

# **DeSmet C**

## **Development Package**

**Published and Distributed by**

**C WARE CORPORATION**

**Sunnyvale, California**



# **DeSmet C**

## **Development Package**

**Mark DeSmet**

Published and Distributed by

**C Ware Corporation**

Sunnyvale, California

## DeSmet C Development Package

Version 2.5 — October, 1985  
Version 2.4 — October, 1984  
Version 2.3 — April, 1984

Published by: C Ware Corporation  
P.O. Box C  
Sunnyvale, CA 94087  
USA  
(408) 720-9696  
Telex 358185 C WARE SNVLE

Copyright © 1982, 1983, 1984, 1985 by DeSmet Software

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means electronic, mechanical, photocopying, recording or otherwise without prior written permission of the publisher.

### DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITIES

The author has taken due care in preparing this book and the programs and data on the electronic media accompanying this book including research, development, and testing to ascertain their effectiveness. The author and the publisher make no expressed or implied warranty of any kind with regard to these programs nor the supplemental documentation in this book. In no event shall the author or C Ware Corporation be liable for incidental or consequential damages in connection with or arising out of the furnishing, performance or use of any of these programs.

DeSmet C Development Package and SEE are Trademarks of C Ware Corporation.

CP/M-86 is a Trademark of Digital Research, Inc.  
IBM is a Registered Trademark of International Business Machines.  
MSDOS is a Trademark of Microsoft, Inc.  
UNIX is a Trademark of Bell Laboratories.

## Preface

This manual describes the DeSmet C Development Package for the IBM-PC personal computer and the other MS-DOS based personal computers. If you are unfamiliar with the C language or UNIX, the book *The C Programming Language* by Brian Kernighan and Dennis Ritchie is available. If you plan on coding in assembly language, it is advisable to get a manual on the Intel 8086 microprocessor.

Books such as Intel's *ASM86 Language Reference Manual* or *The 8086 Family User's Guide* are good choices. These manuals fully describe the architecture and the instruction set of the 8086/8088 family of microprocessors.

We thank Mike Ouye (pronounced Oh'Way) for converting the manual into MacWrite format, and both Glen Emigh and Pacific Data Works for proofreading its many revisions.

We especially thank the following for repeatedly throwing their bodies in front of and on top of the more explosive parts of the Version 2.5 Beta release — Clopper Almon, Dave Auslander, Glen Emigh, Panos Galidas, Scott Guthery, Bill Hunt, George O'Neal Keyes II, Scott Lewis, Greg Mansfield, Bill Morrison, and Jim Rome, as well as the folks at Kris Jamsa Software (Las Vegas, NV), Nanosoft (Ft. Belvoir, VA), and Pacific Data Works (Santa Monica, CA).



## Table of Contents

1.	Introduction	
2.	Getting Started	
2.1	Distribution Disks	2.1
2.2	A Short Example	2.2
2.3	Setting Up DOS 2.xx, 3.xx	2.6
2.3.1	RAM Disk	2.6
2.3.2	Completion Codes	2.7
3.	The SEE Text Editor	
3.1	Introduction	3.1
3.2	Getting Started	3.2
3.2.1	Concepts	3.2
3.2.2	Starting the Editor	3.3
3.2.3	Inserting and Editing Text	3.4
3.2.4	Saving the File	3.9
3.2.5	Editing Existing Files	3.10
3.3	The Invocation Line	3.11
3.4	The Keyboard	3.12
3.4.1	Cursor Movement Keys	3.12
3.4.2	Editing Keys	3.13
3.4.3	The DOS Key	3.14
3.5	Commands	3.15
3.6	Configuration	3.29
4.	The C88 C Compiler	
4.1	Introduction	4.1
4.2	Invocation	4.1
4.3	Examples	4.2
4.4	The C88 Language	4.3
4.4.1	Preprocessor directives	4.3
4.4.2	Data Types	4.4
4.4.3	Extensions	4.5
4.4.4	Forward References	4.7
4.4.5	Externs	4.7
4.4.6	Macros	4.9
4.4.7	Strings	4.9

## 5. The ASM88 8088/8086 Assembler

5.1	Introduction	. . . . .	5.1
5.2	Invocation	. . . . .	5.1
5.3	Examples	. . . . .	5.3

## 6. The BIND Object File Linker

6.1	Introduction	. . . . .	6.1
6.2	Invocation	. . . . .	6.1
6.3	Examples	. . . . .	6.3
6.4	Space Considerations	. . . . .	6.3
6.5	Overlays	. . . . .	6.4
6.6	Libraries	. . . . .	6.6

## 7. The LIB88 Object File Librarian

7.1	Introduction	. . . . .	7.1
7.2	Invocation	. . . . .	7.1
7.3	Examples	. . . . .	7.2
7.4	Libraries	. . . . .	7.2

## 8. The D88 C Language Debugger

8.1	Introduction	. . . . .	8.1
8.2	D88 Usage	. . . . .	8.1
8.3	Command Input	. . . . .	8.3
8.4	Expressions	. . . . .	8.3
8.5	D88 Commands	. . . . .	8.5

## 9. Utility Programs

9.1	Dump: Hex file display program	. . . . .	9.1
9.2	CList: C program list utility	. . . . .	9.1
9.3	Profile	. . . . .	9.3

## 10. The CSTDIO.S Standard Library

10.1	Introduction	. . . . .	10.1
10.2	Names	. . . . .	10.1
10.3	Program Initialization	. . . . .	10.1
10.4	Calling Conventions	. . . . .	10.3
10.5	Library Conventions	. . . . .	10.4
10.6	Disk Input/Output Routines	. . . . .	10.6
10.7	Math Routines	. . . . .	10.7
10.8	IBM-PC Screen and Keyboard Interface	. . . . .	10.8
10.9	Alphabetical Function Index	. . . . .	10.8

## Appendix A: Messages

A.1	SEE Messages . . . . .	A.1
	A.1.1 Banner and Termination Messages . . . . .	A.1
	A.1.2 Error and Status Messages . . . . .	A.1
A.2	C88 Compiler Messages . . . . .	A.2
	A.2.1 Banner and Termination Messages . . . . .	A.2
	A.2.2 Messages . . . . .	A.3
	A.2.2.1 C88 Fatal Errors . . . . .	A.3
	A.2.2.2 C88 Errors . . . . .	A.5
	A.2.2.3 C88 Warnings . . . . .	A.9
	A.2.2.4 ASM88 Detected Errors . . . . .	A.10
A.3	Assembler Messages . . . . .	A.10
	A.3.1 Banner and Termination Messages . . . . .	A.10
	A.3.2 Messages Produced by ASM88 . . . . .	A.11
	A.3.2.1 Fatal Errors From ASM88 . . . . .	A.11
	A.3.2.2 Errors from ASM88 . . . . .	A.12
A.4	BIND Error Messages . . . . .	A.15
	A.4.1 Banner and Termination Messages . . . . .	A.15
	A.4.2 Warnings from BIND . . . . .	A.16
	A.4.3 Fatal Errors from BIND . . . . .	A.16
A.5	LIB88 Messages . . . . .	A.17
	A.5.1 Banner and Termination Messages . . . . .	A.17
	A.5.2 Warnings from LIB88 . . . . .	A.17
	A.5.3 Fatal Errors from LIB88 . . . . .	A.18
A.6	D88 Messages . . . . .	A.18
A.7	CLIST Messages. . . . .	A.20
	A.7.1 Banner and Termination Messages . . . . .	A.20
	A.7.2 Messages Produced by CLIST. . . . .	A.20

## Appendix B: The ASM88 Assembly Language

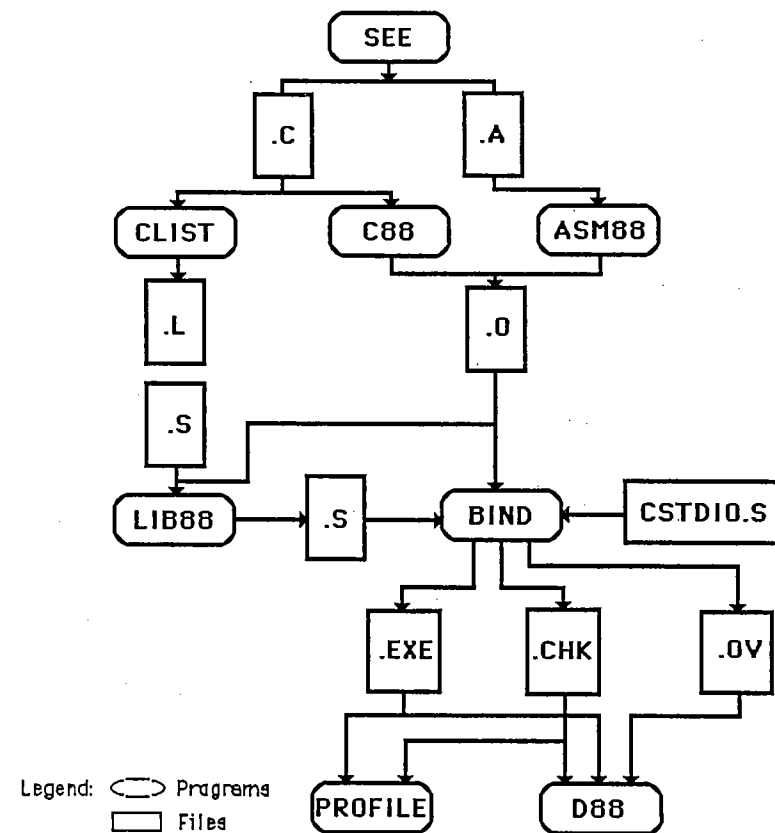
B.1	Identifiers . . . . .	B.1
B.2	Constants . . . . .	B.1
B.3	Expressions . . . . .	B.2
B.4	Addressing Modes . . . . .	B.3
B.5	8086 Flags . . . . .	B.4
B.6	Address Expressions . . . . .	B.5
B.7	Address Typing . . . . .	B.5
B.8	Comments . . . . .	B.5
B.9	Assembler Directives . . . . .	B.6
B.10	Reserving Storage . . . . .	B.7
B.11	Differences Between Intel ASM86 and ASM88 . . . . .	B.7
B.12	8086 Instructions . . . . .	B.8
	B.12.1 Elements of Instructions . . . . .	B.8

DeSmet C Development Package, Version 2.5

	B.12.2 Instructions	.	.	.	.	.	B.8
B.13	Floating Point	.	.	.	.	.	B.21
	B.13.1 Control Word	.	.	.	.	.	B.22
	B.13.2 Status Word	.	.	.	.	.	B.23
	B.13.3 Tag Word	.	.	.	.	.	B.23
	B.13.4 Condition Codes.	.	.	.	.	.	B.24
	B.13.5 8087 Instructions	.	.	.	.	.	B.25

# 1. Introduction

The DeSmet C Development Package is a set of programs and files for developing applications in the C programming language for the IBM-PC personal computer and its clones. The programs provided in this package require a minimum of 128K of Random Access Memory (RAM) and at least one disk drive. D88 requires 192K. Most programs will run under all versions of MS-DOS, 1.xx, 2.xx, and 3.xx. The program execution profiler requires the use of MS-DOS 2.x or later versions.



The diagram above outlines the interrelationships between some of the programs which are provided.

**SEE** is a full-screen, command oriented text editor designed for program editing rather than word processing. While SEE can edit any standard ASCII text file, its main purpose is to produce C [.C] and Assembler source files [.A]

**CLIST** reads C source files [.C] and produces a listing file with a symbol cross-reference.

**C88** is the C compiler. It reads C source files [.C] and produces either object files [.O] or assembler files [.A]. It supports the complete Kernighan and Ritchie C language plus the UNIX V7 extensions — structure assignment and parameter passing, and enumerated types.

**ASM88** is the 8086/8088 assembler. It reads assembler source files [.A] and produces linkable object files [.O].

**BIND** is the object file linker. It reads object files [.O] and library files [.S] and produces an executable file [.EXE]. BIND optionally produces the debugger information file [.CHK] and overlay files [.OV].

**LIB88** is the object file librarian. It reads object files [.O] and other library files [.S] and produces library files [.S].

**D88** is the C source-level symbolic debugger. It provides access to program variables by name, breakpoints by function name and line number, and special support for debugging interactive programs. Source code display and stepping by source lines are also supported.

**PROFILE** is the C program execution profiler. It monitors the execution of the application program and indicates where time is spent in the program.

**CSTDIO.S** is the Standard Library used by BIND to provide the Operating System and machine-level functions supported by the C language. Two libraries are provided in the development package, one that support the 8087 math coprocessor directly (CSTDIO7.S) and one that provides numeric support in software (CSTDIO.S).

## 2. Getting Started

First things first. Copy all of the files from the distribution disks onto a set of working floppy diskettes or hard disk. The package is not copy-protected so the MS-DOS copy command can be used to copy the files. The package is distributed on two DOS 2 double-sided (360KB) diskettes. If your machine has single-sided drives, or only supports DOS 1, the package is available on "flippies" (each side of each diskette is formatted as a single-sided diskette and the diskette must be physically flipped over to access the other side). The distribution diskettes should never be used, they should be kept as the backup copy of the package.

If the package is to be run on a system other than an IBM PC, XT, AT, PCjr or PC-clone, the screen interface for SEE and D88 must be configured before they can be used. See the notes in the file CONFIG.C on Disk #2 for configuration details.

### 2.1 Distribution Disks

The package is distributed on two 5 1/4 inch floppy diskettes, labeled Disk #1 and Disk #2. If you requested "flippies", Disks #3 and #4 are on the back sides of Disk #1 and #2, respectively. Disk #3 contains the rest of Disk #1, while Disk #4 completes Disk #2.

#### Disk #1:

C88.EXE:	The first pass of the C compiler.
GEN.EXE:	The second pass of the C compiler.
ASM88.EXE:	The assembler and third pass of the C compiler.
SEE.EXE:	The full-screen editor.
STDIO.H:	Include file for the Standard I/O package.
MATH.H:	Include file for the Standard Math package.
BIND.EXE:	The object file linker.
LIB88.EXE:	The object file librarian.
EXEC.O:	The Exec() and Chain() functions.
CSTDIO.S:	The standard C function library with software floating-point support.
CSTDIO7.S:	The standard C function library with 8087 support. To use this library, rename it to CSTDIO.S.

Disk #2:

RAM.COM: RAM Disk driver for DOS 2.0 and later operating systems.

PCIO.A: Source code for the PC screen functions.

BUF128.EXE: 128 byte type-ahead buffer program.

BUF128.A: Source code for BUF128.EXE.

DUMP.EXE: The hex file display utility.

DUMP.C: Source code for DUMP.EXE.

LIFE.EXE: Full screen game of Life.

LIFE.C: Source code for LIFE.EXE.

BUGS!.EXE: Arcade game (use 'BUGS! c' for color displays).

LATER.C: Source code for a file modification date checking program.

CB.C: Source code for a brace matching program.

CLIST.EXE The C listing and cross-reference utility.

PROFILE.EXE: The program execution profiler.

PROFSTAR.EXE: Used by PROFILE.EXE.

PROFEND.EXE: Used by PROFILE.EXE.

COMPARE.EXE: The source code comparison utility.

D88.EXE: The C source-level symbolic debugger.

Disk #2 also contains various object files and configuration files for generating the SEE editor and D88 debugger for systems with displays other than the IBM PC. CONFIG.C contains the display and keyboard dependent routines. See the comments in the CONFIG.C file for instructions on customizing SEE and D88.

## 2. 2 A Short Example

This example shows the general method for creating executable programs with this package. It assumes that the disk in the default drive, in this case drive A:, contains the compiler (C88.EXE and GEN.EXE), the assembler (ASM88.EXE), the binder (BIND.EXE), the standard library (CSTDIO.S) and the text editor (SEE.EXE). The source code will reside on drive B:.

Enter the example program with the SEE text editor. To start the SEE text editor, type:

```
see b:example.c
```

The screen will look as follows:

```
Again Buffer Copy Delete Find -find Get Insert Jump --space--
---- reading file: b:example.c... -- new file 0 characters
-

```

Type the letter 'I', or press the 'Ins' key, to put the editor into Insert mode. Now type in the following program:

```
main() {<Ret>
<Tab>printf("%d plus %d is %d\n", 2, 2, 2+2);<Ret>
<Tab>}<Ret>
<Esc>
```

Note that the items <Tab>, <Ret>, and <Esc> indicate the Tab, Return, and Esc keys, respectively. The <Esc> will terminate insert mode and return the editor to command mode. The screen should now appear as follows:

```
Again Buffer Copy Delete Find -find Get Insert Jump --space--
```

```
-----  
main() {  
    printf("%d plus %d is %d\n",2,2,2+2);  
}
```

Now that the program is entered, it must be saved to the disk. Since the name of the file was specified on the command line, just type the sequence of characters, 'Q' for Quit and 'S' for Save-exit. This will store the copy of the file in memory onto the disk and exit from the editor.

To compile the program just entered, type:

**c88 b:example**

and the computer will display:

```
)
```

```
A>c88 b:example  
C88 Compiler V2.5 (c) Mark DeSmet,1982,1983,1984,1985  
end of C88 001E code 0012 data 1% utilization  
A>_
```

If there were errors during the compilation, go back to the editor and make sure that the program was entered correctly. (Note that the .C extension was not needed on the filename given to the compiler. When no extension is given, the compiler automatically assumes .C as the extension, just as the assembler assumes .A and the binder assumes .O).

Now it's time to bind (link) the program to the standard library and create the application program. Do this by typing:

**bind b:example**

The binder will automatically search the CSTDIO.S library so it should not be included in the command line. When the binder successfully links the program, the screen should appear as follows:

```
A>c88 b:example
C88 Compiler V2.5 (c) Mark DeSmet,1982,1983,1984,1985
end of C88 001E code 0012 data 1% utilization
A>bind b:example
Binder for C88 and ASM88 V1.5 (c) Mark DeSmet,1982-85
end of BIND 8% utilization
A>_
```

The application is now ready to run. Type:

**b:example**

and the computer should respond with:

```
A>c88 b:example
C88 Compiler V 2.5 (c) Mark DeSmet,1982,1983,1984,1985
end of C88 001E code 0012 data 1% utilization
A>bind b:example
Binder for C88 and ASM88 V1.5 (c) Mark DeSmet,1982-85
end of BIND 8% utilization
A>b:example
2 plus 2 is 4
A>_
```

Don't worry if some of the utilization numbers are different from those shown in the example. These numbers will vary depending on the system being used (i.e. amount of memory, RAM disk installed, etc.).

## 2.3 Setting Up DOS 2.xx, 3.xx, ...

For systems utilizing DOS 2.x or later versions of the operating systems, make sure that the ASCII text file CONFIG.SYS exists on your boot disk. If it doesn't exist, you can create it with SEE. The file must contain the line:

```
FILES=20
```

since C88 supports 20 open files — stdin, stdout, stderr, and 17 other files. The default number of eight is insufficient for the BIND program. If there is enough memory available, add the line:

```
BUFFERS=20
```

to improve file performance in the operating system. 512 bytes are allocated for each buffer specified.

### 2.3.1 RAM DISK

If you have a system with more than 256 kilobytes of memory, then the Ram Disk driver RAM.COM can be used to create an extremely fast disk. To add a Ram Disk,

copy the RAM.COM file from the distribution diskette to the system disk and add the line:

```
DEVICE=RAM.COM n
```

to the CONFIG.SYS file. The parameter, n, is a decimal number from 32 to 650 indicating the size of the Ram Disk. The value is specified in units of one kilobyte (1024).

Re-boot the system to install the Ram Disk. The drive letter used for this 'disk drive' is dependent on the configuration of the system. DOS will install the Ram Disk at the first free device "slot". For an IBM PC with two floppies, this will probably be drive C:. For an XT, it will probably be drive D:. Sanyo 550/5 reserves the first four slots for its floppies, so the Ram Disk is drive E:. To find where DOS has installed the Ram Disk, use

```
chkdsk x:
```

where x takes on the values c, d, .... You will get either a disk error, or a return showing the size of the Ram Disk. Once you find it, the Ram Disk will always be the same until you add other device drivers before it in the CONFIG.SYS file.

### 2.3.2 Completion Codes

The C88, ASM88, BIND and LIB88 programs set the completion code to:

- zero if no warnings or errors occurred,
- one if warnings were issued, and
- two if errors occurred.

Batch files can take advantage of these values to stop execution or otherwise handle these exceptional cases.

The batch file CC.BAT listed below will stop if C88 or BIND reports an error:

```
c88 %1
if errorlevel 1 goto stop
bind %1
if errorlevel 1 goto stop
%1
:stop
```

More complicated development situations can be handled with the program LATER which is supplied in source form in the file LATER.C. LATER takes a list of

## DeSmet C Development Package, V2.5

filenames as arguments. It sets the errorlevel to one if the last file does not exist or if the last file has an earlier modification date than any other file in the list. It can only be used on systems with a battery backup clock or where users are careful about setting the date and time when the system is brought up. Assume a program is composed of the files moda.c, modb.c, modc.c and the include file mod.h. The following .BAT file can be used to regenerate the program whenever a module changes:

```
later moda.c mod.h moda.o
if errorlevel 1 c88 moda
if errorlevel 1 goto stop
later modb.c mod.h modb.o
if errorlevel 1 c88 modb
if errorlevel 1 goto stop
later modc.c mod.h modc.o
if errorlevel 1 c88 modc
if errorlevel 1 goto stop
later moda.o modb.o modc.o mod.exe
if errorlevel 1 bind moda modb modc -omod
:stop
```

This provides a service similar to the UNIX MAKE program. Only those files that need to be compiled will be compiled.

## 3. The SEE<sup>TM</sup> Text Editor

### 3.1 Introduction

SEE is a general purpose full-screen text editor designed for program entry rather than word processing. It handles files larger than memory and can edit two files simultaneously. Its macro facility allows you to capture a series of keystrokes and replay them to ease repetitive tasks. It also features automatic indentation and brace/bracket/parenthesis matching to ease program entry.

SEE is shipped configured for the IBM-PC and its clones. SEE may be reconfigured to run on other machines which support MS-DOS but have different keyboard and/or screen interfaces than the IBM-PC (see Section 3.6).

