Color Computer Disk System

Quick Reference Guide

* TRADEMARKS OF TANDY CORPORATION
Once you connect a floppy disk drive to your Color Computer, you automatically start up in Disk BASIC whenever you turn on the computer. You can verify that you have properly connected your disk interface cartridge and disk drive by looking at the screen. If everything is connected properly, the screen displays the DISK EXTENDED COLOR BASIC copyright message. If it is not, the screen displays the EXTENDED COLOR BASIC message.

This guide summarizes the commands you can use while in Disk BASIC. It also lists error messages you might receive.
Disk BASIC Commands

BACKUP source drive TO destination drive
Duplicates the contents of the disk in the source drive on the disk in the destination drive. If you have only one drive, specify it as the source drive.

```
BACKUP 0 TO 1  BACKUP 0
```

CLOSE #buffer,...
Closes communication to the buffers specified. If you omit the buffer, the computer closes all open files.

```
CLOSE #1  CLOSE #1,#2
```

COPY "filename1" TO "filename2"
Copies the contents of filename1 to filename2. Each filename must include an extension.

```
COPY "FILE/BAS" TO "NEWFILE/BAS"
COPY "ORG/DAT:0" TO "ORG/DAT:1"
```

CVN(string variable)
Converts a 5-byte coded string (created by MKN$) back to the number it represents.

```
X=CVN(A$)
```

DIR drive number
Displays a directory of the disk in the drive you specify.

```
DIR0  DIR
```
Sample display:

```
MYPROG  BAS  0  B  3
YOURPROG  BAS  0  A  1
HERDATA  DAT  1  A  5
USPROG  BIN  2  B  2
```
From left to right, the columns contain:
- The filename
- The extension
- The file type
  - 0 = BASIC program
  - 1 = BASIC data file
  - 2 = machine-language file
  - 3 = editor source file
- The storage format
  - A = ASCII, B = binary
- The length of the file, in granules

DOS
With the OS-9 system diskette in Drive 0, the DOS command boots the OS-9 operating system.

```
DOS
```
Disk BASIC Commands

**DRIVE drive number**
Changes the default drive to the drive you specify. If you do not use the DRIVE command, the computer uses Drive 0 as the default.

```
DRIVE 1
```

**DSKIS$ drive number, track, sector, string variable1, string variable2**
Inputs data from a particular sector within a particular track on the disk in the drive you specify.

```
DSKIS$ 0, 12, 3, MS$, NS$
```

**DSKINI drive number**
Formats a disk in the drive you specify. Executing this command erases memory.

```
DSKINI0   DSKINI1
```

**DSKOS$ drive number, track, sector, string1, string2**
Writes string data on the sector, track, and drive number you specify.

```
DSKOS$ 0, 2, 1, "FIRST DATA," "SECOND DATA"
```

**EOF(buffer)**
Returns a value of 0 if there is more data to read in the buffer and a value of -1 if there is no more data in it.

```
IF EOF(1) = -1 THEN CLOSE 111
```

**FIELD #buffer, field size AS field name,...**
Organizes the space within a direct access buffer into fields. Specify the size and name of each field.

```
FIELD #111,10 AS AS, 12 AS BS, 5 AS CS
```

**FILES number of buffers, size**
Tells the computer the number of buffers to reserve in memory, and the total number of bytes (size) to reserve for these buffers. If you do not specify the size, the computer reserves a total of 256 bytes.

```
FILES 1, 1000   FILES 5
```

**FREE (drive number)**
Returns the number of free granules on the disk in the drive you specify.

```
PRINT FREE (0)
```

**GET #buffer, record number**
Gets the next record or the record you specify, and puts it in the buffer.

```
GET #1, 5   GET #2, 3
```
INPUT #buffer, variable name,...
Inputs data from the buffer you specify, and assigns each data item in the buffer to the variable name you specify.

```plaintext
INPUT #1, A$, B$
```

KILL "filename"
Deletes the filename you specify from the disk directory. You must include the extension with the filename.

```plaintext
KILL "FILE/BAS" KILL "FILE/DAT:1"
```

LINE INPUT #buffer, data
Inputs a line (all data up to the ENTER character) from the buffer you specify.

```plaintext
LINE INPUT #1, XS
```

LOAD "filename", R
Loads the specified BASIC program file into memory from disk. If you include the R, the computer runs the program immediately after loading it.

```plaintext
LOAD "PROGRAM", R
LOAD "ACCTS/BAS:1"
```

LOADM "filename", offset address
Loads the specified machine language program file from disk. You can specify an offset address to add to the program's loading address.

```plaintext
LOADM "PROG/BIN", 3522
```

LOC(buffer)
Returns the current record number of the buffer you specify.

```plaintext
PRINT LOC(1)
```

LOF(buffer)
Returns the highest-numbered record of the buffer you specify.

```plaintext
FOR R = 1 TO LOF(1)
```

LSET field name = data
Left-justifies the data within the field you specify.

```plaintext
LSET A$="BANANAS" LSET B$=T$
```

MERGE "filename", R
Loads the specified program file from disk, and merges it with the program that exists in memory. If you include the R, the computer runs the program immediately after merging it.

```plaintext
MERGE "SUB/BAS" MERGE "NEW", R
```
## Disk BASIC Commands

### MKNS$(number)
Converts the specified `number` to a 5-byte coded string, for storage in a formatted disk file.

```basic
LSET BS = MKNS$(53678910)
```

### OPEN "mode", # buffer, "filename", record length
Opens a buffer that transfers data to and from a particular device. If you don’t specify the `record length`, the computer uses 256 bytes.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Allows</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data input from a sequential access file.</td>
</tr>
<tr>
<td>0</td>
<td>Data output to a sequential access file.</td>
</tr>
<tr>
<td>D</td>
<td>Data transfer to or from a direct access file.</td>
</tr>
</tbody>
</table>

Buffer  Communicates With
---      ------------------------------
- 2      The printer.
- 1      The tape recorder.
0        The screen or printer. (It is not necessary to open this buffer.)
1-15     The disk drives.

```basic
OPEN "D", #1, "FILE", 15
OPEN "I", #2, "CHGE/DAT"
```

### PRINT # buffer, data list
Prints the `data` to the `buffer`. Use a comma or a semicolon to separate each item in the list.

```basic
PRINT #1, "DATA"
```
Disk BASIC Commands

PRINT \#buffer, USING format; data list
Prints the data to the buffer, using the format you specify. The format is a string; enclose it in quotation marks.

The format commands are:

# Holds a space for one digit.
. Prints a decimal point.
, Prints a comma immediately preceding every third digit (counting to the left from the decimal point).
** Fills leading spaces with asterisks.
$ Prints a leading dollar sign.
$$ Prints a floating dollar sign.
+ Prints the sign of the number. To print the sign in front of the number, place the plus sign at the beginning of the format string. To print the sign following the number, place the plus sign at the end of the format string.
^~~~ Prints the number in exponential format.
_ Prints a minus sign after the number if the number is negative. This command does not print a sign if the number is positive. Place the minus sign at the right end of the format string.
! Prints the first character of the string.
%spaces% Sets the field for the string. The length of the field is the number of spaces plus 2.

PRINT #1, USING "##.#"; 53.76
PRINT #2, USING "**$#.###"; -3.678
PRINT #1, USING "!"; "WHITE"

PUT \#buffer, record number
Assigns a record number to the data in the buffer you specify. If you do not specify a record number, the computer assigns it to the current record.

PUT #2, 3 PUT #1, 4

RENAME "old filename" TO "new filename"
Renames a disk file. You must specify the extension of both filenames.

RENAME "MFILE/DAT:1"
TO "BFILE/DAT:1"

RSET field name = data
Right-justifies the data within the field you specify.

RSET M$ = "SOAP"
Disk BASIC Commands

RUN "filename", R
Loops filename from disk, and runs it. If you include the R, all open files remain open.

RUN "FILE" RUN "PROG/BAS", R

SAVE "filename", A
Saves filename on disk. By using the A option, you save the program in ASCII format.

SAVE "PROG/BAS" SAVE "TEST:1", A

SAVEM "filename", first address, second address, third address
Saves filename, a machine-language program residing in the memory location that begins at first address and ends at second address. The third address is the execution address.

SAVEM "FILE/BIN:1", &H5200, &H5800, &H5300

UNLOAD drive number
Closes any open files on the disk in the drive you specify. If you do not specify a drive number, the computer uses Drive 0 (or the drive you specified in the DRIVE command).

UNLOAD 0 UNLOAD

VERIFY ON VERIFY OFF
Turns the verify function on or off. When VERIFY is on, the computer verifies all writes to the disk.

VERIFY ON

WRITE #buffer, data list
Writes the data to the buffer you specify. Use a comma to separate each data item in the list.

WRITE #1, A$, B$, C
<table>
<thead>
<tr>
<th>Error No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Division by zero</td>
</tr>
<tr>
<td>AE 33</td>
<td>File already exists</td>
</tr>
<tr>
<td>AO 27</td>
<td>Attempt to open a data file that is already open</td>
</tr>
<tr>
<td>BR</td>
<td>Bad record number</td>
</tr>
<tr>
<td>BS</td>
<td>Bad subscript</td>
</tr>
<tr>
<td>CN</td>
<td>Cannot continue</td>
</tr>
<tr>
<td>DD</td>
<td>Attempt to redimension an array</td>
</tr>
<tr>
<td>DF 28</td>
<td>Disk full</td>
</tr>
<tr>
<td>DN</td>
<td>Drive number or device number error</td>
</tr>
<tr>
<td>DS</td>
<td>Direct statement</td>
</tr>
<tr>
<td>ER 37</td>
<td>Write or input past end of record (direct access only)</td>
</tr>
<tr>
<td>FC</td>
<td>Illegal function call</td>
</tr>
<tr>
<td>FD</td>
<td>Bad file data</td>
</tr>
<tr>
<td>FM</td>
<td>Bad file mode</td>
</tr>
<tr>
<td>FN 31</td>
<td>Bad file name</td>
</tr>
<tr>
<td>FO 34</td>
<td>Field overflow</td>
</tr>
<tr>
<td>FS 32</td>
<td>Bad file structure</td>
</tr>
<tr>
<td>HP</td>
<td>High-resolution print error (Color Computer 3 only)</td>
</tr>
<tr>
<td>HR</td>
<td>High-resolution graphics error (Color Computer 3 only)</td>
</tr>
<tr>
<td>ID</td>
<td>Illegal direct statement</td>
</tr>
<tr>
<td>IE</td>
<td>Input past end of file</td>
</tr>
<tr>
<td>IO</td>
<td>Input/output error</td>
</tr>
<tr>
<td>LS</td>
<td>String too long</td>
</tr>
<tr>
<td>NE 26</td>
<td>Cannot find the disk file</td>
</tr>
<tr>
<td>NF</td>
<td>NEXT without FOR</td>
</tr>
<tr>
<td>NO</td>
<td>File not open</td>
</tr>
<tr>
<td>OB 29</td>
<td>Out of buffer space</td>
</tr>
<tr>
<td>OD</td>
<td>Out of data</td>
</tr>
<tr>
<td>OM</td>
<td>Out of memory</td>
</tr>
<tr>
<td>OS</td>
<td>Out of string space</td>
</tr>
<tr>
<td>OV</td>
<td>Overflow</td>
</tr>
<tr>
<td>RG</td>
<td>RETURN without GOSUB</td>
</tr>
<tr>
<td>SE 35</td>
<td>Set to non-fielded string</td>
</tr>
<tr>
<td>SN</td>
<td>Syntax error</td>
</tr>
<tr>
<td>ST</td>
<td>String formula too complex</td>
</tr>
<tr>
<td>TM</td>
<td>Type mismatch</td>
</tr>
<tr>
<td>UL</td>
<td>Undefined line</td>
</tr>
<tr>
<td>VF 36</td>
<td>Verification</td>
</tr>
<tr>
<td>WP 30</td>
<td>Write-protected disk</td>
</tr>
</tbody>
</table>