

CDOS USER'S MANUAL

CROMEMCO INCORPORATED

2400 Charleston Road, Mountain View, California

Copyright 1977

## TABLE OF CONTENTS

| <u>Section</u> |                                                   | <u>Page</u> |
|----------------|---------------------------------------------------|-------------|
| 1              | Introduction . . . . .                            | 2           |
| 2              | CDOS System Structure . . . . .                   | 3           |
|                | 2.1 System Organization                           |             |
|                | 2.2 File Organization                             |             |
|                | 2.3 File-Naming Conventions                       |             |
|                | 2.3.1 Device Names                                |             |
|                | 2.3.2 Disk-File Names                             |             |
|                | 2.4 Command Structure                             |             |
| 3              | CDOS Operations . . . . .                         | 11          |
|                | 3.1 System Start-up                               |             |
|                | 3.1.1 "Cold" Bootstrap                            |             |
|                | 3.1.2 Drive Selection and "Warm" Start            |             |
|                | 3.2 Control Functions                             |             |
|                | 3.2.1 Console Control Characters                  |             |
|                | 3.2.2 Printer Control Functions                   |             |
| 4              | CDOS Commands . . . . .                           | 17          |
|                | 4.1 Command Syntax                                |             |
|                | 4.2 CDOS Intrinsic Commands                       |             |
|                | BYE, DIR, ERA, REN, SAVE, TYPE                    |             |
|                | 4.3 CDOS Extrinsic Command Programs               |             |
|                | @(Batch), DUMP, EDIT, INIT, STAT, WRTSYS,<br>XFER |             |
| 5              | System Failure. . . . .                           | 28          |
|                | 5.1 Error Messages from the Diskette              |             |
|                | 5.2 Precautions Concerning Diskettes              |             |
| 6              | Command Summary . . . . .                         | 31          |

## SECTION 1

INTRODUCTION

CDOS is the Cromemco Disk Operating System. CDOS is designed to allow users of Cromemco microcomputer systems to create and manipulate disk files using symbolic names.

CDOS, or a subset of CDOS, is provided on diskette with each of the Cromemco disk software packages. The following disk software packages (with their model numbers) have now been announced by Cromemco:

|                | 5" Diskette | 8" Diskette |
|----------------|-------------|-------------|
| FORTTRAN IV    | FDF-S       | FDF-L       |
| 16K BASIC      | FDB-S       | FDB-L       |
| Z-80 ASSEMBLER | FDA-S       | FDA-L       |
| Dazzler Games  | FDG-S       | FDG-L       |

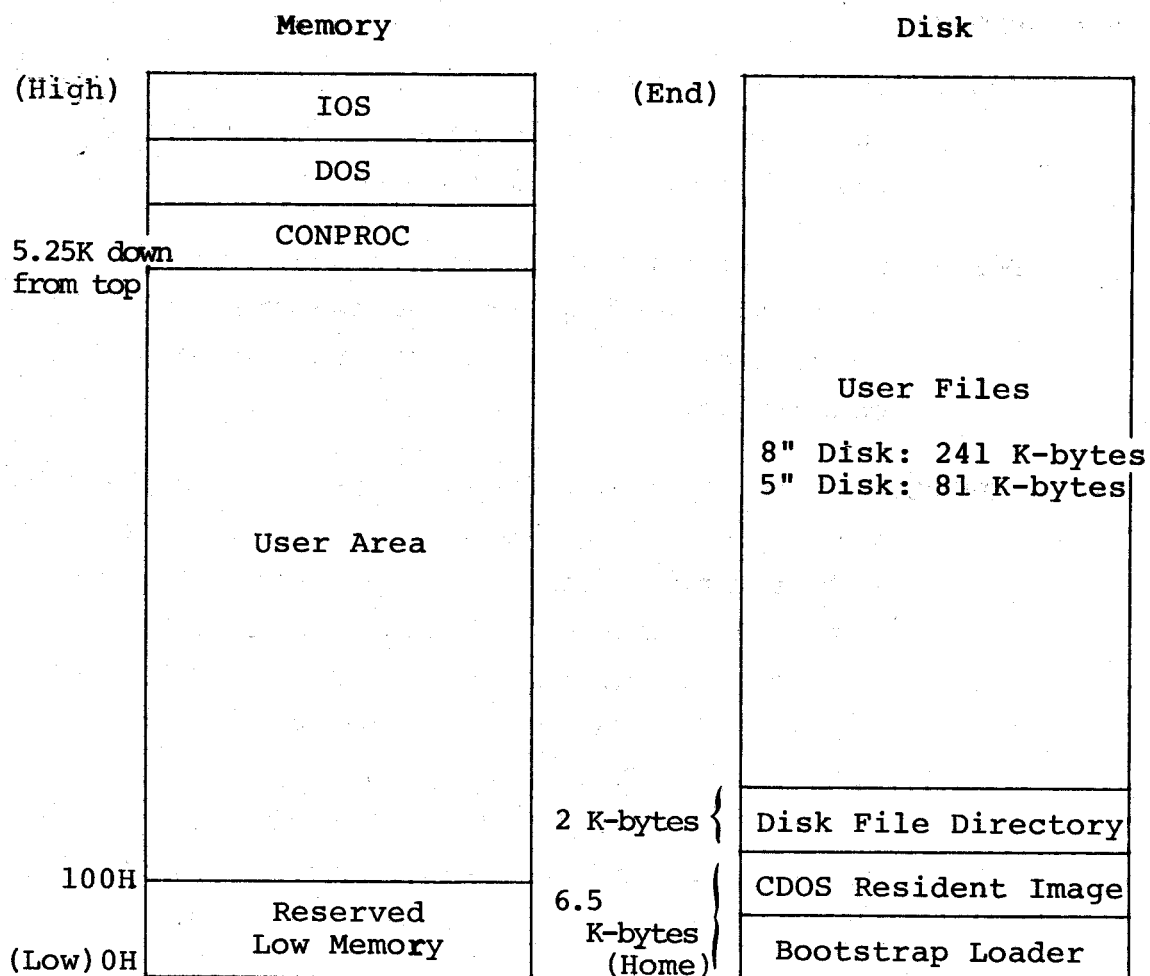
This manual describes how to use CDOS and the CDOS commands. The CDOS system structure and memory allocation are described in Section 2. A step-by-step description of the use of CDOS is given in Section 3. A complete description of CDOS commands can be found in Section 4. Error messages (should you ever get any) are described in Section 5. Finally a summary of the CDOS commands is given in Section 6 with cross-references to page numbers in this manual.

## SECTION 2

CDOS SYSTEM STRUCTURE

## 2.1 System Organization

CDOS is an operating system that is loaded from diskette during the "cold" boot sequence (which is initiated by resetting the computer). The system resides in high memory above any user programs. During initial system configuration, the memory location of CDOS is fixed, and the bootstrap loader is set-up to load CDOS properly. However, by special use of low memory (0-100H), all user programs call CDOS through a standard sequence which is transparent to the size of memory. The addition of more memory will expand the available user area.

CDOS Memory Use Map

As shown in the memory map, the system consists of the following modules:

|            |                                 |
|------------|---------------------------------|
| In memory: | IOS - I/O System                |
|            | DOS - Disk Operating System     |
|            | CONPROC - Console Processor     |
|            | User Area - User Program Region |
|            | Low Memory - Reserved Space     |
| On disk:   | Bootstrap Loader                |
|            | CDOS Resident Image             |
|            | Disk File Directory             |
|            | User File Storage Region        |

These functions work together to handle all standard disk and device I/O for the user. They are described next:

#### RAM MEMORY:

IOS - Input/Output System contains the basic input/output functions for the devices (console, printer, punch, reader) and the disk I/O driver.

DOS - Disk Operating System is responsible for managing, creating, opening, reading and writing disk files, as well as calling user programs or editing console input.

CONPROC - The CONSOLE PROCessor acts as the normal user interface. It calls a user program based on a simple command line typed on the console and passes parameters from the command to the program. It has some internal functions called intrinsic commands; these are explained elsewhere (TYPE, DIRectory, ERase, etc.)

The external functions are the extrinsic commands which cause execution of a disk file corresponding to the command name. EDIT, STAT, and

XFER are examples. These programs are first loaded into the user area from the disk and then executed.

User Area - This is where programs are actually run. It begins at 100H and extends to the system stack below the Console Processor. All system external commands and user command programs run here. Bootstrapping and intrinsic CONPROC functions do not alter the User Area.

Low Memory - The first 256 bytes in memory (0H to 0FFH) are reserved for the following special purposes by the system:

- 0H - System "warm" start vector
- 5H - System call vector for user requests
- 8-40H - Reserved for interrupt vectors
- 40H-5BH - Reserved for system
- 5CH-7FH - Standard user file control blocks
- 80H-0FFH - Standard user I/O buffer

Low Memory Use Map

DISK:

Bootstrap Loader - This loads the system initially from the disk.

CDOS Resident Image - This is a copy of CDOS which is loaded when the disk is first started up. It is created by the user during initial system configuration and written onto the disk by WRTSYS.

Disk File Directory - This section of the disk is used by CDOS to keep track of the files stored on the disk. The CONPROC allows the user to list the contents of the directory. One entry is required for each file section of up to 16K bytes, and a maximum of 64 entries or "file extents" are

allowed (one file may include more than one extent and thus counts as more than one entry).

User Files - The rest of the disk is used for storage of program, text, or data files.

## 2.2 File Organization

A file is a string of bytes which is usually stored on a medium such as a disk or paper tape and is given a filename. This string of bytes may form either a program, a list of data, or a body of text; but in all cases, the filename is used as a means of referring to or calling the string. Files may be coded in binary or the ASCII character set.

Files may be read from or written to a number of devices. The devices available under CDOS are:

Console Input & Output

Printer Output

Reader Input

Punch Output

Disk Input & Output

As normally delivered, only the console, printer, and disk I/O are active.

The primary use of CDOS is to perform I/O with the disk. Up to four 8" floppy disk drives and three 5" floppy drives may be connected for a total not to exceed four drives. Unlike some large computer systems, all disk files under CDOS may be accessed in either random or sequential order.

Devices are pre-defined to CDOS, but disk files are dynamically created, extended, or deleted as required. As will be described in the next section, symbolic names are often used to describe files.

## 2.3 File-Naming Conventions

The following symbolic names may be used when referring to devices or disk files (square brackets refer to optional quantities).

### 2.3.1 Device Names

The following is a table of standard device names having the format:

3 character name (XXX), colon, optional number 0 thru 7 (#)  
 XXX:[#]                      Examples: CON: or PUN:1

| <u>Device</u> | <u>Name</u> | <u>Number Range</u> |                         |
|---------------|-------------|---------------------|-------------------------|
| Console       | CON:        | 0...7               |                         |
| Card Reader   | RDR:        | 0...3               |                         |
| Paper-tape    |             |                     |                         |
| Punch         | PUN:        | 0...3               |                         |
| Line Printer  | LST:        | 0,1                 | Either term may be used |
|               | PRT:        | 0,1                 |                         |
| Dummy         | DUM:        | —                   | Bit Bucket or EOF       |

Be sure to use a ":" to separate the device name and number.

### 2.3.2 Disk File Names

Disk files have more complex names specifying both the disk drive and the filename plus extension. Also, for certain cases, a generic or general name may be used which will match all similar names in the directory. These two types are described below:

#### (1) Unambiguous File References (ufr):

The format is as follows:

[X:] filename[.ext]



where [X:] is an optional disk drive specifier (A:, B:, C:, or D:); if this prefix is omitted, the system defaults to the current drive.

<filename> is a required 1 to 8 character filename.

[ext] is an optional 1 to 3 character extension to the filename.

(Note that the square brackets do not actually appear in the above but are used only to indicate optional quantities.)

Examples:     FILE        or        A:FILE.ASM

- (a) A filename or extension may include any printable ASCII character except the following:

\$ \* ? = / . , : <space> -

- (b) Although lower case characters are accepted by some user programs, all system functions convert lower to upper case.
- (c) Note the use of the "." to separate the filename and extension for disk file references.

**Filename Extensions:** There are several standard types of filename extensions expected by system programs. These are listed below:

|      |                                  |
|------|----------------------------------|
| .Z80 | Z-80 Assembler source file       |
| .BAK | Editor back-up file              |
| .BAS | BASIC source file                |
| .CMD | Batch command file               |
| .COM | Executable command program       |
| .FOR | FORTRAN source file              |
| .LIB | Source library                   |
| .PRN | Printer or listing file          |
| .REL | Relocatable module (object file) |
| .SYS | System image file                |

## (2) Ambiguous File References (afr):

There are two symbols which may be used when referring to files in a general way; these are used as abbreviations to the regular complete filename. Note that they do NOT become parts of the filename but merely substitute for them. These two characters are the asterisk "\*" and the question mark "?". The "\*" substitutes for anything to the right of it up to the next delimiter, "." or  $\langle CR \rangle$ , and the "?" substitutes for any single character in its exact position. These are best described by a few illustrations:

Suppose you had just created the FORTRAN IV file "EXAMPLE .FOR"; following compilation you wish to see what additional files were created. You might issue the command:

```
DIR EXAM*.*
```

and the results would be displayed on the console:

```
EXAMPLE .FOR      (sourcefile)
EXAMPLE .REL      (object file)
EXAMPLE .PRN      (listing file)
EXAMS   .TXT      (other sourcefile)
```

Alternatively you might have typed:

```
DIR EXAM?.*
```

and the results would be:

```
EXAMS   .TXT
```

where one "?" is needed for every replaced letter.

## 2.4 Command Structure

A CDOS extrinsic command calls a utility program stored on the disk, loads it into memory beginning at 100H, and begins execution. A user may also create programs which may be called by a CDOS command. Both these types of files (user-generated and utility programs supplied with CDOS) must have a .COM extension to the filename. The third type

are the intrinsic commands, which are internal to CDOS and are not stored in regular disk files. There are seven of these which the CONPROC recognizes: ATTRIB, BOOT, DIR, ERA, REN, SAVE, and TYPE.

A .COM file will be passed the user command line by the console processor although it may not use all of it. A command line also does not necessarily use CDOS. A few examples will illustrate this; if the user types the command line:

```
STAT THISFILE
```

the entire command line is passed on, but "THISFILE" is ignored by the STAT utility file. On the other hand, if the user types:

```
EDIT THISFILE
```

a file called "THISFILE" will be created by EDIT, and the console will issue the Editor prompt.

## SECTION 3

CDOS OPERATION

## 3.1 System Start-up

## 3.1.1 "Cold" Bootstrap

The following is a description of starting up CDOS from power-off. However, first it would be desirable to briefly describe the bootstrap options available using the 4FDC Disk Controller. There are four switches on the 4FDC card which are used to set operation.

Switch 1 is the RDOS (PROM resident disk operating system) DISABLE switch. When ON, the PROM containing RDOS cannot be accessed. When OFF, the PROM resides from C000H to C3FFH in address space.

Switch 2 is the RDOS DISABLE AFTER BOOT switch. When ON, RDOS will automatically be disabled from address space following CDOS boot. When OFF, RDOS remains in memory at C000H following CDOS boot.

Switch 3 is the BOOT ENABLE switch. When ON, CDOS bootstrap is executed from power-on or a computer reset. When OFF, RDOS comes up under console control.

Switch 4 is the INITIALIZATION INHIBIT switch. When ON, diskettes cannot be initialized under software control. When OFF, disks may be initialized.

These switches are described in more detail in the 4FDC Disk Controller manual. To start with, however, switches 1, 2, and 3 should be set to OFF, and switch 4 should be set ON. Now, after all circuit boards are installed in the computer and the terminal is connected, then perform the following steps:

- 1) Turn power on to the computer and terminal.
- 2) Press carriage RETURN key up to four times to set the console baud rate; RDOS will prompt with a ";".

- 3) Place the CDOS system disk supplied with this manual in disk drive A.
- 4) Type "B <CR>" and CDOS will boot up and show the system prompt "A.".

The system is now running. Pressing "reset" on the back of the computer at any time will return you to RDOS, from which you can again boot CDOS using the "B", boot command.

If you would prefer to have CDOS boot up during power-on, then set switch 3 on the 4FDC card to the ON position. Then do steps 1 and 3 above followed by pressing carriage return up to four times. It is advisable to insert the disks after power-up; however, the disks may be left in the drives when pressing "reset", which will then reboot CDOS.

Either of the above procedures is known as a "cold" bootstrap which includes read-in of the I/O routines from disk.

Note: The "cold" boot does not alter the contents of the user area. Therefore, if program failure occurs, the user area may be saved (see SAVE) and examined following reboot. The user area begins at the 100H and extends to just below CDOS. To determine this address:

- 1) Set switch 3 on the 4FDC to the off position.
- 2) Boot up CDOS and then type "BYE",
- 3) Use RDOS to look at locations 6 and 7 for the low address of CDOS. For a 32K CDOS, this address is 6B00H; thus, the top of the user area is 6AFFH.

### 3.1.2 Drive Selection and "Warm" Start

When a command is issued, the current disk drive is always referred to unless another drive is specified in the filename. Control can be transferred from one drive to another by issuing the disk specifier followed by a colon and carriage return (to terminate). Suppose that drive A is currently being used and it is desired to switch control drive to B. The user would type:

B:<CR>

and the console would display: "B.", the prompt for drive B. When a drive is first selected in this way, the diskette is logged in, so the system knows what free space is on the diskette. However, whenever a diskette is changed, the new one must be logged-in by typing ↑C (CTRL-C). This is essential or else the old disk will still be logged-in and the system will improperly write the new disk directory. Note that the ↑C is needed only when changing diskettes and not when specifying another drive. Issuing a CTRL-C is known as a "warm" start of the system which means it logs-in the disk. This means CDOS reads in the disk directory and builds a bit map showing which sectors may be written to.

## 3.2 Control Functions

### 3.2.1 Console Control Characters

Once the system is running, basic user interaction is with the console processor CONPROC. This program takes care of file maintenance by means of commands internal to CDOS ("intrinsic" commands) or by disk commands and programs external to CDOS ("extrinsic" commands). The prompt given by the system is the disk identifier for the current drive, e.g., "A." for drive A. A command may be typed any time the prompt is displayed.

While typing a command, the standard buffer input mode is active and certain special control characters are usable. A control character is displayed on the console: ↑l (up-arrow, followed by the letter). To type a control character, press the CTRL key first and hold it (like a shift key); then type the letter. The following control characters are allowed:

↑E - Physical ~~CR~~LF only -

Go to next line without terminating.

BACKspace (↑H), Underscore ( \_ ), RUBout, or DElete keys -

Delete current character without echo -

These will backspace the cursor on a CRT terminal but will have no effect on a hardcopy terminal.

↑M or RETURN key - Terminate a command line.

↑R - Retype current line after many corrections.

↑S - Pause during device I/O -

This is primarily used to start and stop a listing on the console. Any key may be typed to resume processing, but only ↑S is used to pause; pressing other keys will abort the listing.

↑U - Delete the current line.

↑X - Delete character with character echo -

This deletes and echoes the character following 3 backslashes; 3 forward-slashes are generated by resuming typing. This control may be used with hard-copy terminals.

### 3.2.2 Printer Control Functions

There are three additional control-characters used for handling output to the printer:

↑P - Send all console output to the printer -

This is a toggle action; if off, the printer is turned on, and vice versa. Do NOT press ↑P if the printer is not connected or not enabled. If any character is typed before a second ↑P, the computer will go into a wait state and CDOS will have to be rebooted.

↑T - Turn off all output to the printer (can be output by user program).

↑W - Send all output to the printer as well as console (can be output by user program).

Cromemco line printers may be interfaced to the computer using the TUART I/O board. CDOS, the Cromemco operating system, recognizes the printer at port 54H. The DIP switch in the lower right-hand corner of the TUART is used to set port assignments. It must be set for 5 to correspond to the most significant digit of this port number.

The line printer must be connected to one of the TUART's parallel ports. To connect the line printer to parallel B of the TUART, J3, set DIP switches 7 and 9 off, 8 and 10 on. For parallel A (J2) set 3 and 5 off, 4 and 6 on. If both these ports are set at 5, then port A has precedence and the printer must be connected to port A. Also, DIP switch 2 must be set to the OFF position to disable the Address-Reverse feature for ports A & B. Switch 1 may be set to either position, Z-80 or 8080 Interrupts. Note that neither port may be set at 0 (ie., you may not have DIP switches 3, 4, 5, 6 or switches 7, 8, 9, 10 all ON simultaneously) when you are using the TUART in conjunction with a 4FDC Controller, because the 4FDC makes use of port 0. Thus, it is wise to set the other port at some high value like 8. A typical set-up for connecting the printer to port B would be: DIP switches 1,2,6,7, and 9 OFF and all others ON.

The final step is to connect the printer to the TUART board using a 12-conductor cable wired with the following plug connections:

(See chart on the following page)



## Cromemco 3700 Series Printers

Printer Interface Connector  
(Amphenol #57-30360)EIA Male Connector  
(25 Conductor)

| <u>Description</u>                         | <u>Pin</u> | <u>Pin</u>                    | <u>Description</u>                                  |
|--------------------------------------------|------------|-------------------------------|-----------------------------------------------------|
| Data Strobe                                | 1          | 22                            | Bit 7, Out                                          |
| Data 1                                     | 2          | 13                            | Bit 0, Out                                          |
| Data 2                                     | 3          | 25                            | Bit 1, Out                                          |
| Data 3                                     | 4          | 12                            | Bit 2, Out                                          |
| Data 4                                     | 5          | 24                            | Bit 3, Out                                          |
| Data 5                                     | 6          | 11                            | Bit 4, Out                                          |
| Data 6                                     | 7          | 23                            | Bit 5, Out                                          |
| Data 7                                     | 8          | 10                            | Bit 6, Out                                          |
| Data 8----- (not used for ASCII data)----- |            |                               |                                                     |
| <u>ACKNLG</u>                              | 10         | 15                            | <u>SENSA</u> or <u>SENSB</u><br>(depending on port) |
| BUSY                                       | 11         | 17                            | Bit 5, In                                           |
| <u>+0v.</u> (signal gnd.)                  | 16         | 14                            | Signal Gnd.                                         |
| Chassis Gnd.                               | 17         | (connect to computer chassis) |                                                     |

The EIA male connector will plug into a female connector mounted on the back of the computer. The female connector is part of a standard cable (Cromemco TUART cable) which connects to any of the ports on the TUART. For use with the printer, plug the TUART cable into TUART J2 or J3, Parallel A or Parallel B Port.

Refer to the printer manual for appropriate shield pin numbers if extra long cables to the printer are desired. Note in the above that all eight Out bits are used; bit 7 is used as a Data Strobe; bits 0-6 carry the ASCII data. The printer indicates if it's busy on Input line 5. The ACKNLG line is for use with interrupt-based systems or software.

## SECTION 4

CDOS COMMANDS

## 4.1 Command Syntax

A request for a particular action is made to CDOS by typing the command into the console. CONPROC then processes it and determines whether the command is internal to itself (intrinsic) or located on the disk (extrinsic). Any unrecognized command is assumed to be on the disk, and CONPROC attempts to locate a file of that name with the .COM extension. The current disk is searched first, and if that fails, the drive-A disk is searched as a Master Library. If the program is found, it is called into memory and the remainder of the command line is passed to it as control information (Sec. 2.4).

A command line starts with the command name; a disk specifier is allowed (Sec. 2.3) but no filename extension is allowed (a .COM extension is assumed). The contents of the rest of the line depend on the command, but the following conventions are observed:

- (a) All options are preceded by a slash "/".
- (b) The following form is always used for assignment functions:  
Destination-Filename=Source-Filename
- (c) Commas, equal sign, or blanks are the delimiters which may separate filenames.
- (d) Blanks are ignored except as delimiters for separation of filenames.
- (e) All letters in command lines are translated into upper case; thus, all filenames are upper case.

The following abbreviations are used throughout the next two sub-sections:

ufr - Unambiguous File Reference, ie., the exact filename must be specified.

afr - Ambiguous File Reference, ie., "?" and "\*" are allowed in the filename; a ufr is allowed wherever an afr is indicated.

[ ] - Optional quantity or expression.

<> - Used to distinguish an enclosed expression.

<CR> - Carriage Return.

␣ - stands for at least one blank space as a delimiter.

All CDOS commands are terminated by a <CR>.

## 4.2 CDOS Intrinsic Commands

The intrinsic commands are built-in to CONPROC and reside in memory following a system boot. All files referred to by intrinsic commands are disk files.

BYE

The BYE command forces a return to RDOS. Once in RDOS you may return to CDOS by typing B (for Boot). Note that the BYE command should not be used if switch 2 on the 4FDC is ON as this disables the RDOS ROM after the CDOS boot.

DIR [afr]

Lists disk files giving size (in K-bytes) and number of extents, followed by a summary of the total disk space (K-bytes and extents) used by all files which were listed. Pressing any key will abort listing. ↑S will cause the listing to pause. If no filename is given, "\*. \*" is assumed.

ERA afr

Erase all files whose names match that given. Note that more than one file may be specified by issuing a command using the "\*" or "?" characters. For example, ERA EXAMPLE.\*

will erase the first 3 files of the example in Section 2.3.2(2). (ERA \*.\* will clear the disk.)

REN afr=afr   Rename <new file name> = <old file name>  
If an afr is used, a search is made for all occurrences of the new name. However, during renaming no test is made, and when the non-ambiguous characters in the new name replace those of the old, a duplicate directory entry may be formed. For example,

REN \*.Z80=\*.ASM is okay, but

REN TEST.ASM=\*.ASM is bad because several files might receive the name TEST.ASM.

SAVE ufr n   Save n (decimal) pages of 256 bytes each from the User Area onto disk. Since no system functions modify the User Area, it can be saved even after a reboot. This is valuable in cases of program failure when the contents of the User Area would otherwise be lost. If the ufr specified is named a .COM file, it could become a user command file. To determine the number of pages to be saved, convert to decimal the 2 high digits of the highest address to be saved. For example, to save up to 0BFFH in a user command file, type:

SAVE FILENAME.COM 11

TYPE afr   The first file found in the directory to match the filename given will be printed in ASCII on the console. Note that binary files cannot be TYPed, but they can be DUMPed. Tabs will be expanded to the

appropriate number of spaces. ↑S will cause the listing to pause at any time; pressing any other key (except to restart listing after a ↑S) will abort the listing.

#### 4.3 CDOS Extrinsic Command Programs

The following command programs are supplied on the master disk included with CDOS. They will be found as .COM files. The command for calling each of them (except Batch) is the same as the program name shown below. The ufr used by command programs may be disk filenames or device names (Sec. 2.3). Square brackets below indicate optional quantities.

##### @ (Batch)

The Batch command may be used to perform an entire list of CDOS commands by issuing only one. Its format is one of the following:

```
@<CR>                                (local mode)
@%ufr%[Parameter-1%Parameter-2%...Parameter-9]<CR>
                                     (file mode)
```

where the % stands for at least one space as a delimiter.

Batch takes its commands sequentially from a file containing all the commands to be executed. In local mode this file is created just prior to execution and disappears afterward. That is to say, local mode Batch uses the commands only once. This is useful if you would like to issue a long string of commands and then leave the computer for some time. An example will best illustrate local mode. The user might type the following sequence into the console; note that the "A." is the CDOS prompt and the "!" is the Batch local mode prompt:

```
A.@<CR>                                (indicates local)
!DIR<CR>                                (types the directory)
```

```

!TYPE THISFILE      (types the file named "THISFILE")
!REN THATFILE=THISFILE (renames "THISFILE")
!                (null line terminates Batch)

```

Following the null line, Batch immediately begins execution of the three commands issued, giving the command line for each one just prior to execution.

Batch file mode allows the user to create a file containing the desired command stream and then to execute this file as often as desired. This is useful for copying CDOS programs to another disk, for example. The file may be created using the Text Editor and it must have a .CMD extension to be found by Batch. The following example will illustrate this further; suppose the user creates (using the Editor) a file called COPY.CMD containing the following:

```

XFER B:@.COM=A:@.COM
XFER B:DUMP.COM=A:DUMP.COM
XFER B:EDIT.COM=A:EDIT.COM
XFER B:INIT.COM=A:INIT.COM
XFER B:STAT.COM=A:STAT.COM
XFER B:WRTSYS.COM=A:WRTSYS.COM
XFER B:XFER.COM=A:XFER.COM

```

Then the user inserts a new diskette containing only CDOS into drive-B and types: A.@COPY<CR>.

The system then proceeds to copy the seven command programs explained here onto the disk in drive-B. This procedure can be implemented as many times as desired.

Parameters may also be passed to the .CMD file via the Batch command; this is an optional feature. Parameters are inserted wherever ↑0, ↑1, ... ↑9 appear(s) in the .CMD file. (Note that these are NOT control-characters, but rather are the two separate characters, up-arrow, "↑" followed by a number.) Parameter0 stands for the command filename and with it you may refer to the .CMD name itself. 1 through 9 are parameters in the command line. Parameter numbers may be repeated as often as desired in a file. Note: If an

up-arrow "↑" is desired for some reason as a character within a command, then use two up-arrows "↑↑" to distinguish it, and only one will be passed on. When the Batch command line is given, each word after the filename is treated as a parameter; however, more complex parameters may be enclosed in single quotes. If too many or not enough parameters are given, Batch ignores either the extra parameters or the extra commands, respectively. An example is needed to illustrate the above features.

Suppose the user creates (using EDIT) a file called EXAMPL.CMD containing the following:

```
DIR ↑1
REN OLDFILE↑2
```

The user then types:

```
@ EXAMPL NEWFILE '=NEWFILE'
```

to which the system then types the directory listing "NEWFILE" showing its size in K-bytes followed by:

```
REN OLDFILE=NEWFILE
```

and continues by renaming "NEWFILE". Note that the = sign was included in the single quotes, and hence, could be passed as part of the second parameter.

### DUMP

DUMP is used to display the contents of a file by records (128 bytes). The format is:

```
DUMP[afr]⟨CR⟩
```

If an afr is used, only the first file in the directory found to match the name is typed. The file is dumped in hex, with the first address of a line displayed along the left margin and the ASCII corresponding to the hex displayed as characters on the right margin. Thus, unlike the TYPE command, both ASCII and binary files may be DUMPed and it is possible to read them both. When binary files are dumped, the ASCII displayed is meaningless. The records are numbered starting

with 0; two records form one 256 byte "page".

## EDIT

The EDIT command is used to call the powerful Cromemco Text Editor. This program enables one to create and store on diskette, data files, program listings (any language), or bodies of text. It includes capabilities for character, word, or line editing in both the forward and backward directions through the text buffer. It is able to perform unconditional command iterations (macros) or to perform iterations based on the satisfaction of certain conditional tests. These and other features are described in the CDOS Text Editor Manual. The form of the command is one of the following:

EDITØufr<CR> (current drive)

EDITØD:ufr=S:afr<CR>

where S and D stand for the letter of the source and destination drives, respectively, and "Ø" stands for at least one blank space.

## INIT (Initialize)

The INIT program is used to initialize or format (ie., number the tracks and sectors) a new diskette. Most 8" floppies are now supplied formatted according to the IBM 3740 Data Entry System Format. It is unnecessary to re-initialize such diskettes when new. However, since no standard has been set for the 5" floppies, these must be formatted using INIT. Occassionally, a large floppy must also be re-initialized due to magnetic damage. The form of the command is simply:

INIT<CR>

Following this, the system will ask for the drive letter (A,B,C, or D) of the disk to be initialized and whether the disk is a mini (Y or N). Before typing the Y or N, be sure to insert the disk as the system starts to format it immediately. See Section 5 for a description of the errors which may occur during INIT.



STAT (Disk Status) NOTE: STAT is not currently implemented in CDOS.

The STAT command is used to check the remaining space on the disk. The form is simply: STAT<CR>. Following this STAT will print on the console, (a) the number of bytes still available on a diskette (based on 241 K-bytes maximum), (b) the number of directory entries (actually the number of extents; 64 is the maximum available), (c) the names of any null files (these may be deleted), and (d) any error messages. STAT runs a validation of the disk directory to see if any cross-linked files have occurred (which will happen if the "warm" start ↑C was forgotten when changing diskettes). Thus, the possible error messages are:

XXH NOT ALLOCATED

CTRL-C was not pressed

XXH LINKED BLOCK filename-1 filename-2

This means that a disk block has been allocated to two files because ↑C was not pressed prior to a write. The more recently created of the two files is unchanged; therefore, the first will have to be deleted using ERA, ↑C must be pressed, and then the second file will be okay. ASCII files can first be checked by using TYPE to display them and determine which file is bad.

WRTSYS (Write System)

The WRTSYS program is used to initialize or copy to another disk the CDOS resident image at the front of a system diskette. The form of the command is:

WRTSYS D: [ufr] [=] S: [ufr]

where D and S stand for the destination and source disks, respectively, and the other bracketed quantities are optional. The equal sign may be replaced by a space. If no ufr is given on either side, the program assumes that it will copy from and to the beginning of a disk. Hence, if a CDOS system disk is used as the source, the following command will copy the CDOS resident image to a blank disk:

WRTSYS D:=S:

However, if it is desired to copy CDOS to a file, or copy other system programs, the source and/or destination ufr's may be used. Note that any file copied in this way should have an .SYS filename extension. The following examples will illustrate these points:

```
WRTSYS A:=B:      (put CDOS of disk B onto disk A)
WRTSYS B:CDOS.SYS=A: (copy resident CDOS of disk A
                    into a file on disk B)
```

### XFER (Transfer)

The XFER command program allows files to be transferred from disks or devices to other files, diskettes, or devices or to be concatenated with other files. There are two forms of the command:

```
XFER<CR>          (repeat mode)
```

```
XFER[/s] [D:] ufr=[S:] ufr[,ufr,...] <CR> (one-time mode)
```

where S and D stand for the source and destination drives respectively (which are not needed if both files reside on the current drive) and where /s stands for one of the switches: /A, /C, /T, /V, described below. The optional ufr's on the right side of the second expression would be used if it was desired to concatenate several files to one. Following execution of the command in one-time mode, control is returned to CDOS. If the user desires to transfer several files without returning to CDOS each time, repeat mode may be used. XFER then prompts with a "!". Any time this character appears, a command may be issued of exactly the form shown in the second expression without having to type the XFER. The sequence may be terminated by the null string, ie., by typing <CR> to the prompt "!".

The switches which may be used with XFER are as follows:

/A This specifies that the source file is ASCII: otherwise binary is assumed. This serves to strip the parity bit (bit 7,MSB) of each byte and insert a ↑Z to indicate the

end of file. This is ordinarily only important when concatenating two ASCII files; if the /A were not used, the entire data block at the end of file (128 bytes) would be transferred and some extraneous information would be included.

- /C This compares one file to another without actually performing a transfer. Note that both filenames must be existing files to perform a comparison.
- /T This switch expands all tab characters into eight equivalent spaces while transferring the file to the new file or device. This is useful for printer listings as most printers will not expand tabs.
- /V This verifies the transferred file after the transfer is complete. This is similar to the /C switch, except that the transfer is performed and thus only the sourcefile need exist prior to XFER.

If a transfer is unsuccessful due to an incorrect filename specification, XFER will print the rest of the command line following the character which was unsuccessfully read and return to CDOS.

The following examples will serve to illustrate the many features of XFER:

|                   |                                       |
|-------------------|---------------------------------------|
| A.XFER<CR>        | (select repeat mode)                  |
| !SAVE=ORIGINAL    | ("ORIGINAL" is copied into "SAVE")    |
| !/C SAVE=ORIGINAL | (the two files are compared)          |
| !<CR>             | (null string returns control to CDOS) |

Note that the following single command would have the same effect as the above four commands:

XFER/V SAVE=ORIGINAL<CR>

An example of the use of a device name is:

```
XFER/T PRT:=A:THISFILE
```

An example of the transfer of a file to another disk is:

```
XFER C:EXAMPLE.FOR=B:EXAMPLE.FOR
```

Note that the transfer does not have to involve a file on the current drive (A in the above). An example of the concatenation of several ASCII files is:

```
XFER/A PROGRAM=PART1,PART2
```

## SECTION 5

SYSTEM FAILURE

## 5.1 Error Messages from the Diskette

CDOS prints out a full error message for failures involving disk operations. This error message is of the following form:

DISK word ERROR, DISK=xx, STATUS=bb, TRACK=yy, SECTOR=zz

where "word" stands for one of the following:

- . SEEK - error occurred in seeking a track on the disk.
- READ - error occurred during a read from the disk.
- WRITE - error occurred during a write to the disk.
- INIT - error occurred during disk initialization.

"xx" stands for:

- 00 - drive A
- 01 - drive B
- 02 - drive C
- 03 - drive D

"yy" stands for the TRACK number in hexadecimal,

"zz" stands for the SECTOR number in hex,

"bb" stands for an 8-bit STATUS byte displayed in hex which describes the error depending on the type it is as follows:

(See chart next page)

| STATUS<br>Bits | SEEK            | READ             | WRITE            | INIT           |
|----------------|-----------------|------------------|------------------|----------------|
| 7              | NOT READY       | NOT READY        | NOT READY        | NOT READY      |
| 6              | WRITE PROTECT++ | RECORD TYPE++    | WRITE PROTECT    | WRITE PROTECT  |
| 5              | HEAD ENGAGED++  | RECORD TYPE++    | WRITE FAULT      | WRITE FAULT    |
| 4              | SEEK ERROR      | RECORD NOT FOUND | RECORD NOT FOUND | NOT USED++     |
| 3              | CRC ERROR       | CRC ERROR        | CRC ERROR        | NOT VALID      |
| 2              | TRACK 0++       | LOST DATA        | LOST DATA        | LOST DATA      |
| 1              | INDEX++         | DATA REQUEST++   | DATA REQUEST++   | DATA REQUEST++ |
| 0              | BUSY++          | BUSY++           | BUSY++           | BUSY++         |

### Status and Error Summary

CRC error in the above means Cyclic Redundancy Check.

"++" in the above means that the given condition is not an error; it is merely a condition that was present when the error occurred. For example, if the STATUS byte was 30 (H) during a SEEK error, this means that bits 4 and 5 are "1's". Thus, there is a SEEK ERROR and the disk HEAD ENGAGED. However, having the head engaged is not an error, but a condition that was present when the error occurred. Hence, HEAD ENGAGED is marked with a "++".

When an error message is given, the user may take one of the following actions:

Press R - tells system to retry the disk I/O. Note:  
A disk error is not displayed until after  
CDOS has already done 10 retries.

Press RETURN or any other key - ignore the error and  
continue.

Press ↑C- do a "warm" start, enabling the user to try  
again from the beginning; this is the usual  
step if the disk error is something obvious  
like not having inserted a diskette into the  
drive.

## 5.2 Precautions Concerning Diskettes

Following are several precautions to take to minimize the chance of damage to files stored on a diskette.

- 1) Whenever changing diskettes, log-in the newly-inserted disk by doing a "warm start" (CTRL-C) or by rebooting CDOS (by resetting the computer). Execute STAT (type: STAT<CR>) occasionally to verify the directory.
- 2) Diskettes are magnetic media, and as such, should receive special care.
  - a) Keep them away from all sources of magnetic fields such as power transformers or solenoids.
  - b) Store them in their dust covers and NEVER lay the bare disk down on a dusty surface.
  - c) Keep them out of direct sunlight as the black plastic heats up quite rapidly. Normal storage temperatures are 50°-125°F. (10-50°C.)
  - d) Do not write on plastic disk jacket with anything but a felt-tip pen.
  - e) Do not touch or try to clean disk surface; abrasions may cause loss of data.
  - f) It is highly recommended that you NOT have a disk loaded (ie., inserted into the drive with the door closed) while powering up or down. Under these conditions random data may be written onto the disk. In the case of a power failure it is wise to check the disk for errors following return of power.
- 3) As an additional safety precaution, maintain adequate archives of back-up disks. Data will occasionally be lost and the additional cost of back-up disks is well worth the valuable programs and data which are saved.

## SECTION 6

COMMAND SUMMARY

| <u>Intrinsic Commands</u> |                                                             | <u>Page</u> |
|---------------------------|-------------------------------------------------------------|-------------|
| BYE                       | Causes a return to RDOS                                     | 18          |
| DIR [afr]                 | List directory entry(s).                                    | 18          |
| ERA afr                   | Erase the file(s) named.                                    | 18          |
| REN afr=afr               | Rename the file(s).                                         | 19          |
| SAVE ufr n                | Save n pages of 256 bytes each.                             | 19          |
| TYPE afr                  | Type the file in ASCII.                                     | 19          |
| <u>Extrinsic Commands</u> |                                                             | <u>Page</u> |
| @(Batch) ufr              | Perform the command stream contained in the indicated file. | 20          |
| DUMP afr                  | Dump the file in binary and ASCII by records.               | 22          |
| EDIT ufr                  | Edit the indicated file.                                    | 23          |
| INIT                      | Initialize diskettes.                                       | 23          |
| STAT                      | (Not currently implemented)                                 |             |
| WRTSYS D:=S:              | Copy CDOS system image to another diskette.                 | 24          |
| XFER[/s] ufr=ufr          | Transfer the second file into the first.                    | 25          |

where

|       |                                                                      |
|-------|----------------------------------------------------------------------|
| afr   | means ambiguous file references allowing the "?" and "*" characters. |
| ufr   | means unambiguous file references (the complete filename).           |
| S:,D: | are source and destination.                                          |
| /s    | is one of the described switches.                                    |



Explanation of the DIR listing

Below is a sample CDOS directory listing obtained after executing the DIR command:

```

A.DIR
@          COM          2/ 1
XFER       COM          4/ 1
DUMP       COM          2/ 1
WRTSYS     COM          1/ 1
EDIT       COM          6/ 1
INIT       COM          2/ 1
FOR        COM          23/ 2
LINK       COM          4/ 1
FORLIE     REL          15/ 1
DEMO       FOR          1/ 1
COPY       CMD          1/ 1
11(12)  FILES (EXTENTS)  TOTAL KBYTES:   61

```

The first column contains the file names of the programs currently on the disk. Each file name may be up to 8 letters in length. The second column contains the 3-letter extension to each file name. The third column gives first the number of kilobytes occupied by each file followed by the number of extents (16 kilobyte blocks) occupied by each file. Thus a file under 16 kilobytes occupies only one extent while longer files occupy more.

The last line of the Directory listing shows the total number of files followed by the total number of extents. The total number of kilobytes used for file storage is also shown on this line. The total number of kilobytes available for user files on a 5" diskette is 81 kilobytes. The total available on a 8" diskette is 241 kilobytes.

Index to Additional Information

| <u>Information</u>                         | <u>Page</u> |
|--------------------------------------------|-------------|
| Allowed Device Names                       | 7           |
| Standard Filename Extensions               | 8           |
| Control Characters                         | 13          |
| Printer Control Characters                 | 14          |
| Reboot CDOS Descriptions (Getting Started) | 11          |