

2.33. DIRECTORY ENTRIES

Offset	Length	Description	Format	Comments
0 (0)	8 bytes	File name**	ASCII chars.	Must be padded with spaces to fill field
8 (8)	3 bytes	File type (Extension)	ASCII chars.	Must be padded with spaces to fill field
B (11)	byte	File attribute byte	Bit codes: Bit 0 = read-only Bit 1 = hidden Bit 2 = system Bit 3 = volume label Bit 4 = directory Bit 5 = archive Bit 6 = RESERVED Bit 7 = RESERVED	See 2.34. File Attribute Byte
C (12)	10 bytes	RESERVED		
16 (22)	word	Time file last updated*	Coded word: (unsigned 16-bit integer) Time = Hr*2048+Min*32+Sec+2	See 2.35. Date/Time Formats
18 (24)	word	Date file last updated*	Coded word: (see above) Date = (Yr-1980)*512+Mon*32+Day	See 2.35. Date/Time Formats
1A (26)	word	Starting cluster number*	Word binary integer*	See 1.15. Common 8086 Family Data Formats
1C (28)	dbl word	File size*	Double word binary integer*	See 1.15. Common 8086 Family Data Formats

Notes:

- There is no period separating the file name and type fields
- Least significant byte first
- **The first byte of the file name indicates status of directory entry, as follows:
 - 00H = name never used
 - 05H = first character of name is really E5H (sigma)
 - E5H = file was used, but has been erased
 - 2EH = entry is a directory (if second byte also 2EH, cluster field contains cluster # of parent directory)

Source: IBM DOS 3.3 Technical Reference, pages 5-10 to 5-13

See Also: 1.15. Common 8086 Family Data Formats
2.34. File Attribute Byte
2.35. Date/Time Formats
2.50. Allowable Characters in Filenames
2.51. File Separator Characters

2.34. FILE ATTRIBUTE BYTE

Bit Number

7	6	5	4	3	2	1	0	Meaning if Set to 1	Meaning if Set to 0
							X	Read-only file	Read/write file
						X		Hidden file	Visible file
					X			System file	Regular file
				X				Volume name	Regular file
			X					Directory name	Regular file
		X						File changed since last backup	File unchanged since last backup
X	X							RESERVED	RESERVED

Version Info: DOS 1.x used only bits 0-3

Notes:

- Bits 3 and 4 are mutually exclusive; you may set none, one or the other one, but not both
- Only one file (in the root directory) may have Bit 3 set

Source: IBM DOS Technical Reference, pages 5-11 to 5-12

See Also: 2.33. Directory Entries

2.33. DIRECTORY ENTRIES

Offset	Length	Description	Format	Comments
0 (0)	8 bytes	File name**	ASCII chars.	Must be padded with spaces to fill field
8 (8)	3 bytes	File type (Extension)	ASCII chars.	Must be padded with spaces to fill field
B (11)	byte	File attribute byte	Bit codes: Bit 0 = read-only Bit 1 = hidden Bit 2 = system Bit 3 = volume label Bit 4 = directory Bit 5 = archive Bit 6 = RESERVED Bit 7 = RESERVED	See 2.34. File Attribute Byte
C (12)	10 bytes	RESERVED		
16 (22)	word	Time file last updated*	Coded word: (unsigned 16-bit integer) Time = Hr*2048+Min*32+Sec+2	See 2.35. Date/Time Formats
18 (24)	word	Date file last updated*	Coded word: (see above) Date = (Yr-1980)*512+Mon*32+Day	See 2.35. Date/Time Formats
1A (26)	word	Starting cluster number*	Word binary integer*	See 1.15. Common 8086 Family Data Formats
1C (28)	dbl word	File size*	Double word binary integer*	See 1.15. Common 8086 Family Data Formats

Notes:

- There is no period separating the file name and type fields
- *Least significant byte first
- **The first byte of the file name indicates status of directory entry, as follows:
 - 00H = name never used
 - 05H = first character of name is really E5H (sigma)
 - E5H = file was used, but has been erased
 - 2EH = entry is a directory (if second byte also 2EH, cluster field contains cluster # of parent directory)

Source: IBM DOS 3.3 Technical Reference, pages 5-10 to 5-13

See Also: 1.15. Common 8086 Family Data Formats
2.34. File Attribute Byte
2.35. Date/Time Formats
2.50. Allowable Characters in Filenames
2.51. File Separator Characters

2.34. FILE ATTRIBUTE BYTE

Bit Number

7	6	5	4	3	2	1	0	Meaning if Set to 1	Meaning if Set to 0
							X	Read-only file	Read/write file
						X		Hidden file	Visible file
					X			System file	Regular file
				X				Volume name	Regular file
			X					Directory name	Regular file
		X						File changed since last backup	File unchanged since last backup
X	X							RESERVED	RESERVED

Version Info: DOS 1.x used only bits 0-3

Notes:

- Bits 3 and 4 are mutually exclusive; you may set none, one or the other one, but not both
- Only one file (in the root directory) may have Bit 3 set

Source: IBM DOS Technical Reference, pages 5-11 to 5-12

See Also: 2.33. Directory Entries

2.35. DATE/TIME FORMATS

In DOS Functions 2AH, 2BH, 2CH, and 2DH, the Date and Time are Passed Using Registers, as Follows:

Element	Register	Format	Allowable Values
Day of Week	AL	Coded value	0=Sunday 1=Monday 2=Tuesday 3=Wednesday 4=Thursday 5=Friday 6=Saturday
Day	DL	Binary value	1-31 (corresponds to date)
Month	DH	Binary value	1-12 (corresponds to month number)
Year	CX	Binary value	1980-2099 (must be in this range)
Hundredths	DL	Binary value	0-99 (corresponds to hundredths of a second)
Seconds	DH	Binary value	0-59 (corresponds to seconds)
Minutes	CL	Binary value	0-59 (corresponds to minutes)
Hour	CH	Binary value	0-23 (hour in military 24-hour style)

In Directory Entries and Function 57H the Date and Time are Kept as Separate 16-bit Values (Least Significant Byte First), as Follows:

Element	Bits Used	Format	Allowable Values
Day	0 - 4	5-bit binary value	1-31 (corresponds to date)
Month	5 - 8	4-bit binary value	1-12 (corresponds to month number)
Year	9 - 15	7-bit binary value	0-119 (year biased by 1980)
Seconds	0 - 4	5-bit binary value	0-29 (multiply by 2 to get seconds)
Minutes	5 - 10	6-bit binary value	0-60 (corresponds to minutes)
Hours	11 - 15	5-bit binary value	0-24 (corresponds to military hours)

Notes: Note unusual format of seconds in directory entries

Source: IBM DOS 3.3 Technical Reference, pages 5-12 to 5-13, 6-98, 6-100

See Also: 3.006. System Functions
 3.046. Function 2AH -- Get System Date
 3.047. Function 2BH -- Set System Date
 3.048. Function 2CH -- Get System Time
 3.049. Function 2DH -- Set System Time
 3.101. Function 57H,00H -- Get Date/Time of File
 3.102. Function 57H,01H -- Set Date/Time of File

2.36. FAT LAYOUTS

12-bit FAT Layout		
Entry #	Example Value	Use
0	FF8	Disk ID byte
1	FFF	Filler
2	003	Cluster value: 000 = unused cluster 001-FEF = next cluster number FF0-FF6 = reserved cluster FF7 = cluster marked bad FF8-FFF = last cluster in file
3	004	
4	005	
5	FFF	
6	000	

Reserved for DOS → { 0, 1 }
 From Directory Entry's Starting Cluster Number → { 2, 3, 4, 5, 6 }

Note:

In this example FAT, the first entry indicates that it is a FAT for a hard disk (FF8). The first directory entry in the directory for that disk has a starting cluster of 2, thus pointing to cluster number two in this table. The second cluster points to the third, the third to the fourth, the fourth to the fifth. The fifth cluster is the last cluster in the file, thus has a value of FFFH.

(Continued)

Table 2.36. Continued

16-bit FAT Layout			
Entry #	Example Value	Use	
0	FFF8	Disk ID byte	
1	FFFF	Filler	
2	0003	Cluster value:	0000 = unused cluster
3	0004		0001-FFFF = next cluster number
4	0005		FFF0-FFF6 = reserved cluster
5	FFFF		FFF7 = cluster marked bad
6	0000		FFF8-FFFF = last cluster in file

Reserved for DOS

From Directory Entry's
Starting Cluster Number**Note:**

In this example FAT, the first entry indicates that it is a FAT for a hard disk (FFF8H). The first directory entry in the directory for that disk has a starting cluster of 2, thus pointing to cluster number two in this table. The second cluster points to the third, the third to the fourth, the fourth to the fifth. The fifth cluster is the last cluster in the file, thus has a value of FFFFH. Remember, words in the FAT are byte swapped (i.e., least significant byte first).

Source:

IBM DOS 3.3 Technical Reference, pages 5-5 to 5-9

See Also:

2.37. Disk ID Bytes

2.37. DISK ID BYTE

ID Byte	Tracks/side	Sectors	Sides	Format
FFH	40	8	2	5.25-inch floppy disk
FEH	40	8	1	5.25-inch floppy disk
	77	2, 6, or 8	1	8-inch floppy disk
FDH	40	9	2	5.25-inch floppy disk
	77	26	2	8-inch floppy disk
FCH	40	9	1	5.25-inch floppy disk
FBH	80	8	2	5.25-inch floppy disk
	80	8	2	3.5-inch microfloppy disk
FAH	80	8	1	5.25-inch floppy disk
	80	8	1	3.5-inch microfloppy disk
F0H	80	18	2	3.5-inch high-density microfloppy disk
F9H	80	9	2	3.5-inch microfloppy disk
	80	15	2	5.25-inch high-density floppy disk
F8H	-	-	-	Fixed disk

Version Info: Beginning with DOS 2.x, the usefulness of the disk ID byte in the FAT was reduced, and it is now considered meaningless, since multiple formats may have the same ID. Microsoft recommends that you use the information in the media descriptor table to determine the type of disk being used.

Notes:

The disk ID byte is the low-order byte of the first cluster indicator in the FAT (e.g., a first cluster value of FFF8H yields a disk ID byte of F8H).

Source:

IBM DOS 3.3 Technical Reference, page 5-6

See Also:

2.39. Disk Partition Table Layout
3.165. Media Descriptor Table

2.38. DISK BOOT RECORD LAYOUT

Offset	Length	Description
0 (0)	3 bytes	JMP to boot code*
3 (3)	8 bytes	OEM name and version
B (11)	word	Bytes per sector
D (13)	byte	Sectors per cluster (must be a power of 2)
E (14)	word	Reserved sectors (for Dir, FAT, etc.)
10 (16)	byte	Number of copies of FAT
11 (17)	word	Maximum number of root directory entries
13 (19)	word	Total number of sectors in logical image
15 (21)	byte	Media descriptor byte
16 (22)	word	Number of sectors in FAT
18 (24)	word	Number of sectors per track
1A (26)	word	Number of heads
1C (28)	word	Number of hidden sectors

Version Info: •Note that media descriptor bytes are not necessarily valid beginning with DOS 2.x
 •*For DOS 2.x = 3-byte near jump
 For DOS 3.x = 2-byte short jump + NOP

Notes: OEM name and version are not always present (IBM does not use)

Source: IBM DOS 3.3 Technical Reference, page 2-31

See Also: 1.27. Powers of Two

2.39. DISK PARTITION TABLE LAYOUT

Offset	Length	Name	Contents
0 (0)	byte	Partition status	0=inactive; 80H=bootable, active
1 (1)	byte	Starting head	Binary value
2 (2)	word	Starting sector and cylinder	See note *
4 (4)	byte	Partition type	1=DOS with 12-bit FAT 4=DOS with 16-bit FAT 5=extended DOS 6=reserved for future use DBH=concurrent DOS
5 (5)	byte	Ending head	Binary value
6 (6)	word	Ending sector and cylinder	See note *
8 (8)	dbl word	Starting absolute sector	Binary value (least significant word first and byte swapped in each word)
C (12)	dbl word	Number of sectors	Binary value (least significant word first and byte swapped in each word)

16-byte Block Repeats, as Above, for Each Partition, and Is Followed By:

01FEH	word	Signature	55AAH (indicates valid boot record)
-------	------	-----------	-------------------------------------

Notes: •Some manufacturers allow additional partition types in order to divide large capacity hard disks into several drives.
 •The partition tables begin at an offset of 1BEH in the boot record. The actual boot record is defined by the starting head, cylinder, and sector number, and that sector is loaded to location 7C00H.
 •*Cylinder and sector are stored in bit-position-coded notation. This applies to the starting cylinder and head and the ending cylinder and head. See below.

byte n								byte n+1							
c	c	s	s	s	s	s	s	c	c	c	c	c	c	c	c
M		M					L							L	
S		S					S							S	
B		B					B							B	

The two most significant bits of byte n precede the eight bits of byte n+1 to form the ten-bit cylinder number. The six least significant bits of byte n form the sector number.

Source: IBM DOS 3.3 Technical Reference, pages 9-6 to 9-16

See Also: 2.38. Disk Boot Record Layout

2.42. EXE FILE HEADER

Offset	Length	Usual Contents	Description	Comments
0 (0)	word	4D5AH	EXE file signature	
2 (2)	word		Length of last used sector in file	Modulo 512
4 (4)	word		Size of file, including header	In 512-byte pages
6 (6)	word		Number of relocation table items	
8 (8)	word		Size of header	In 16-byte paragraphs
A (10)	word		Minimum paragraphs needed above program	In 16-byte paragraphs
C (12)	word		Maximum paragraphs desired above program	In 16-byte paragraphs
E (14)	word		Displacement of stack segment in module	Relative to start of program, in paragraphs
10 (16)	word		Contents of SP register at entry	
12 (18)	word		Checksum	Two's complement
14 (20)	word		Contents of IP register at entry	
16 (22)	word		Displacement of code module	Relative to start of program (in paragraphs)
18 (24)	word		Offset to first relocation item in file	Relative to start of file (in bytes)
1A (26)	word		Overlay number	0 for resident part of program
1C (28)	varies		Variable RESERVED space	
varies	varies		Relocation table	
varies	varies		Variable RESERVED space	
varies	varies		Program and data segments	
varies	varies		Stack segment	

Notes: EXE files created for use with Microsoft Windows use a different format (See 6.10. Windows EXE File Format)

Sources: IBM DOS 3.3 Technical Reference, pages 10-3 to 10-6

See Also: 2.43. COM Program Layout
6.10. Windows EXE File Format

2.43. COM PROGRAM LAYOUT

Offset	Length	Description	Comments
0 (0)	256 bytes	Program segment prefix	Values filled in by DOS
100 (256)	varies	Code and data segment	Only one segment allowed
varies	varies	Stack	Usually at top of segment

Notes: The program segment prefix is not usually part of the actual file. It is created and filled in by DOS at program load time. COM files must have code segment ORGed at 100H.

Source: IBM DOS 3.3 Technical Reference, page 7-9

See Also: 2.42. EXE File Header
2.44. COM versus EXE File Differences
3.136. Program Segment Prefix Layout

2.44. COM VERSUS EXE FILE DIFFERENCES

Item	COM Programs	EXE Programs
Max. program size	65278*	No limit
Segment use	One segment only	Multiple segments allowed
Entry point	PSP:0100H	Defined by END Segment
CS at entry	PSP	Segment containing module with entry point
IP at entry	0100H	Offset of entry point within its segment
DS at entry	PSP	PSP
ES at entry	PSP	PSP
SS at entry	PSP	Segment with STACK attribute
SP at entry	0FFFEH or top word, whichever is lower	Size of segment defined with STACK attribute
Stack at entry	Zero word on stack	Initialized or uninitialized
Stack size	65536 - (ProgramSize+256)	Defined in segment with STACK attribute (up to 65536 bytes)
Memory allocation	All free memory allocated to program	May be set to allocate portion of memory (offset 0CH in EXE header)
Subroutine calls	NEAR CALLs only	NEAR or FAR CALLs allowed
Size of file	Exact size of program (might not include PSP)	Size of program plus EXE header (which is multiple of 512 bytes)

Notes: *65536 - 256-byte PSP - 2-byte STACK

Sources: Advanced MS-DOS (Microsoft Press), Ray Duncan

See Also: 2.42. EXE File Header
2.43. COM Program Layout
3.136. Program Segment Prefix Layout

2.45. FONT FILE (CODE PAGE) LAYOUT

Offset	Length	Name	Contents
0 (0)	8 bytes	File tag	FFH followed by "font," followed by three spaces
8 (8)	8 bytes	RESERVED	
10 (16)	word	Number of pointers in header	1 for DOS 3.3
12 (18)	byte	Type of pointer	1 for DOS 3.3
13 (19)	dbl word	Offset to info from start of file	Binary value
17 (23)	word	Number of entries	Binary value
19 (25)	word	Size of code	Binary value
1B (27)	dbl word	Pointer to header of next entry	0000H for last header
1F (31)	word	Device type	1=display, 2=printer
21 (33)	8 bytes	Device name (ID)	ASCII text padded with spaces
29 (41)	word	Code page ID	437
2B (43)	3 words	RESERVED	
31 (49)	dbl word	Pointer to font info	Binary value
35 (53)	word	RESERVED	Must be 1
37 (55)	word	Number of fonts	Binary value
39 (57)	word	Length of font data	Binary value

For Display Font:

3B (59)	byte	Rows in character box	Binary value
3C (60)	byte	Columns in character box	Binary value
3D (61)	2 bytes	Aspect ratio	Currently not used, = 0,0
3F (63)	word	Number of characters in font	Usually 256
41 (65)	varies	Font data	Stored as pixel descriptions

For Printer Font:

3B (59)	word	Printer selection type	1=4201, 2=5202
3D (61)	word	Total bytes in control sequences	Binary value
3F (63)	varies	Hardware code page	Maximum length of 31
varies	varies	Downloadable code page	Maximum length of 31
varies	varies	Downloadable character definitions	See Printer Technical Reference

Version Info: Applies to DOS 3.3 only

Source: IBM DOS 3.3 Technical Reference, pages 7-17 to 7-20

See Also: 3.147. Code Page Assignments

2.46. OPERATING SYSTEM FILES SUMMARY**IBM PC-DOS Version**

File	1	1.1	2	2.1	3	3.1	3.2	3.3
IBMBIO.COM	1920	1920	4608	4736	8964	9564	16369	22100
IBMDOS.COM	6400	6400	17152	17024	27920	27760	28477	30159
COMMAND.COM	3231	4959	17664	17792	22042	23210	23791	25307
Total file sizes	11551	13279	39424	39552	58926	60534	68637	77566

Microsoft MS-DOS Version

File	1	1.1	2	2.1	3	3.1	3.2	3.3
IO.SYS	*	*	*	*	*	*	16138	22357
MS-DOS.SYS	*	*	*	*	*	*	28480	30128
COMMAND.COM	*	*	*	*	*	*	23612	25276
Total file sizes	*	*	*	*	*	*	68230	77761

Notes:

- *The first total shown is for the entire operating system files only. The actual amount of memory used by the operating system is dependent upon the environment size, device drivers that have been loaded, and the settings of the BUFFERS and FILES parameters.
- *MS-DOS released only through OEMs, so file sizes vary.

Source: DOS Disks

See Also: 2.47. Included System Files Summary
2.49. Typical DOS Memory Usage

2.47. INCLUDED COMMAND FILES SUMMARY*

DOS Version Number

Command File	1	1.1	2	2.1	3	3.1	3.2	3.3
APPEND								X
ASSIGN			X	X	X	X	X	X
ATTRIB					X	X	X	X
BACKUP			X	X	X	X	X	X
BASIC	X	X	X	X	X	X	X	
BASICA	X	X	X	X	X	X	X	X
CHCP								X
CHKDSK	X	X	X	X	X	X	X	X
COMP	X	X	X	X	X	X	X	X
DEBUG	X	X	X	X	X	X	X	**
DISKCOMP	X	X	X	X	X	X	X	X
DISKCOPY	X	X	X	X	X	X	X	X
EDLIN	X	X	X	X	X	X	X	X
EXE2BIN			X	X	X	X	X	**
FASTOPEN								X
FDISK			X	X	X	X	X	X
FIND			X	X	X	X	X	X
FORMAT	X	X	X	X	X	X	X	X
GRAFTABL			X	X	X	X	X	X
GRAPHICS			X	X	X	X	X	X
JOIN					X	X	X	X
KEYBUK					X	X	X	X
KEYBGR					X	X	X	X
KEYBFR					X	X	X	X
KEYBIT					X	X	X	X
KEYBSP					X	X	X	X
LABEL					X	X	X	X
LINK	X	X	X	X	X	X	X	**
MODE	X	X	X	X	X	X	X	X
MORE			X	X	X	X	X	X
NLSFUNC								X
PRINT			X	X	X	X	X	X
RECOVER			X	X	X	X	X	X
REPLACE					X	X	X	X
RESTORE			X	X	X	X	X	X
SELECT							X	X
SHARE						X	X	X
SORT			X	X	X	X	X	X
SUBST					X	X	X	X
SYS	X	X	X	X	X	X	X	X
TREE			X	X	X	X	X	X
XCOPY						X	X	X

Notes:

*These COM and EXE files are from the IBM PC-DOS versions. The MS-DOS versions may differ slightly.

**Supplied with Technical Reference manual

(Continued)

INCLUDED COMMANDS (BUILT-IN)

DOS Version Number

Command Name	1	1.1	2	2.1	3	3.1	3.2	3.3
CD/CHDIR			X	X	X	X	X	X
CLS			X	X	X	X	X	X
COPY	X	X	X	X	X	X	X	X
CTTY			X	X	X	X	X	X
DATE	X	X	X	X	X	X	X	X
DEL/ERASE	X	X	X	X	X	X	X	X
DIR	X	X	X	X	X	X	X	X
MD/MKDIR			X	X	X	X	X	X
PATH			X	X	X	X	X	X
PROMPT			X	X	X	X	X	X
REN/RENAME	X	X	X	X	X	X	X	X
RM/RMDIR			X	X	X	X	X	X
SET			X	X	X	X	X	X
TIME	X	X	X	X	X	X	X	X
TYPE	X	X	X	X	X	X	X	X
VER					X	X	X	X
VERIFY			X	X	X	X	X	X
VOL			X	X	X	X	X	X

BATCH FILE COMMANDS (BUILT-IN)

DOS Version Number

Command	1	1.1	2	2.1	3	3.1	3.2	3.3
CALL								X
ECHO	X	X	X	X	X	X	X	X
FOR	X	X	X	X	X	X	X	X
GOTO	X	X	X	X	X	X	X	X
IF	X	X	X	X	X	X	X	X
PAUSE	X	X	X	X	X	X	X	X
REM	X	X	X	X	X	X	X	X
SHIFT	X	X	X	X	X	X	X	X

See Also: 2.46. Operating System Files Summary

2.48. COMMON FILE TYPES (EXTENSIONS)

File Type	Program	Description
\$\$\$	DOS	A "pipe" file created by using the redirection flag () in a DOS command
@@@	CodeviewDisk	
ACT	BITCOM	Communications account data file
ACT	Actor	Source code file for Actor programming language
AIO	APL	APL file transfer format file
AMG	Actor	System image file for Actor programming language
APL	APL	APL work space format file
APP	SQL Windows	Application file
ARF	BASCOM	Automatic response file created by the BM series of compilers; similar to batch files
ARF	FORTTRAN	Automatic response file created by the BM series of compilers; similar to batch files
ARF	COBOL	Automatic response file created by the BM series of compilers; similar to batch files
ASC	Many	ASCII text file; may be typed to the screen
ASM	MASM	Assembly language source code file
AUX	Paradox	
BAK	Many	A backup file; contains a previous version of the information in the file
BAS	BASIC	A file containing BASIC program code; may not be in ASCII format!
BAS	BASICA	A file containing BASIC program code; may not be in ASCII format!
BAS	MS-QuickBASIC	A file containing BASIC program code; may not be in ASCII format!
BAS	Turbo BASIC	A file containing BASIC program code; may not be in ASCII format!
BAT	DOS	Batch file; contains commands to be executed by DOS, in order
BIN	Many	Binary file; often same as an OBJ file; contains 8-bit information (i.e., not ASCII)
BLK	ShowPartner	Block file; contains information about a block manipulated by ShowPartner
BMP	MS-Windows	Bitmap file; contains data for a Windows bitmap structure
C	C compilers	Contains C source code
CAL	SuperCalc	Spreadsheet file; contains contents of a spreadsheet
CCL	Intalk	Communication command language file
CFG	Many	A configuration file; contains information about machine and environment
CHK	CHKDSK	Recovered data file; contains data recovered when using the /F option in CHKDSK
CLR	ShowPartner	Color palette file
CLS	Actor	Class library file for Actor programming language
CMD	dBASE	Command file; used for file that contains dBASE programs
CMD	CP/M-86	Transient command file (similar to DOS EXE and COM files)
CMP	MS-Word	Compare file; contains dictionary of words to compare for spelling
CNF	Many	A configuration file; contains information about machine environment
COB	COBOL	COBOL program source code
COD	FORTTRAN	FORTTRAN program compiled code file
COL	MS-Multiplan	Spreadsheet data file; contains contents of a spreadsheet
COM	DOS	Command (program) file
CRF	MASM	Cross reference file; listing produced by MASM compiler
CRS	World Tour Golf	Course data file
CTX	Microsoft	Course text file; contains information for on-line tutorials
CUR	MS-Windows	Cursor file; contains data for a Windows cursor
DAT	Many	Data file; usually contains ASCII or specifically-formatted data
DB	Paradox	Data file; contains data for a Paradox table
DBD	Bricklin's DEMO	Demonstration data file
DBF	dBASE	Data file; contains data for a dBASE database
DBS	SQL Windows	Data file; contains data for a SQL Windows database
DBT	dBASE	Data file; contains dBASE textual database information
DBT	SQL Windows	Temporary data file
DCT	SpellStar	Dictionary file; contains spelling dictionary
DEF	MS-Windows	Module definition file
DEF	Access	
DES	Access	
DEV	Many	Device driver file; contains code needed by CONFIG.SYS to install a new device
DFM	Palantir Filer	Data entry form file
DGS	PC-DOS	Diagnostics file
DIC	Many	Dictionary file; contains spelling dictionary
DIF	Many	Data interchange format file; used to interchange data between programs
DIR	Sidekick	Directory file; used with dialing options
DIS	Q&A	Startup file used by Q&A
DOC	Many	Document file; may be in ASCII or word processor-specific format
DOC	MS-Word	Document file; contains formatted document in non-ASCII form
DRV	Many	Device driver file; contains information to drive a specific device
DTF	Q&A, PFS	Data file; contains data for a PFS or Q&A database
EMU	BITCOM	Terminal emulation file; contains definitions used to emulate a terminal
EPS	Pagemaker	Encapsulated postscript file; contains condensed postscript printer data
EXE	DOS	Executable program file
FMT	dBASE	Screen format file; contains information about how data is to be displayed on screen
F#	Paradox	Form file; contains form definition information

(Continued)

Table 2.48. Continued

File Type	Program	Description
FNT	Windows	Font file; contains description of what a font should look like
FNT	LaserFonts	Font file; contains description of what a font should look like
FNT	Paintbrush	Font file; contains description of what a font should look like
FON	Windows	GDI loadable font file
FOR	FORTTRAN	FORTTRAN source code file
FRM	dBASE	Report form file; contains information about how a dBASE report should be formatted
GRB	MS-Windows	
GX1	ShowPartner	Graphics screen capture file
H	C compilers	Header file; contains C source code definitions to be merged with other files
HEX	DEBUG	Hex file; contains ASCII only numbers formatted in Intel HEX format
HIN	Access	
HLP	Many	Help file; contains information to help user understand command or function
ICO	MS-Windows	Icon file; contains bit image of an icon
IDX	Q&A	Index file; contains indexing information for a database
IMG	MS-Windows	Hi-res scanned image file
IMP	Pascal	Implementation file for IBM Pascal
INC	Pascal	Include file for Microsoft Pascal
INC	Turbo BASIC	Include file for Borland Turbo BASIC
INI	MS-Windows	Initialization file; contains information about initial state of system
INI	MS-Word	Printer initialization file
INT	Pascal	Interface file for IBM Pascal
INT	Xywrite	Command file for Xywrite
IT	Intalk	Settings file
JOR	SQL Windows	Journal file
KBD	Xywrite	Keyboard configuration file
LAY	Superkey	Layout file; contains keyboard reconfiguration information
LBL	dBASE	Label file
LIB	Many	Library file; normally created by a compiler in one of several standard formats
LNK	MS-Windows/C	
LOD	Many	Load file; used by one copy-protection scheme
LST	MASM	Listing file; lists assembled source code
MAC	Prokey	Keyboard macro file; contains instructions to execute when certain keys are pressed
MAC	Superkey	Keyboard macro file; contains instructions to execute when certain keys are pressed
MAP	LINK	Map file; a list file created by LINK during the linking process
MDM	Access	Modem file; contains information about modems
ME	Many	Usually a READ.ME file containing information about files on disk
MEM	dBASE	Memory file
MNU	Access	Menu file; contains menu definition
MSG	Multiplan	Message file
MSG	Sidekick	Message file; used with appointment calendar
MSP	MS-Windows	Windows Paint file; contains data for a picture drawn with Windows Paint
MOD	MS-Windows	
NDX	dBASE	Index file; contains indexing information for a database
NET	Paradox	Network configuration file
OBJ	LINK	Object code file; contains result of an assembly or compile in a specified format
OVD	Paradox	Overlay file
OVL	Many	Overlay file; contains part of program to be loaded at a later time
OVR	Many	Overlay file; contains part of program to be loaded at a later time
OV#	Many	Overlay file; contains part of program to be loaded at a later time
PAL	Paintbrush	Palette file
PAS	Pascal	Pascal source code file
PCX	Paintbrush	Picture file
PCC	Paintbrush	Cutout picture file

(Continued)

Table 2.48. Continued

File Type	Program	Description
PFM	MS-Windows	Printer font metric file
PGM	Many	Usually a program overlay file
PHB	Access	
PIC	Many	Picture file
PIX	Many	File containing one or more pictures
PIF	MS-Windows	Program information file; used by TopView and Windows to load program into memory
PJ	SuperProject	Project file; contains information about a scheduling project
PRD	MS-Word	Printer definition file; contains information about how to talk to printer
PRF	VisiCalc	Print format file (spreadsheet printed to disk)
PRG	dBASE	Procedure or program file
PRJ	Harvard TPM	Project data file
PRN	Many	Print format file (print to disk)
PRS	MS-Word	
PUB	Pagemaker	Publication file; contains data for page layout
PX	Paradox	Primary Index file
RC	MS-Windows	Resource Script file; contains a list of resource definitions used by MS-Windows
REF	CREF	Printable cross-reference file (see CRF)
R#	Paradox	Report format file; contains a report definition
SC	Paradox	Script file; contains a PAL script (program)
SCN	Microsoft	Screen file; contains screen displays for on-line tutorials
SCP	BITCOM	Script file; contains a macro script for communications session
SCR	Access	Script file
SET	Paradox	Settings file; contains information about settings for a form or table
SOB	Microsoft	Part of on-line tutorials
SOM	Paradox	Sort information file
SPL	SQL Windows	SQLTALK Spooler file
SPS	Mouse	
SQL	SQL Windows	Data file
STY	MS-Word	Style sheet; contains style formatting information
SYM	MS-Windows	Symbolic debugging definitions
SYN	Word Finder	Synonym file; contains information for thesaurus program
SYS	Many	Device driver file; contains information to create a device driver under CONFIG.SYS
TIF	Microsoft	Tagged info file format (see 6.11. Tag Image File Format)
TMP	Many	Temporary file
TPL	Access	
TXT	Many	Text file
VAL	Paradox	Validity check file
VC	VisiCalc	VisiCalc spreadsheet file
WK1	Lotus 1-2-3	1-2-3 spreadsheet file (version 2)
WKS	Lotus 1-2-3	1-2-3 spreadsheet file (version 1)
WMF	MS-Windows	Metafile picture (see 6.16. Metafile Format)
WRI	MS-Windows	Windows Write document file
X#	Paradox	Index file
XLC	MS-Excel	Chart file
XLS	MS-Excel	Spreadsheet file
Y#	Paradox	Index file
Z#	Paradox	Index file

Notes:

- A # sign indicates a position held by a digit, 0-9
- MS-Windows can associate file types with a program. Registration of types is done in the MS-Windows programming SIG on Genie.