Sydex

22DISK

A CP/M-to-DOS Diskette Interchange Utility

22DISK

A CP/M-to-DOS Diskette Interchange Utility

Copyright 1988-91, Sydex
All Rights Reserved

Version 1.34, September 1990

Sydex
P.O. Box 5700
Eugene, OR 97405
Voice: (503) 683-6033
FAX: (503) 683-1622
Data: (503) 683-1385

NOTICE

THIS IS NOT FREE SOFTWARE! If you paid a "public domain" vendor for this program, you paid for the service of copying the program, and not for the program itself. Rest assured that *nothing* ever gets to the originators of this product from such a sale. You may evaluate this product, but if you make use of it, you must register your copy.

We offer several inducements to you for registering. First of all, you receive the most up-to-date copy of the program that we have -- and we do update the product on a regular basis. You also receive support for 22DISK--which can be quite valuable at times. You also receive complete *printed* documentation for the product. A "do-it-yourself" update service is offered to registered users through our own BBS. And finally, we include an evaluation package of some of our other software products. Make no mistake however -- this is a fully functional version of 22DISK and not "crippled" in any way.

REGISTRATION INFORMATION

The non-commercial single-user registration fee for 22DISK is $25.00 US. The commercial and multisystem site fee for 22DISK is $100.00. Users within the United States and Canada should include an additional $2.50 for postage. Users in other countries should include an additional $5.00 for international airmail. *We can accept payment only in U.S. funds.*

Send a check or company P.O. for the appropriate amount to:

SYDEX
P.O. Box 5700
Eugene, OR 97405

We can also accept VISA and MasterCard. Please indicate what product you are ordering and if you have a requirement for 3.5" media. We normally ship 5.25" 360K diskettes, but we will furnish 3.5" 720K media upon request.

If you are a registered user of this product and desire an update, please send $5.00 for handling and we will ship you an up-to-date copy. Please add $10.00 for printed documentation to cover duplication costs.

DISTRIBUTION NOTICE

This is "user-supported" software. You are hereby granted a license by Sydex to distribute this evaluation copy of 22DISK and its documentation, subject to the following conditions:

1.   22DISK may be distributed freely without charge in evaluation form only.

2.   22DISK may not be sold, licensed, or a fee charged for its use. If a fee is charged in connection with 22DISK, it must cover the cost of copying or dissemination only. Such charges must be clearly identified as such by the originating party. Under no circumstances may the purchaser be given the impression that he is buying 22DISK itself.

3.   22DISK must be presented as a complete unit, including this documentation. Neither 22DISK nor its documentation may be amended or altered in any way.

4.   By granting you the right to distribute the evaluation form of 22DISK, you do not become the owner of 22DISK in any form.

     Any other use, distribution or representation of 22DISK is expressly forbidden without the written consent of Sydex.

1

---

1. *CP/M, CP/M-80 and CP/M 2.2 are trademarks of Digital Research. MS-DOS is a trademark of Microsoft. IBM PC, PC-XT and PC-AT are trademarks of International Business Machines. V20, V30, uPD70108, uPD70116 are trademarks of NEC Electronics. Some products of Lear-Siegler, Kaypro, Osborne and Televideo are also trademarked.*

Table of Contents

22DISK, Briefly

22DISK is a product which enables one to convert files on diskette from various CP/M formats to MS-DOS, and vice versa. It is a companion product to 22NICE, a Z-80/8080 CP/M 2.2 emulator. The operation of one part is not dependent on the other--if you want to use 22DISK to convert data from other disk formats to DOS or vice-versa, you need not install the 22NICE CP/M emulator. Similarly, if you already have CP/M programs available as DOS files, you may not want to use the 22DISK disk conversion utilities.

Our packaging for 22NICE and 22DISK offers the 22DISK program set separately for those who may be using a different emulator or have no need of a CP/M emulator. The 22NICE emulation package, however, is bundled with 22DISK because of the need to transport CP/M files to the PC before emulation can be performed.

The 22DISK disk conversion package offers the following features:

*   "Roll your own" disk formats - we provide support for several common CP/M 2.2 diskette formats, but if you have a format that isn't in our list, you can simply define the characteristics in an ASCII parameter file. Registered users get an even larger file of diskette types.

*   Support for single- double- and quad-density formats, hardware permitting.

*   Support of the the following operations:

        CP/M-to-DOS
        DOS-to-CP/M
        Directory display
        Display CP/M files
        Erase CP/M files
        Format CP/M diskette

*   Full user number support.

*   Support of MicroSolutions' CompatiCard.

22DISK currently will *not* handle "hard sectored" diskettes, nor will it work with diskettes written on an Apple][or Commodore 64 or 128.

Installing 22DISK

22DISK will not run without first being installed. To do this, enter the following command at the DOS prompt:

> **A:22DINST**   and press the ENTER key.

We're assuming that the file *22DINST.EXE* is located on the diskette in drive **A:**; if you are installing **22DISK** from some other drive or directory, substitute it for the **A:** in this command.

Generally, the installation process is self-explanatory. A few of the more important points to note are:

1.   **22DISK** will support up to 8 diskette drives--4 per controller card. Support for two controller cards is provided.

2.   If you're using **22DISK** on an XT-class computer with a high-density controller, you'll need to specify that your computer is an AT-type computer. This is necessary to ensure that your controller is correctly programmed by **22DISK**.

3.   **22DINST** installs all files in the same directory. However, the file *CPMDISKS.DEF* and the files with an extension of "*LBL*" may be placed in any directory specified by the DOS **PATH** environment variable. For information on **PATH**, consult the DOS reference manual.

4.   The files comprising **22DISK** can be installed or re-installed "in place". That is, the source drive and directory may be the same as the destination.

You'll need about 512K of base memory to install **22DISK**. If you don't have that much available, you can still install **22DISK** by creating an installation diskette having all of the **22DISK** files *except CPMDISKS.DEF* and all of the files having a extension of "*LBL*". These files can be copied over to your destination drive or directory *after* you've run **22DINST**.

Getting Started

The following sections deal with various specific features of **22DISK**. If the diskette format desired is one of those included in the standard definition file, entering the command

CMENU

will cause a menu-driven interface to **22DISK** to be executed. The appropriate function may then be selected from the menu. **CMENU** displays the following choices:

```
22DISK Version 1.32 (12-Oct-89)

Please select one of the following:

    0.  Exit to DOS
    1.  Set CP/M diskette type
    2.  Set CP/M diskette drive
    3.  Copy CP/M file(s) to DOS
    4.  Copy DOS file(s) to CP/M
    5.  Format a CP/M diskette
    6.  Display a CP/M directory
    7.  Display (type) CP/M file(s)
    8.  Erase CP/M file(s)
```

No diskette type selected yet.

Please enter function number (0...8):

When **CMENU** is used for the first time, choice 1 should be selected to determine what CP/M diskette format will be used. Until a format is selected, no CP/M diskette operations can be performed; attempts to perform any operation before selecting a type will produce an error message.

When choice 1 is selected, a list of CP/M diskette types, in display "pages" of 23 types will be displayed. The cursor keys may be used to select a format type. When the desired type is located, pressing the **ENTER** keys will select that format. A screen of general information about systems requirements will be displayed, after which the opening menu is again shown.

After the diskette type is selected, **CMENU** will "remember" it from session to session. The next step is to select choice **2** to specify what diskette drive will be used to access CP/M diskettes.

The desired function may then be selected from the menu. Each function may then request additional input for items such as CP/M diskette type and names of source and destination files.

CMENU prompts for parameters and formats them into a command line for one of **22DISK**'s utility programs. If a menu interface is not desired, each program may also be invoked as a normal DOS command. The following sections deal with each specific function in terms of the program that performs that function.

Selecting A Disk Format

All of the **22DISK** utilities make use of a common interface to specify the format of CP/M diskette to be used. Over 200 different formats are provided with **22DISK** and "custom" definitions may also be written.

Each different CP/M format is given a unique one- to four-character *tag*. For example, the Osborne 1, single-sided, double-density format is given a tag of "OSB2", while the single-sided, double-density Kaypro 2 is assigned a tag of "KAY1". So, to specify a format, its tag is used.

The format tag may be indicated to the **22DISK** utilities in any of three ways:

(1)   The tag may be specified interactively. This is the default situation if the tag has not been indicated by either of the other two ways. The following prompting message is displayed -

        Enter 1-4 character disk format type or ENTER alone for list -

        If the **ENTER** key is pressed, a list of diskette formats and their tags is displayed, 23 lines at a time. The cursor keys (up, down, PgUp and PgDn) may be used to select a type. Pressing **ENTER** causes the current type to be used.

(2)   The tag may be specified directly on the DOS command line. In this case, the tag is the first field of the **22DISK** command and is preceded with a slash (/) character. For example,

        CFMT /KAY1 A:

would result in the diskette in drive A: being formatted as a Kaypro 2 single-sided, double-density diskette.

(3)   The DOS environment variable "CPMDISK" may be set to the tag corresponding to the desired format. So,

SET CPMDISK=OSB2

would result in all **22DISK** operations being performed in terms of Osborne 1 double-density format, unless overridden by an explicit DOS command line specification as in (2), above.

The file named **CPMDISKS.DEF** contains definitions for all diskette types and must be present somewhere in the DOS command search path. That is, one of the directories indicated by the DOS **PATH** statement must contain the **CPMDISKS.DEF** file.

In addition to the predefined formats given in **CPMDISKS.DEF** as supplied with **22DISK**, custom definitions may also be prepared using any ASCII text editor. For details, consult the section titled *Building Custom Diskette Definitions*.

22DISK CP/M File Naming Conventions

All of the **22DISK** utilities use the same convention for naming CP/M files, which corresponds roughly to that used by CP/M. However, to supply a means of indicating CP/M user numbers and to afford an alternate way to refer to diskette drives, the convention has been extended somewhat.

A CP/M file designator as used by **22DISK** consists of the following items:

*   A disk drive specifier. This is a letter referring to those drives configured by the **22DINST** program.

*   An optional user number between 0 and 31. If omitted, user number 0 is assumed. A wildcard, or asterisk (*), may be specified to signify all user numbers.

*   A colon (:).

*   The file name consisting of up to 8 letters, numbers or special characters. Upper- and lower-case letters are considered to be equivalent. A special character may be any of the following -

$$! @ \# \$ \% \wedge \& () - \_ = + \{ \} [ ] ' " , /$$

*   The characters "?" and "*" serve as "wildcard" specifiers; "?" matches any single character. "*" matches any remaining characters in the file name.

*   A period (.), which may be omitted if no file extension is present (see next item).

*   The file extension which consists of up to 3 letters, numbers or special characters. Its rules of formation are identical to those associated with the file name, including use of wildcard characters.

Here are some examples of CP/M file names accepted by the **22DISK** conversion utilities:

A:*.COM	All files with an extension of "COM" from drive A:, user number 0.
D*:*.*	All files from any user number on diskette drive **D**.
B2:X*.DAT	All files having an extension of "DAT" under user number 2, from drive B:, starting with the letter "X".

In all of the **22DISK** utilities, a drive designator followed by a colon without any filename or extension assumes all files on that drive. Thus,

B:	is identical in meaning to
B:*.*	

A Quick Reference Guide to 22DISK Commands

The following is a short description of the command syntax of each conversion program. The *disk type* field can be omitted if the **CPMDISK** environment variable has been set to the appropriate disk type "tag".

Convert CP/M Files to DOS

Command syntax:

> **CTOD** */disk-type CP/M-file-specifier DOS-disk-name*
> **CTOD** */disk-type CP/M-file-specifier DOS-disk-and-path*
> **CTOD** */disk-type CP/M-file-name DOS-path-and-file-name*

Examples:

> **CTOD /OSB2 A3:*.* C:**
> **CTOD /KAY1 A:*.DAT C:\MYDIR**
> **CTOD A1:TEXT.DOC C:\DOCS\MINE.TXT**

Convert DOS Files to CP/M

Command syntax:

> **DTOC** */disk-type DOS-file-specifier CP/M-disk-unit-and-user*
> **DTOC** */disk-type DOS-file-name CP/M-file-name*

Examples:

> **DTOC C:*.BAS A:**
> **DTOC C:MARY.BAS B:TEST.BAS**

Display a CP/M Diskette Directory

Command syntax:

> **CDIR** */disk-type CP/M-disk-unit*
> **CDIR** */disk-type CP/M-file-specifier*

Examples:

> **CDIR A:**
> **CDIR A:*.ASM**

Erase a CP/M Diskette File

Command syntax:

CERA */disk-type CP/M-file-specifier*

Example:

CERA /IBM1 WS?.COM

Display (Type) a CP/M Diskette File

Command syntax:

CTYPE */disk-type CP/M-file-specifier*

Example:

CTYPE B2:*.DOC

Format a CP/M Diskette

Command syntax:

CFMT */disk-type CP/M-disk-unit*

Example:

CFMT /OSB1 B:

Generate an Indexed Definition File

Command syntax:

GENINDEX *source-file destination-file*

Example:

GENINDEX CPMDISKS.NOI CPMDISKS.DEF

Strip an Index From a Definition File

Command syntax:

> **STRIPIDX** *source-file destination-file*

Example:

> **STRIPIDX CPMDISKS.DEF CPMDISKS.NOI**

CTOD - Convert A CP/M Diskette To DOS

The program **CTOD** is used to read a CP/M diskette and copy files to any selected DOS directory.

The syntax of the **CTOD** command is -

> **CTOD** */disk-type CP/M-file DOS-path*

Disk-type is an optional CP/M format type designator. *CP/M-file* refers to the CP/M file specifier for the diskette drive containing the CP/M diskette and the file(s) to be copied. The *wildcard* characters * and *?* are used to specify "wildcard" positions in the CP/M file name.

DOS path refers to the name of a DOS disk drive or subdirectory to which the CP/M files will be copied by **CTOD**. If this term is omitted, the current DOS default drive and path are used.

Some transformations of CP/M file names are made by **CTOD** when copying to DOS. CP/M file naming conventions allow just about any character in a file name; DOS naming conventions are much more restrictive. Characters which are not accepted by DOS are translated by **CTOD** to the character #, a legal DOS file name character.

In addition, the file extension *COM* is changed to *CPM* by **CTOD**. DOS considers any file with the extension *COM* to be executable; if execution of a CP/M file is attempted by DOS, the result could be disastrous.

The **22NICE** CP/M emulator also performs the same name translation internally for CP/M programs, so programs will operate correctly with the transformed file extensions.

Here are a few examples of **CTOD** commands:

CTOD A:*.*

Copy all files from user 0, on drive A: to the current directory. Use whatever format that was defined by the **CPMDISK** environment variable, or prompt for one if **CPMDISK** is not defined.

CTOD /A1 B*:*.COM C:\CPMFILES

Copy all files with an extension of *COM* from all users, from drive B: to the directory named **C:\CPMFILES**. Use the 8" SSSD format. These files, when copied, will be given an extension of *CPM*.

CTOD

Performs no action, but gives a synopsis of the **CTOD** command syntax.

DTOC -- Convert DOS Files TO CP/M Files

The program **DTOC** is the complement of **CTOD**; that is, it writes DOS files to a CP/M diskette. The syntax of the **DTOC** command is:

DTOC /*disk-type DOS-file-name CP/M-drive-and-user*

The optional *disk-type* is the name of a CP/M diskette type, as discussed earlier. *DOS-file-name* refers to the name of a DOS file or group of files and may include drive and subdirectory specifiers. *CP/M-drive-and-user* specifies the diskette drive to receive the DOS files and the user number under which they are to be created.

In addition to copying DOS files to CP/M diskettes, **DTOC** carries out the *CPM* to *COM* name translation described in the **CTOD** command. That is, all DOS files with an extension of *CPM* will be created as *COM* files on the CP/M target diskette.

Some examples of **DTOC** commands are:

DTOC C:*.* A:

Copy all files from DOS drive C: to CP/M diskette A:, user 0. Use whatever format that was defined by the **CPMDISK** environment variable, or prompt for one if **CPMDISK** is undefined.

DTOC /A1 \UTILS*.ASM B3:

Copy all files from the DOS default drive in the subdirectory named *UTILS* with an extension of *ASM* to user 3 on CP/M diskette drive B:. Use the 8" SSSD format.

DTOC

Performs no action, but gives a synopsis of the **DTOC** command syntax.

CTYPE -- Display A CP/M File

The program **CTYPE** copies a CP/M diskette file or files to DOS standard output. Before each file is copied, its name is also copied to standard output. **CTYPE** is the analogue of the DOS and CP/M TYPE utilities.

The syntax of the **CTYPE** command is:

CTYPE */disk-type CP/M-file*

Disk-type is the optional diskette type code as discussed earlier. *CP/M-file* refers to the CP/M file designator for the diskette drive containing the CP/M diskette and the file(s) to be displayed. The "wildcard" characters * and ? may be used to specify that a group of CP/M files are to be displayed.

Some examples of the **CTYPE** command are:

CTYPE A:*.*

Copy all files from user 0, on drive A: to standard output.

CTYPE /A1 B*:*.DOC

Copy all files with an extension of *DOC* from all user numbers, from drive B: to standard output. The source diskette is assumed to be in 8" SSSD format.

CTYPE

Performs no action, but gives a synopsis of the **CTYPE** command syntax.

CDIR -- Display A CP/M Directory

The program **CDIR** displays a list of CP/M diskette files in much the same way that the CP/M command *DIR* does. However, the user number for each file is also displayed, along with the available CP/M diskette space remaining.

The syntax of the **CDIR** command is:

> **CDIR** */disk-type CP/M-file*

Disk-type is the optional CP/M diskette type code as discussed earlier. *CP/M file* refers to the CP/M file designator for the diskette drive containing the CP/M diskette and the file(s) to be displayed. The "wildcard" characters "*" and "?" may be used to specify that a group of CP/M file names are to be shown. Some examples of the **CDIR** command are:

> **CDIR A:*.***

> Displays a directory of all files from all user areas, on drive A.

> **CDIR /A1 B*:*.DOC**

> Displays a directory of all files with an extension of *DOC* from all user numbers, on drive B:. The source diskette is assumed to be in 8" SSSD format.

> **CDIR**

> Performs no action, but gives a synopsis of the **CDIR** command syntax.

CERA -- Erase CP/M Files

The program **CERA** erases a list of CP/M diskette files in much the same way that the CP/M command *ERA* does. In addition, the available remaining CP/M diskette space is reported.

The general syntax of the **CERA** command is:

> **CERA** */disk-type CP/M-file*

Disk-type is the optional diskette tag as discussed earlier. *CP/M-file* represents the names of the CP/M files to be erased. The "wildcard" characters "*" and "?" can be use to erase a group of CP/M files.

Each file specified is erased by writing the hexadecimal character *E5* as the first byte (user number) of the CP/M diskette directory entry. The rest of the affected directory entry is left alone. Examples of the **CERA** command are:

CERA A:*.*

Erase all files belonging to user 0, on drive A:.

CERA /A1 B*:*.DOC

Erase all files with an extension of *DOC* from all user numbers, on drive B:. The source diskette is assumed to be in 8" SSSD format.

CERA

Performs no action, but gives a synopsis of the **CERA** command syntax.

CFMT -- Format A CP/M Diskette

The **CFMT** utility formats blank CP/M diskettes. **CFMT** corresponds directly to the CP/M FORMAT or NEWDISK command.

The syntax of the **CFMT** command is:

CFMT */disk-type CP/M-disk-name*

Disk-type is optionally used to specify the CP/M diskette type as discussed earlier. *CP/M-disk-name* refers to the diskette drive containing the diskette to be formatted.

Before a diskette is formatted by **CFMT**, a prompt is issued to the console in order to confirm that a diskette is to be reformatted. If the letter "G" followed by the **ENTER** key is typed, the diskette if formatted. Otherwise, **CFMT** exits without formatting.

Examples of the **CFMT** command are:

CFMT /MOR1 A:

Formats the diskette in drive A: with a format pattern suitable for a Morrow MD1 computer system.

CFMT F:

Formats the diskette in diskette unit F: with a format pattern corresponding to whatever is specified by the **CPMDISK** environment variable. If **CPMDISK** is not defined, **CFMT** will prompt for a diskette type.

Building Custom Diskette Definitions

As mentioned earlier, the file **CPMDISKS.DEF** contains definitions for various common CP/M diskette types. This sections describes how you can add your own definitions to this file for diskette formats not already present. This section requires some knowledge of CP/M disk structure; it is not intended for the casual user.

The following is a typical diskette format definition:

```
BEGIN
 SUP3  Superbrain QD -- DSDD 48 tpi 5.25"
DENSITY
 MFM,LOW
COMPLEMENT
CYLINDERS
 35
SIDES
 2
SECTORS
 10,512
SIDE1 0
 1,3,5,7,9,2,4,6,8,10
SIDE2 1
 1,3,5,7,9,2,4,6,8,10
ORDER
 CYLINDERS
BSH 4 BLM 15  EXM 1  DSM 169  DRM 63
AL0 080H  AL1 00H  OFS 2
END
```

Diskette definitions are keyword-oriented. That is, there are a few words which have special meaning to the routines with interpret the diskette definitions. Each keyword identifies the data which immediately follows it.

The following rules of construction apply to a diskette definition:

(1)   Definitions are not sensitive to alphabetic case; terms and keywords may be in upper- or lower-case characters, or a mixture of both.

(2)   Spaces, tabs and line boundaries are treated as separators between terms. In some cases, a comma is required as an additional separator.

(3)   A term must be completely contained on a single line; otherwise, line boundaries are generally ignored. The only exceptions to this are the **NOTE** keyword, which causes the rest of the current line to be ignored, and the **BEGIN** keyword, which uses the rest of the line following the diskette type code or "*tag*" as a descriptive comment.

(4)   Numbers may be expressed in decimal, hexadecimal or binary. Decimal quantities are the default and are made up of digits between 0 and 9 inclusive. Hexadecimal quantities may also include the letters A through F, but must begin with a number. Hexadecimal quantities are denoted by a final "H". Binary quantities are made up of the numbers 0 and 1 and are denoted by a final "B". These are the same conventions as those used in CP/M assembler coding.

(5)   All-blank or empty lines are ignored and may be used freely to enhance readability.

BEGIN signals the start of a definition. The first term following this keyword is a one- to four-character diskette tag code by which this definition may be accessed. The first character of this type code must be alphabetic; the second character may be numeric or alphabetic. The remainder of the the line following the type code is saved as commentary describing the diskette type.

DENSITY takes two terms, separated by a comma. The first term is either **MFM** for a double-density diskette or **FM** for single density. Your disk controller must be capable of supporting **FM** operation if the **FM** density option is given. The second term is used to program the diskette clock rate on AT-class machines. **LOW** signifies that a 250 Kbps data rate should be used; **HIGH** signifies that a 500 Kbps (HD density) data rate is to be used. When the PC-AT 1.2M 5.25" diskette drive is being used, a data rate of **LOW** is taken to mean 300 Kbps instead of 250 Kbps.

CYLINDERS specifies the number of cylinders contained in the diskette format. If this number is less than 50, quad-density (80 track) drives will be *double stepped* to go from cylinder to cylinder.

SKEW specifies the *physical* interleaving of sectors. This specification is optional; if omitted a 1-to-1 physical interleave is assumed.

SIDES specifies the number of tracks or surfaces per cylinder. **SIDES** may be either 1 or 2 for single- or double-sided media, respectively.

SECTORS specifies both the number of sectors per track (cylinder surface) and the size of a sector. The sector count and the size terms must be separated by a comma. **SECTORS** must precede **SIDE1** and **SIDE2** in a definition.

COMPLEMENT is used to denote certain formats which record data in complemented or *NOT-ed* form. The primary offender in this area is the Superbrain line. **COMPLEMENT** instructs **22DISK** to complement data before writing and after reading a CP/M diskette. If **COMPLEMENT** is not used within a definition, normal *true* format recording is assumed.

ORDER specifies the ordering of data on a double-sided diskette. There are several possible values:

> **SIDES** specifies that first one side, then the other side are written or read before the cylinder number is advanced.

> **CYLINDERS** specifies that all tracks on the first side are recorded, followed by all tracks on the second side, beginning with the highest or innermost track on the second side. That is, tracks 0, 1, 2... are first recorded on side 0, then tracks 39, 38, 37... on the second side (for a 40 cylinder diskette).

> **EAGLE** is similar to **CYLINDERS** except that the second side is recorded from the lowest or outermost track. Thus, the order would be 0, 1, 2... on the first side, then 0, 1, 2... on the second side.

> **COLUMBIA** specifies a **CYLINDERS**-type ordering except that the track numbering on the second side *continues* in sequence from the first side. Thus, the first side might contain tracks 0 to 39, and the second side contain tracks 40 through 79.

if **EVEN-ODD** is specified, the first side of a diskette contains even-numbered tracks; the second side, odd-numbered.

SIDE1, which must come after the **SECTORS** specification, specifies the sector ordering on the first cylinder surface. CP/M 2.2 allows a *software interleave* of sectors on a diskette; the terms given here reflect that interleave. The first term after the **SIDE1** keyword is the logical side ID found in each ID address mark for that disk. Usually, this number is 0 for the first side of a diskette and 1 for the second side. Certain computers, however, deviate from this, as shown by the Kaypro example above, where the second side also has a side ID of 0. After the side ID, there follows a list of sector numbers in their CP/M ordering. Each num-

ber specifies the next logical sector to be addressed. Sector numbers in this list must be separated by commas; the number given must correspond to that stated in the **SECTORS** specification.

SIDE2 is specified only if the diskette has two surfaces per cylinder. The terms following **SIDE2** are the same as those for **SIDE1**, except that they define the sector addressing and ordering for the second surface of the diskette.

If **SIDE2** is declared before **SIDE1** in a definition, the diskette will be treated as if it were "upside down". That is, accessing will begin with the *second* diskette side.

LABEL is specified for diskette formatting only. The term following the word **LABEL** is used as a file name with an implied suffix of **.LBL**. When a diskette is formatted, the contents of this file are written to the diskette, starting with the first sector. If the data length of the file does not work out to be an integral multiple of the diskette sector length, the last sector is filled to its end with the value of the last byte read from the file. The **LABEL** feature may also be used to enable one to create CP/M "system" diskettes.

If the keyword **SKIP** is included in a format definition, **CFMT** will not format "system" tracks--the tracks specified by the **OFS** parameter. A mixed-density format may be created by writing two definitions for a diskette--the first for the system tracks, the second for the data area.

SEE may be used to specify that a diskette definition is the same as another already defined. If used, **SEE** must be the only parameter in a definition. For example, the Casio FP-1100 diskettes have the same format as those used for the Fujitsu Micro-8. The Casio definition may then be encoded simply as:

> **BEGIN CAS1 Casio FP-1100 - DSDD 48 tpi 5.25"**
> **SEE FUJ2**

Note that no **END** statement is used. The **SEE** keyword may be used only in an indexed definition file.

The remainder of the keywords in the definition are taken from the CP/M Disk Parameter Block for this diskette type. The *CP/M System Alteration Guide* presents details on the meaning of each field. A brief description of each is presented here:

> **BSH** refers to the *block shift* or the number of left shifts needed to translate an allocation block number to a relative sector address.

> **BLM** refers to the *block mask* or a bit mask of ones corresponding to the number of bits specified by **BSH**. For example, if **BSH** is 3, **BLM** is 7 (3 bits).

EXM refers to the *extent mask* or a bit mask of ones used in delimiting extents, or groups of 128 128-byte records.

DSM refers to the total number of sectors present on a diskette. This quantity, like all of the CP/M system variables, assumes a 128-byte sector.

DRM refers to the total number of directory entries available on a diskette. **DRM** is also related to the **AL0** and **AL1** fields.

AL0 and **AL1** form a bit mask, beginning with the most significant bit in byte **AL0** and ending with the least significant bit in **AL1**. These two bytes map the first 16 allocation blocks of the disk. A *1* bit in a position indicates that an allocation block is reserved, usually for the directory (see **DRM**, above).

OFS indicates the number of tracks that are used for storing the CP/M system and BIOS at the beginning of a diskette.

SOFS indicates the number of sectors that are used for storing the CP/M system and BIOS at the beginning of a diskette. **SOFS** is used when the number of sectors used by the CP/M is not an integral number of tracks, such as for the Coleco Adam.

Every diskette definition must end with the **END** keyword, which follows the OFS specification.

If the following Microsoft BASIC program is run on the host CP/M (8 bit) system, the values of the relevant DPB fields are displayed for the currently logged-in drive:

```
100 REM PROGRAM TO DISPLAY A DEFAULT DPB
110 REM
120 REM FOR CP/M 2.2 ONLY
130 REM
140 DEF FNP(X) = PEEK(X) + 256*PEEK(X+1)
150 IF PEEK(0) <> 195 THEN PRINT "Requires CP/M" : STOP
160 PRINT "DPB VALUES FOR THE CURRENT DISK -- " : PRINT
170 REM
180 REM STUFF S$ WITH THE FOLLOWING PROGRAM -
190 REM
200 REM PUSH H          ; Save address of FAC-3
210 REM LHLD 1          ; Get BIOS Cold boot entry
220 REM MVI L,27        ; Change to SELDISK vector
230 REM LDA 4           ; Get current disk
240 REM MOV C,A
250 REM MVI A,PCHL      ; for indirect call
```

```
260 REM STA 38H            ; set in RST 7 vector
270 REM RST 7              ; call SELDISK
280 REM POP D              ; FAC-3
290 REM XCHG
300 REM MOV M,E            ; Store pointer to DPH
310 REM INX H
320 REM MOV M,D
330 REM RET                ; exit...
340 REM
350 S$ = ""
360 READ SD : IF SD < 0 THEN 380   ' Read the USR code
370 S$ = S$ + CHR$(SD) : GOTO 360
380 SX = VARPTR(S$) +1
390 SR = FNP(SX)
400 DEF USR = SR                   ' User program set up
410 DPH = USR(0)                   ' Get DPH Pointer
420 IF DPH = 0 THEN PRINT "Select error" : STOP
430 SPT = FNP( FNP( DPH +10)) ' Get Sectors/Track
440 PRINT "Sectors per Track";TAB(30);SPT
450 IF FNP(DPH) = 0 THEN 510   ' if no interleave
460 PRINT " Sectors interleaved as follows:"
470 FOR I = 1 TO SPT
480   PRINT TAB(10); PEEK( FNP(DPH) + I-1)
490 NEXT I
500 PRINT : GOTO 520
510 PRINT " Sectors are not interleaved"
520 DPB = FNP(DPH+10)
530 PRINT "Block Shift (BSH)";TAB(30);PEEK(DPB+2)
540 PRINT "Block Mask (BLM)";TAB(30);PEEK(DPB+3)
550 PRINT "Extent Mask (EXM)";TAB(30);PEEK(DPB+4)
560 PRINT "Total Sectors (DSM)";TAB(30);FNP(DPB+5)
570 PRINT "Directory Entries (DRM)";TAB(30);FNP(DPB+7)
580 PRINT "Allocation 0 (AL0)";TAB(31);HEX$(PEEK(DPB+9));"H"
590 PRINT "Allocation 1 (AL1)";TAB(31);HEX$(PEEK(DPB+10));"H"
600 PRINT "Cylinder Offset (OFS)";TAB(30);PEEK(DPB+13)
610 PRINT
620 STOP
630 DATA   229,42,1,0,46,27,58,4,0,79,62,233,50,56,0
640 DATA   255,209,235,115,35,114,201,-1
650 END
```

Some Hints on Developing a Definition

Quite frequently, the only information available when developing a definition for **22DISK** is
a diskette with a few files and nothing more. Here's a guide on what to do if you'd like to try
working up a definition yourself:

1.   Get a copy of Sydex's **ANADISK**. This utility will furnish far more infor-
mation than will any other program.

2. Run **ANADISK** on the diskette in **SECTOR EDIT** mode. Note that the interleave (or SKEW) is given, as well as the number, size and addresses of sectors. You then have the figures for **DENSITY, CYLINDERS, SIDES, SKEW SIDE1** and **SIDE2**.

3. Locate the directory on the diskette. Almost invariably, it starts on the first sector of a track. The number of tracks up the point where the directory begins will give the figure for **OFS**. The directory stands out because it shows file names every 32 bytes, or every other line on the **ANADISK** display.

4. Examine the first directory entry. The second 16 bytes (the second line of the entry) gives a list of allocation unit numbers belonging to the file. The first one of these will tell how many allocation units have been reserved to hold the directory. Thus, if the first allocation byte is 04, four allocation units have been allocated to the directory. Another thing to note is whether the allocation unit number appears to be an 8 bit or 16 bit quantity. For example, if an entry has the following form:

```
00 50 49 50 20 20 20 20 20 43 4F 4D 00 00 00 10   .PIP     COM....
04 00 05 00 06 00 00 00 00 00 00 00 00 00 00 00   ................
```

the allocation unit designators are 4, 5 and 6 and are 16-bit quantities. On the other hand, the following item:

```
00 50 49 50 20 20 20 20 20 43 4F 4D 00 00 00 10   .PIP     COM....
04 05 06 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

shows the same allocation units, but as 8-bit quantities. If 16-bit quantities are involved, more than 255 allocation units are defined for the diskette, if an 8-bit quantity, less than 256 units. This serves as a check for later computations. In any case, the first allocation number of the first directory entry gives the number of bits to be set in **AL0** and **AL1**. Thus, for an allocation number of 04:

AL0 11110000B Al1 0

5. Starting with the first directory sector, advance the sector (and track, if necessary) until you find the data belonging to the first file. Unused directory entries are almost always filled out with a pattern of E5's. The number of sectors dedicated to the directory will give an indication of both the number of directory entries and the allocation unit size. Sup-

pose that a our hypothetical diskette shows 16 sectors of 512 bytes allocated to the directory. Since a directory entry is 32 bytes long, the number of directory entries possible for this format is:

16 entries/sector X 16 sectors = 256 entries total

Thus, **DRM** is one less than this or 255. Since we know that these 16 sectors represent 4 allocation units or 8,192 bytes, the allocation unit size is:

8,192 / 4 = 2,048 bytes

Given this, we can use the following table to compute the values for **BSH** and **BLM**:

Unit Size	BSH	BLM
1,024	3	7
2,048	4	15	← This one
4,096	5	31
8,192	6	63
16,384	7	127

7. We still need a value for **EXM** and **DSM**. **DSM** is simply the total number of allocation units on the diskette, not counting the system **OFS** area. Let's assume that our diskette has an offset of 2 tracks and is single-sided with 40 cylinders with 8 sectors per track. The value of DSM is then:

((40 cylinders X 1 side) - 2 tracks) X 8 sectors X 512 bytes) = 155,648 bytes
155,648 / 2,048 = 76 Allocation units, total

To get **EXM**, we then use the following table:

Unit Size	DSM < 256	DSM > 255
1,024	0	na
2,048	1	0
4,096	3	1
8,192	7	3
16,384	15	7

8. That's almost the complete picture, but for the **SIDE1** and **SIDE2** sector ordering. This is the hardest to determine of any of the parameters. Find a readable text file on the diskette and piece the text back together by observing how words break across sector boundaries. After a bit, a pattern in the ordering will become apparent, such as:

    1, 3, 5, 7, 2, 4, 6, 8

9. The **ORDER** value can be determined on double-sided diskettes by observing whether file data appears to move from one track to the next on the same side (**CYLINDERS**), or appears to go from one side to the other and then advance to the next cylinder (**SIDES**).

10. **COMPLEMENT** is rare. Generally, if a diskette appears to be formatted with values of *1A* instead of *E5*, the data on the diskette is complemented.

Indexing A Large Definition File with GENINDEX

CPMDISKS.DEF as supplied with **22DISK** has been indexed so that the entire file does not need to be searched for the specified diskette definition. **CPMDISKS.DEF** is also supplied in a non-indexed form as **CPMDISKS.NOI**.

To generate an indexed definition file, the **GENINDEX** command is used as follows:

GENINDEX *source-file destination-file*

where *source-file* is the non-indexed input definition file and *destination-file* names the file which is to contain the indexed definitions.

An editable form of **CPMDISKS.DEF** may be had by using the utility **STRIPIDX** to remove the index from a definition file:

STRIPIDX *source-file destination-file*

Diskette Densities and Recording Modes

The diskette has undergone quite a number of changes in its development and **22DISK** reflects some of these changes. This section discusses some of the terms used in connection with **22DISK** definitions:

1. The physical size of the diskette. By and large, most diskettes come in one of three sizes: 8, *5.25* or 3.5". The 8" diskette is rare to find in use nowadays, but **22DISK** is able to handle this format, given the appropriate hardware. 5.25" diskettes are the reigning media choice now, but may soon be displaced by the 3.5" "micro" diskettes. There have been other sizes, such as 3.25", but these have not enjoyed much popularity.

2. Recording performed on both surfaces of the diskette. The formats using only one surface are called single-sided, those using both surfaces, double-sided. Double-sided drives were rare until about 1978; now they are the rule, rather than the exception.

3. The track density. An analogue to this would be to ask how close the grooves on a phonograph record are. On 8" and 5.25" drives, a track packing density of 48 tracks per inch is most commonly used, although some high-capacity 5.25" drives double this for 96 tracks per inch (the PC AT's 1.2 megabyte drive is a 96 tpi drive). There was an alternate 5.25" standard in use between 1977 and 1980, 100 tracks per inch, but it was always in the minority of applications. 3.5" diskettes use a track density of 135.5 tracks per inch.

4. The spindle speed, or how fast the diskette spins inside its jacket. All 8" diskettes spin at 360 revolutions per minute; almost all 3.5" diskettes spin at 300 rpm; the first Sony drives spun at 600 rpm. Most 5.25" drives also spin at 300 rpm, with the exception of the PC AT 1.2 megabyte high-capacity drive, which spins at 360 rpm.

5. The recording method used. Early diskette drives used a method called FM, for frequency modulation, in which one pulse per time period signified a zero bit and two pulses signified a one bit. This is the method referred to as *single density*. Later, a recording method called MFM, for Modified FM came into popular use. This scheme uses a somewhat more complex encoding scheme dependent on the position of a pulse within a bit time *window* and the value of the last bit written. What MFM does is get twice the data rate for the same bit rate, hence the term *double density*. Almost all modern diskette drives use MFM recording.

6. The bit rate at which data is read or written. The data rate for low-density 5.25" diskettes are recorded is 250,000 bits per second. The PS/2 and PC AT high-density formats make use of a data rate of 500,000 bits per second. 8" formats generally make use of a 500,000 bit per second data rate. The PC AT 1.2 megabyte high-capacity drive uses a data rate of 300,000 bits per second to record 360K diskettes because of its higher spindle speed:

$$\frac{360 \text{ rpm}}{300 \text{ rpm}} \text{ X } 250{,}000 \text{ bps} = 300{,}000 \text{ bps}$$

7. The sector demarcation method used. All modern diskettes use unique data patterns to show where a sector begins and ends on a track. On some older systems, however, additional index holes were used to mark sector boundaries--known as "hard" sectoring. 22DISK does not support hard sectored formats.

The common PC XT diskette controller is able to handle single- and double-sided, double-density diskettes. The common PC AT diskette controller possesses a variable-rate bit clock generator for 250,000, 300,000 and 500,000 bits per second to handle double density diskettes in both 8", 5.25" and 3.5" sizes, as well as high-density diskettes in the 5.25" and 3.5" profiles.

If there is a need to handle single-density diskettes in a 5.25" form factor, a PC XT *clone* controller can easily be modified to do so, at no loss of normal functioning. The next section shows how.

Single-Density Diskette Controllers

In the past couple of years, a new large-scale integrated circuit has made its appearance in several products. The IC in question is Western Digital's 37C65 integrated diskette controller. This chip not only supports all of the "normal" PC high- and low-density modes, but also supports single-density!

We've found this chip in the Sysgen "OmniBridge" adapter, as well as the WD1002-F002 diskette adapter. Further, it appears on quite a number of AT-style hard/floppy controller combinations put out by Western Digital. All of these are "normal" products at normal prices.

Using 8" Drives with 22DISK

We've had quite a number of inquiries about operating 8" diskette drives with 22DISK. While not all 8" drives have the same connector pinout,, we'll consider the most common, that of the Shugart 800-series here. This particular connection scheme is known to include the following drive families:

1.   Siemens 100/200
2.   Qume 840
3.   Innovex 200/400
4.   Memorex 550

Note that if you decide to construct an external 8" drive setup yourself, you'll have to provide a power supply which furnishes both +5 and +24 volts. In addition, some older drives also require a small -5 volt supply. The simplest way to construct a cable which will connect the drive to a DC-37 male connector which will plug into the external drive socket attached to the mounting bracket of a suitable diskette adapter card is as follows:

1.   Using a 50 conductor ribbon cable, crimp a 50 pin card edge connector to one end.

2.   Use the following chart to pair up the appropriate ribbon cable conductors with the DC 37 pins. Note that the DC 37 numbers its pins from 1 to 19 on the first row and from 20 through 37 on the second row, while the ribbon cable is numbered consecutively from one side (pin 1) to the other (pin 50). With a bit of care you should be able to make use of a DC 37 ribbon cable connector instead of the solder-cup variety:

DC 37 pin	1 - 5 No Connection	DC 37 pin	13 Ribbon 38 (WDATA)
6	Ribbon 20 (INDEX)	14 "	40 (WGATE)
7	" 18 (HD LD)	15 "	42 (TK00)
8	" 28 (DS1)	16 "	44 (WPROT)
9	" 26 (DS0)	17 "	46 (RDATA)
10	No Connection	18 "	14 (HDSEL)
11	Ribbon 34 (DIR)	19 No Connection	
12	" 36 (STEP)	20-37 Ribbon 1,3,5... (GND)	

Set the drive select jumper on the drive for DS0 (the first select). The drive will be addressable as physical unit 2; you can select any drive letter you wish when configuring 22DISK. Note that a diskette controller capable of high-density operation must be used for 8" drives, such as the Western Digital WD 1002A-FOX, Sysgen Omnibridge or MicroSolutions CompatiCard.

This is clearly not a project for a beginner. The cable itself takes some patience to construct, but usually can be built in less than one hour. This arrangement is in use here at Sydex with a Qume 842 drive with excellent results.

Using Cromemco CDOS Diskettes with 22DISK

We've found that while **22DISK** generally will not be able to read Cromemco CDOS diskettes formatted on a Cromemco computer, a diskette that has first been formatted by the Cromemco, then formatted by **22DISK** will work well on both **22DISK** and the Cromemco. In short, format your diskettes with the Cromemco, then with **22DISK**, then copy the files you need to transfer with the Cromemco.

If the original Cromemco equipment is not available, a diskette adapter may be modified to enable the reading of Cromemco diskettes. Contact Sydex for further details.

In Closing

Thank you for taking the time to consider Sydex. We hope that **22DISK** does its job well and that it meets your needs. This product represents several months of work and research; if you have any suggestions on how it might be improved, please let us know.

The Sydex Product Line

AnaDisk -- The compleat diskette utility. Nothing like it anywhere else; scan, edit, repair and copy just about any kind of diskette. $25.00 ($150.00 site) registration fee.

Con>Format -- Concurrent "background" diskette formatter. Features "pop-up" operation and "hot key" activation. You've got to see it to believe. Supports all current DOS formats. $15.00 ($50.00 site) registration fee.

COPYQM -- Mass diskette duplicator. Format, copy and verify multiple diskettes from a single master. Implements "no keyboard" interaction mode and drive "round robin" servicing. Supports all standard DOS formats. $15.00 ($50.00 site) registration fee.

FORMATQM -- Mass diskette formatter - format a box of diskettes at a single sitting. Implements "no keyboard" interaction mode and drive "round robin" servicing. Supports all standard DOS formats. $10.00 ($40.00 site) registration fee.

22DISK -- Transfer files, format, examine and erase files on "foreign" CP/M diskettes on your PC. Includes tips on supporting 8" and 5.25" single-density diskettes. Contains definitions for over 270 different formats. $25.00 ($100.00 site) registration fee.

22NICE -- A CP/M 2.2 emulation package. Supports the NEC V-series chips or performs emulation by software for both the 8080 and Z80 processors. Includes terminal emulation and diskette handling for common CP/M systems. Includes 22DISK. $40.00 ($150.00 site) registration fee.

TeleDisk -- Our diskette "FAX machine". Turn any diskette into a compressed data file and vice-versa. Allows you to send and receive entire diskettes via modem. Even works with some "copy-protected" diskettes. $150.00 commercial license *only*.

Information on any of these products can be obtained from Sydex by calling or writing us at:

Sydex
P.O. Box 5700
Eugene, OR 97405
Voice: (503) 683-6033
FAX: (503) 683-1622
Data: (503) 683-1385