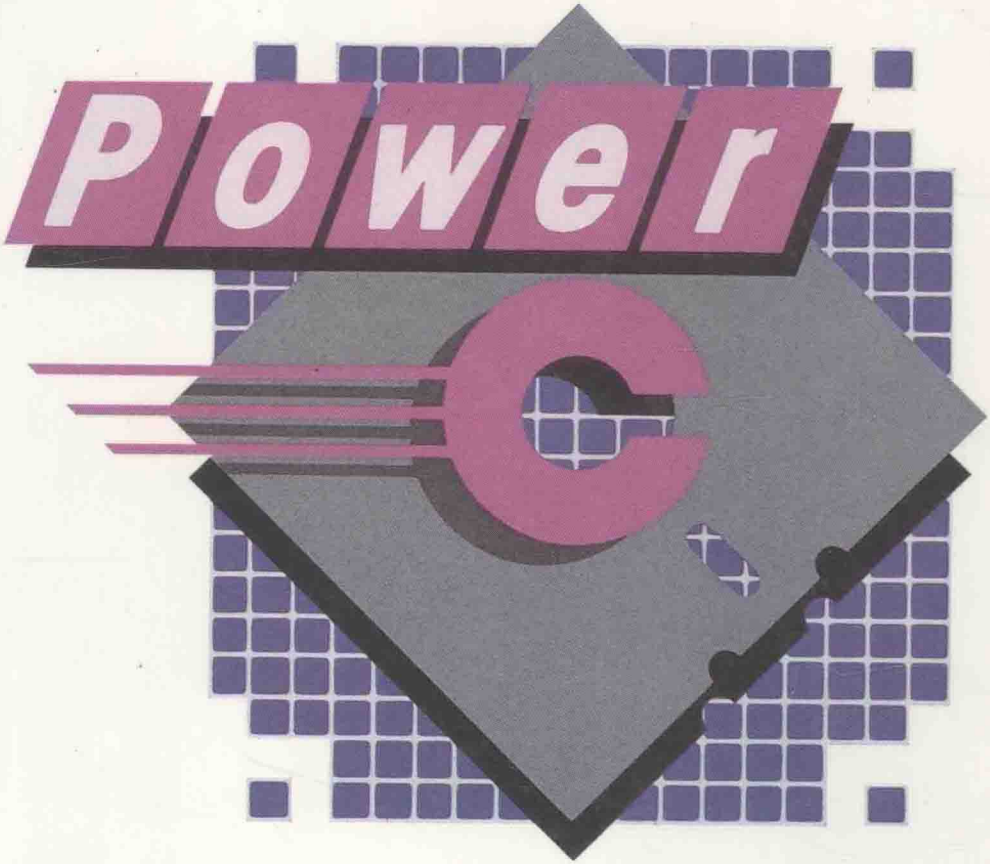

The High-Performance C Compiler

MIX software

Power C

The High-Performance C Compiler

Please see the **READ.ME** file on diskette before proceeding.

Software License Agreement

Mix Software, Inc. grants the purchaser a single CPU license for use of the accompanying software. The software is not to be used on more than one computer at the same time. The purchaser shall not make copies of the software except for personal use. Mix Software, Inc. reserves the right to prosecute any person or organization found to be making illegal copies of the software.

Disclaimer

Mix Software, Inc. makes no warranties as to the contents of this manual or the accompanying software. Although every effort has been made to insure that the manual is accurate and the software reliable, Mix Software, Inc. cannot be held responsible for any damages suffered from the use of this product.

Introduction

Congratulations on your purchase of Power C: the ANSI compatible, high-performance C compiler for creating portable, high-performance programs.

Performance and portability are the two main reasons why programmers choose C over other programming languages. With Power C, you have the capability to develop high-performance C programs that can easily be moved to other computers or operating systems. Support for the ANSI C standard, the UNIX System V functions, and the common features of other C compilers, makes it very easy to move existing C programs to Power C.

Power C is a full-featured C compiler with an integrated *Make* utility. With Power C, your programs can be as large as the available memory. The integrated *Make* utility saves you time and effort by automatically managing your large programming projects. Efficient mixed model programming is supported through the use of **near** and **far** pointers. Power C also includes fast IEEE floating point software and support for the 8087/80287 coprocessor. Advanced features like automatic register allocation make Power C one of the top-performing C compilers.

The Power C function library is extensive, containing approximately 400 functions. The library contains both the ANSI and System V compatible functions. In addition, the library contains a wide variety of functions that make your programming tasks easier to accomplish. Included are system functions that give you direct access to the operating system and hardware. Screen management functions give you complete control of the screen. Graphics functions make it easy to draw lines, boxes, circles, ellipses, and pie charts. The CGA, EGA, VGA, and Hercules graphics cards are supported.

We hope that you enjoy using Power C as much as we've enjoyed creating it for you. The programs that you create with Power C may be distributed royalty free.

Other Mix Software Products

To order one or more of the following products, send a check, money-order, or your mastercard or visa credit card number to Mix Software, 1132 Commerce Drive, Richardson, TX 75081. Please add $5.00 for shipping and handling. Texas residents, please add 8% sales tax.

Power C Library Source Code ($10.00)

The Power C libraries contain more than 400 functions, most of which are written in C. Some of the low-level I/O functions and other speed critical functions are written in assembly language. Th C and assembly language source code for the Power C libraries is available for only $10.00. An assembler and library management utility is included with the library source code.

Power Ctrace Debugger ($19.95)

The Power Ctrace debugger is a state-of-the-art C language source debugger - extremely powerful and very easy to use. Power Ctrace is a great learning aid for beginners and a tremendous time saver for professionals. With animated trace, you watch your C source code execute statement by statement. The values of all variables (including arrays and structures) are automatically displayed. Unlimited statement break points, as well as variable and memory watch points, give you complete control over the execution of your program. Power Ctrace is capable of displaying 8 windows of information: C source code, assembly source code, variables, memory, watch points, symbols, multi-dimensioned arrays, and program output.

Split-Screen Editor ($19.95)

The Mix Editor is a powerful split-screen text processor - the ideal tool for creating and editing your C programs. The split-screen capability allows you to view and edit two files at once, with the screen split either horizontally or vertically. Text can be moved back and forth between files. More than 100 editor commands are predefined - plus you can define your own commands using macros. From within the editor, you can execute any DOS command or run other programs. Combined with Power C and Power Ctrace, the editor creates an integrated programming environment.

Overview of the Power C Manual

The Power C manual is divided into the following six sections.

1. **Getting Started** - Takes you through the steps of compiling, linking, and executing programs with Power C.

2. **Tutorial** - Introduces you to the C language with numerous example programs to aid your learning.

3. **Reference** - A detailed description of the C language.

4. **Functions** - An alphabetic list of all the library functions.

5. **Tools** - A detailed description of the Power C compiler, linker, and other supplied programs.

6. **Appendix** - Miscellaneous information on preprocessor directives, compiler options, memory usage, etc...

Disk Files

Power C includes: Power C compiler (PC.EXE), Power C optimizer (PCO.EXE), header files (*xxx*.H), Power C linker (PCL.EXE), Power C function libraries (PC*xxx*.MIX), object code translator (MIX.EXE), assembly language macros (PCMAC.ASM), and sample programs (*xxx*.C and *xxx*.PRJ). The files are supplied on two diskettes as follows.

Disk 1 of 2

PC.EXE	Power C compiler
PCO.EXE	Power C optimizer
ASSERT.H	defines **assert** debugging macro
BIOS.H	declares BIOS interface functions
CONIO.H	declares console input and output functions
CTYPE.H	declares character handling functions
DIRECT.H	declares directory functions
DOS.H	declares DOS interface functions
ERRNO.H	defines macro names for error numbers
FCNTL.H	defines macros used with **open** function
FLOAT.H	defines limits for floating point values
GRAPHICS.H	declares graphics functions
IO.H	declares low level input and output functions
LIMITS.H	defines limits for character and integer values
LOCALE.H	declares **setlocale** function
MATH.H	declares math functions
MALLOC.H	declares memory allocation functions
MEMORY.H	declares block memory functions
PROCESS.H	declares *exec* and *spawn* functions
SEARCH.H	declares string search functions
SETJMP.H	declares **setjmp** and **longjmp** functions
SHARE.H	defines file sharing macros
SIGNAL.H	declares signal handling functions
STDARG.H	defines variable argument macros
STDDEF.H	defines standard types and macros
STDIO.H	declares standard input and output functions
STDLIB.H	declares general utility functions
STRING.H	declares string handling functions
SYS\LOCKING.H	defines macros used by the **locking** function
SYS\STAT.H	declares **stat** and **fstat** functions
SYS\TIMEB.H	declare **ftime** function
SYS\TYPES.H	defines types used in SYS\STAT.H
SYS\UTIME.H	declares **utime** function
TIME.H	declares time and date functions

Disk 2 of 2

PCL.EXE	Power C linker
PCLIB.MIX	Power C main function library
PCLIB2.MIX	Power C secondary function library
PCAUTO.MIX	Power C auto-sensing floating point library (8087 + IEEE)
PCDMY.MIX	Power C dummy library (no floating point)
PC87.MIX	Power C 8087 hardware only floating point library
PCIEEE.MIX	Power C IEEE software only floating point library
MIXC.MIX	MIX C compatible functions
MIX.EXE	utility for converting .OBJ files to .MIX files
PCMAC.ASM	macros for writing assembly language functions
BARCHART.C	sample program that draws a bar chart
PIECHART.C	sample program that draws a pie chart
PATTERNS.C	function that generates fill patterns for piechart program
SCREEN.C	function that sets screen mode for piechart program
PIECHART.PRJ	sample project file for pie chart program
READ.ME	Contains important information you should read

Table of Contents

GETTING STARTED

GETTING STARTED

The first thing you should do is make backup copies of the two supplied master disks. Then put the master disks in a safe place and use the backups. Label the backup disks the same as the master disks are labeled, *disk 1 of 2* and *disk 2 of 2*. The disks do not contain an operating system.

Arranging Files on a Hard Disk

If your computer has a hard disk drive, you'll want to make a directory for the Power C files on drive C. Type **C:** to make drive C the selected drive, then type **CD ** to make sure that the root directory is the current directory. Type **PROMPT PG** to make the DOS prompt include the current directory name. The DOS prompt should now appear as follows, with the \ indicating the root directory.

```
C:\>
```

Type the following command to make a directory named **POWERC**.

```
C:\>MD POWERC
```

Now place the Power C *disk 1 of 2* into drive A. Type the following command to copy all the files from drive A into the **POWERC** directory.

```
C:\>COPY A:*.* POWERC
```

A few of the files on *disk 1 of 2* are stored in a directory named **SYS**. Type the following DOS command to make a subdirectory named **SYS** in the **POWERC** directory.

```
C:\>MD  POWERC\SYS
```

Now copy the files from the **SYS** directory on drive A to the **SYS** subdirectory in the **POWERC** directory on drive C. Use the following DOS command.

```
C:\>COPY A:SYS\*.* POWERC\SYS
```

Place the Power C *disk 2 of 2* into drive A. Type the following DOS command to copy all the files from drive A to the **POWERC** directory on drive C.

```
C:\>COPY A:*.* POWERC
```

All of the Power C files are now stored in the directory named **POWERC** on drive C. To verify that the files are there, type the following two DOS commands to list the files in the **POWERC** directory and the **POWERC\SYS** subdirectory.

```
C:\>DIR/W POWERC
```

Directory of C:\POWERC

```
.           ..          PC       EXE   PCO      EXE   ASSERT   H
BIOS     H  CONIO    H  CTYPE    H     DIRECT   H     DOS      H
ERRNO    H  FCNTL    H  FLOAT    H     GRAPHICS H     IO       H
LIMITS   H  LOCALE   H  MATH     H     MALLOC   H     MEMORY   H
PROCESS  H  SEARCH   H  SETJMP   H     SHARE    H     SIGNAL   H
STDARG   H  STDDEF   H  STDIO    H     STDLIB   H     STRING   H
SYS         TIME     H  PCL      EXE   PCLIB    MIX   PCLIB2   MIX
PCAUTO   MIX PCDMY    MIX PC87    MIX   PCIEEE   MIX   MIXC     MIX
MIX      EXE PCMAC    ASM BARCHART C     PIECHART C    PATTERNS C
SCREEN   C   PIECHART PRJ READ     ME
```

```
C:\>DIR POWERC\SYS
```

Directory of C:\POWERC\SYS

```
.             <DIR>
..            <DIR>
LOCKING  H
STAT     H
TIMEB    H
TYPES    H
UTIME    H
```

Now define a path to the **POWERC** directory so that the files can be located from any other drive or directory. Directory paths are defined using the DOS **PATH** command. You probably already have a **PATH** command defined in the file named **AUTOEXEC.BAT** in the root directory. This is a batch file that contains DOS commands that are automatically executed each time your computer is restarted (ie. when the computer is first turned on or when CTRL ALT DEL is pressed). Type the following command to list the contents of this file.

```
C:\>TYPE AUTOEXEC.BAT
```

If the **AUTOEXEC.BAT** file exists, it probably contains a line that begins with **PATH=**, followed by a list of one or more directory names that are separated by semicolons. Use an ASCII text editor to add **C:\POWERC** to the list of directory names, preferrably as the first directory name in the list. For example, you should have a line in the **AUTOEXEC.BAT** file

that is similar to the following, where *dirname* represents the names of one or more other directories (if present).

PATH=C:\POWERC;*dirname*;*dirname*

If the **AUTOEXEC.BAT** file does not exist, type the following to create it. The **^Z** in the third line means hold down the **Ctrl** key, then press the **Z** key.

```
C:\>COPY CON AUTOEXEC.BAT
PATH=C:\POWERC;
^Z
```

After creating (or modifying) the **AUTOEXEC.BAT** file, type the following to execute the commands in this batch file.

```
C:\>AUTOEXEC
```

Now you can operate Power C from any disk drive or directory. The operating system will be able to locate the executable (.EXE) programs in the **POWERC** directory without you having to specify a disk drive or directory prefix. For example, you can simply type **PC** rather than **C:\POWERC\PC** to execute the Power C compiler. The Power C compiler (PC.EXE) will also be able to locate all the header (.H) files, and the Power C linker (PCL.EXE) will be able to locate all the library (.MIX) files.

Special Note: *Many hard disk users prefer to put all executable (.EXE) files into a single directory named* ***BIN****, then execute the command* ***PATH=C:\BIN*** *to define a path to this directory. With only one directory to search, the operating system can more quickly locate programs when the file name is specified without a drive or directory prefix. If you have a directory named* ***BIN*** *with a path defined to it, then you may want to copy the executable files from the* ***POWERC*** *directory to the* ***BIN*** *directory (eg. type COPY C:\POWERC*.EXE C:\BIN). Without a path defined to the* ***POWERC*** *directory, you will need to set the environment variables named* ***PCOPTION*** *and* ***LIBRARY*** *so that the compiler can locate header (.H) files and the linker can locate library (.MIX) files. The following two lines may be added to the* ***AUTOEXEC.BAT*** *file to set these two environment variables.*

```
SET PCOPTION=/iC:\POWERC
SET LIBRARY=C:\POWERC
```

You should now create a different directory for storing your program files. For example, type the following command to create a directory named MYFILES.

```
C:\>MD MYFILES
```

Type the following command to make MYFILES the current directory.

```
C:\>CD MYFILES
```

The DOS prompt should now include the directory name MYFILES. Type the following two commands to copy the sample programs from the POWERC directory to the MYFILES directory.

```
C:\MYFILES>COPY \POWERC\*.C
C:\MYFILES>COPY \POWERC\*.PRJ
```

Type **DIR** to display the names of the files in the MYFILES directory.

```
 Directory of C:\MYFILES

.              <DIR>
..             <DIR>
BARCHART   C
PIECHART   C
PATTERNS   C
SCREEN     C
PIECHART   PRJ
```

Now you are ready to compile, link, and execute these sample programs.

Arranging Files on Floppies

If your computer does not have a hard disk drive, here is a suggested configuration for a system with two floppy drives. Insert a DOS system disk into drive A and press **Ctrl Alt Del** if the operating system is not already booted. If the DOS prompt is not **A>**, type **A:** to make drive A the selected drive.

Place a blank disk into drive B and type the following command to format it as a system disk.

```
A>FORMAT B: /S
```

Now replace the DOS system disk in drive A with the formatted system disk from drive B. Type the following lines to create an **AUTOEXEC.BAT** file on the new system disk. The **^Z** on the third line means to hold down the Ctrl key and press the Z key.

```
A>COPY CON AUTOEXEC.BAT
PATH=B:
^Z
```

Type the following command to verify that the **AUTOEXEC.BAT** file has been created and contains the line **PATH=B:**.

```
A>TYPE AUTOEXEC.BAT
```

Type the following command to execute the **AUTOEXEC.BAT** file and define a path to drive B.

```
A>AUTOEXEC
```

Place the disk labeled *disk 2 of 2* into drive B and type the following two commands to copy the sample programs from drive B to drive A.

```
A>COPY B:*.C
A>COPY B:*.PRJ
```

Now type **DIR** to display the names of the files on the disk in drive A.

```
 Directory of A:\

COMMAND    COM
AUTOEXEC   BAT
BARCHART   C
PIECHART   C
PATTERNS   C
SCREEN     C
PIECHART   PRJ
```