TANDY®*

Color Computer Disk System

Quick Reference Guide

* TRADEMARKS OF TANDY CORPORATION

Color Computer
Disk BASIC

Quick
Reference
Guide

Introduction

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 Ø TO 1   BACKUP Ø
```

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:Ø" 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.

```
DIRØ   DIR
```

Sample display:

```
MYPROG      BAS   Ø   B   3
YOURPROG    BAS   Ø   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
```

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
```

DSKI$ *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.

```
DSKI$ 0, 12, 3, M$, N$
```

DSKINI *drive number*

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

```
DISKINI0   DISKINI1
```

DSKO$ *drive number, track, sector, string1, string2*

Writes string data on the *sector*, *track*, and *drive number* you specify.

```
DSKO$ 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 #1
```

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 #1, 10 AS A$, 12 AS B$, 5 AS C$
```

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.

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

KILL "*filename*"

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

```
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.

```
LINE INPUT #1, X$
```

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.

```
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.

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

LOC(*buffer*)

Returns the current record number of the buffer you specify.

```
PRINT LOC(1)
```

LOF(*buffer*)

Returns the highest-numbered record of the buffer you specify.

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

LSET *field name* = *data*

Left-justifies the data within the field you specify.

```
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.

```
MERGE "SUB/BAS"   MERGE "NEW", R
```

MKN$(*number*)

Converts the specified *number* to a 5-byte coded string, for storage in a formatted disk file.

```
LSET B$ = MKN$(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.

Mode	Allows
I	Data input from a sequential access file.
O	Data output to a sequential access file.
D	Data transfer to or from a direct access file.

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.

```
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.

```
PRINT #1, "DATA"
```

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"
```

RUN "*filename*", R

Loads *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
```

Error Messages

Error	*No.*	*Description*
/0		Division by zero
AE	33	File already exists
AO		Attempt to open a data file that is already open
BR	27	Bad record number
BS		Bad subscript
CN		Cannot continue
DD		Attempt to redimension an array
DF	28	Disk full
DN		Drive number or device number error
DS		Direct statement
ER	37	Write or input past end of record (direct access only)
FC		Illegal function call
FD		Bad file data
FM		Bad file mode
FN	31	Bad filename
FO	34	Field overflow
FS	32	Bad file structure
HP		High-resolution print error (Color Computer 3 only)
HR		High-resolution graphics error (Color Computer 3 only)
ID		Illegal direct statement
IE		Input past end of file
IO		Input/output error
LS		String too long
NE	26	Cannot find the disk file
NF		NEXT without FOR
NO		File not open
OB	29	Out of buffer space
OD		Out of data
OM		Out of memory
OS		Out of string space
OV		Overflow
RG		RETURN without GOSUB
SE	35	Set to non-fielded string
SN		Syntax error
ST		String formula too complex
TM		Type mismatch
UL		Undefined line
VF	36	Verification
WP	30	Write-protected disk

CUSTOM MANUFACTURED FOR
RADIO SHACK DIVISION, InterTAN CANADA LTD.
BARRIE, ONTARIO. L4M 4W5

8A7
Cat. No. 26-3133

Printed in Korea
811019890A