

BRADLEY'S BASIC

PROGRAMMER'S REFERENCE GUIDE

by
Matthew Desmond

FactorOfMatt
factorofmatt.com

TABLE OF CONTENTS

INTRODUCTION	3
What's included?.....	3
HOW TO USE THIS REFERENCE GUIDE.....	3
Notes from the author.....	3
BRADLEY'S BASIC LANGUAGE VOCABULARY	3
Bradley's BASIC Keywords and Abbreviations.....	3
Description of Bradley's BASIC Keywords (Alphabetical).....	3
DOS WEDGE	11

INTRODUCTION

The **BRADLEY'S BASIC PROGRAMMER'S REFERENCE GUIDE** has been developed as a working tool and reference for those of you who want to maximize your use of the additional capabilities brought to your **COMMODORE 64** when using **BRADLEY'S BASIC**.

What's included?

- Our complete "BRADLEY'S BASIC dictionary" includes the additional BASIC language commands and variables listed in alphabetical order. We've created a "quick list" which contains the words and their abbreviations. This is followed by a section containing a more detailed definition of each word along with sample BASIC programs to illustrate how they work.
- A guide to using the BRADLEY'S BASIC fully-featured DOS-WEDGE.

How to use this reference guide

Throughout this manual certain conventional notations are used to describe the syntax (programming sentence structure) of BRADLEY'S BASIC commands or statements and to show both the required and optional parts of each BRADLEY'S BASIC keyword. The rules to use for interpreting statement syntax are as follows:

1. BASIC keywords are shown in capital letters. They must appear where shown in the statement, entered and spelled exactly as shown.
2. Items shown within quotation marks (" ") indicate variable data which you must put in. Both the quotation marks and the data inside the quotes must appear where shown in each statement.
3. Items inside the square brackets ([]) indicate an optional statement parameter. A parameter is a limitation or additional qualifier for your statements. If you use an optional parameter you must supply the data for that optional parameter.
4. Items inside angle brackets (<>) indicate variable data which you provide. While the slash (|) indicates that you must make a choice between two mutually exclusive options.

EXAMPLE OF SYNTAX FORMAT:

**COPY [Ddrive number,]"source filename" TO
[Ddrivenumber,]"destination filename"[<ON|,>Udevice]**

EXAMPLES OF ACTUAL STATEMENTS:

```
10 DOPEN#2,"FILE"  
20 RECORD#2,10,1  
30 PRINT#2,A$  
40 DCLOSE#2
```

When you actually apply the syntax conventions in a practical situation, the sequence of parameters in your statements might not be exactly the same as the sequence shown in syntax

examples. The examples are not meant to show every possible sequence. They are intended to present all required and optional parameters.

Programming examples in this book are shown with blanks separating words and operators for the sake of readability. Normally though, BASIC doesn't require blanks between words unless leaving them out would give you an ambiguous or incorrect syntax.

Shown below are some examples and descriptions of the symbols used for various statement parameters in the following chapters. The list is not meant to show every possibility, but to give you a better understanding as to how syntax examples are presented.

SYMBOL	EXAMPLE	DESCRIPTION
<logical-file-numbe>r	15	A logical file number
<device>	8	A hardware device number
<drive>	0	A physical drive number

Notes from the author

This entire reference-guide has been modelled on the **COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE** right down to the font, colours, formatting and even some of the text. It worked in the 80's so it should work now.

Bradley's BASIC was initially intended as a simple cartridge that would allow David Bradley to boot directly to a white-on-black screen instead of typing in a bunch of pokes. Nice idea but adding such into a bootable cartridge would steal 8K of RAM for relatively little gain. So why not add in the BASIC 4.0/7.0 disk commands too? And a couple of cool extras? And a dos-wedge? Sure. And BRADLEY'S BASIC is born.

The documentation for the BASIC-4.0/7.0 disk commands are lifted from the **COMMODORE 128 PROGRAMMER'S REFERENCE GUIDE**, just reformatted a little.

Aside from the DOS-WEDGE code, almost all of BRADLEY'S BASIC pre-existed and was written by me sometime between 1983 and 1987. I did take some time to make the code better and more compatible with the original BASIC 4.0/7.0 versions.

BRADLEY'S BASIC is intended to be placed in a cartridge (at \$8000 with EXROM held low) though I will release a soft-loadable version too.

BRADLEY'S BASIC LANGUAGE VOCABULARY

Introduction

This chapter explains BRADLEY'S BASIC new Language keywords. First we give you an easy to read list of keywords, their abbreviations and what each letter looks like on the screen.

Then we explain how the syntax and operation of each keyword works in detail, and examples are show to give you an idea how to use them in your programs.

BRADLEY'S BASIC consists 27 commands (5 of which aren't currently implemented – but are kept for backwards compatibility), two new functions and a convenience keyword. Anybody who has used disk-commands in BASIC4.0/7.0 (PET computers or the Commodore 128) should be familiar with most of the new keywords and their abbreviations.

Bradley's BASIC Keywords and Abbreviations

COMMAND	ABBREVIATION	SCREEN	FUNCTION TYPE
APPEND	A SHIFT P	A	
BACKUP	B SHIFT A	B	
BLOAD	B SHIFT L	B	
BSAVE	B SHIFT S	B	
CATALOG	C SHIFT A	C	
COLD	none	COLD	
COLLECT	CO SHIFT L	CO	
CONCAT	CON SHIFT C	CON	
COPY	CO SHIFT P	CO	
DCLEAR	D SHIFT C	D	
DCLOSE	DC SHIFT L	DC	
DIRECTORY	DI SHIFT R	DI	
DISK	DI SHIFT S	DI	
DLOAD	D SHIFT L	D	
DOPEN	D SHIFT O	D	
DS	none	DS	NUMERIC
DS\$	none	DS\$	STRING
DSAVE	D SHIFT S	D	
DVERIFY	D SHIFT V	D	

COMMAND	ABBREVIATION	SCREEN	FUNCTION TYPE
HEADER	H SHIFT E	H —	
HELP	HE SHIFT L	HE L	
INIT	IN SHIFT I	IN ↵	
OLD	O SHIFT L	O L	
RECORD	RE SHIFT C	RE —	
RENAME	RE SHIFT N	RE ↗	
SCRATCH	S SHIFT C	S —	
WEDGE	W SHIFT E	W —	

Description of Bradley's BASIC Keywords (Alphabetical)

APPEND

TYPE: I/O Statement (BASIC4.0) [NOT IMPLEMENTED]

FORMAT: APPEND #logical file number,"filename"[,Ddrive number] [<ON** | ,>Udevice]**

Action: Append data to the end of a sequential file.

EXAMPLES of APPEND Statement:

APPEND #8, "MYFILE"

OPEN logical file 8 called "MYFILE" , and prepare to append with subsequent PRINT# statements.

APPEND #7,(A\$),D0,U9

OPEN logical file named by the variable in A\$ on drive 0, device number 9, and prepare to APPEND.

BACKUP

TYPE: I/O Statement (BASIC4.0)

FORMAT: BACKUP source Ddrive number TO destination Ddrive number [<ON** | ,>Udevice]**

Action: Copy the entire contents from one disk to another on a dual disk drive.

NOTE: This command can be used only with a dual-disk drive.

EXAMPLES of BACKUP Statement:

BACKUP DO TO D1

Copies all files from the disk in drive 0 to the disk in drive 1, in dual disk drive device 8.

BACKUP DO TO D1 ON U9

Copies all files from drive 0 to drive 1, in disk drive device 9.

BLOAD

TYPE: I/O Statement (BASIC7.0)

FORMAT: BLOAD "filename"[,Ddrive number][<ON|,U>device number][,Pstart address]

where:

filename is the name of your file

start address is the memory location where loading begins

Action: Load a binary file starting at the specified memory location.

EXAMPLES of BLOAD Statement:

BLOAD "SPRITES", P3584

LOADS the binary file "SPRITES" starting in location 3584.

BLOAD "DATA1", D0, U8, P4096

LOADS the binary file "DATA1" into location 4096 from Drive 0, device 8.

BSAVE

TYPE: I/O Statement (BASIC7.0)

FORMAT: BSAVE "filename"[,Ddrive number][<ON|,U>device number] ,Pstart address TO Pend address

where:

- start address is the starting address where the program is SAVED from
- end address is the last address-1 in memory which is SAVED

Action: Save a binary file from the specified memory locations

EXAMPLES of BSAVE Statement:

BSAVE "SPRITE DATA",P3584 TO P4096

Saves the binary file named "SPRITE DATA", starting at location 3584 through 4095.

BSAVE "PROGRAM.SCR",D0,U9,P3182 TO P7999

Saves the binary file named "PROGRAM.SCR" in the memory address range 3182 through 7998 on drive 0, disk 9.

CATALOG

CONCAT "FILE B" to "FILE A"

FILE B is attached to FILE A, and the combined file is designated FILE A.

CONCAT (A\$) to (B\$), D1, U9

The file named by B\$ becomes a new file with the same name with the file named by A\$ attached to the end of B\$. This is performed on device 9, drive 1 (a dual disk drive).

Whenever a variable is used as a filename, as in the last example, the filename variable ^ must be within parentheses.

COPY

TYPE: I/O Statement (BASIC4.0)

FORMAT: COPY [Ddrive number,]"source filename" TO [Ddrivenumber,]"destination filename" [<ON|,>Udevice]

Action: Copy a file from one drive to another within a dual disk drive. Copy one file to another with a different name within a single drive.

NOTE: Copying between two single or double disk drive devices cannot be done. This command does not support device-to-device copying.

EXAMPLES of COPY Statement:

COPY D0, "TEST" TO D1, "TEST PROG"

Copies "TEST" from drive 0 to drive 1, renaming it "TEST PROG" on drive 1.

COPY D0, "STUFF" TO D1, "STUFF"

Copies "STUFF" from drive 0 to drive 1.

COPY D0 TO D1

Copies all files from drive 0 to drive 1.

COPY "WORK.PROG" TO "BACKUP"

Copies "WORK.PROG" as a file called "BACKUP" on the same disk (drive 0).

DCLEAR

TYPE: I/O Statement (BASIC7.0) [NOT IMPLEMENTED]

FORMAT: DCLEAR [Ddrive number][<ON|,>Udevice]

Action: Clear all open channels on disk drive.

EXAMPLES of DCLEAR Statement:

DCLEAR D0

Clears all open files on drive 0, device number 8.

DCLEAR D1, U9

Clears all open files (channels) on drive 1, device number 9.

DCLOSE

TYPE: I/O Statement (BASIC4.0) [NOT IMPLEMENTED]

FORMAT: DCLOSE [#logical file number][<ON|,>Udevice]

Action: Close disk file.

EXAMPLES of DCLOSE Statement:

DCLOSE	Closes all channels currently open on device 8.
DCLOSE #2	Closes the channel associated with the logical file number 2 on device 8.
DCLOSE ON U9	Closes all channels currently open on device 9.

DIRECTORY

TYPE: I/O Statement (BASIC4.0)

FORMAT: DIRECTORY [Ddrive number][<ON|,>Udevice][,wildcard]

Action: Display the contents of the disk directory on the screen.

EXAMPLES of DIRECTORY Statement:

DIRECTORY	Lists all files on the disk in device 8.
DIRECTORY D1, U9, "WORK"	Lists the file named "WORK" on drive 1 of device 9.
DIRECTORY "AB*"	Lists all files starting with the letters "AB" like ABOVE, ABOARD, etc. on device 8. The asterisk specifies a wild card, where all files starting with "AB" are displayed.
DIRECTORY D0, "?BAK"	The ? is a wild card that matches any single character in that position. For example: FILE 1.BAK, FILE 2.BAK, FILE 3.BAK all match the string.
DIRECTORY D1,U9,(A\$)	LISTS the filename stored in the variable A\$ on device number 9, drive 1. Remember, whenever a variable is used as a filename, put the variable in parentheses.

NOTE: To print the DIRECTORY of the disk in drive 0, device 8, use the following example:
LOAD"\$0",8
OPEN4,4:CMD4:LIST
PRINT#4:CLOSE4

DISK

TYPE: I/O Statement (Bradley's BASIC only)

FORMAT: DISK ["disk-command"] [<ON|,>Udevice number]

Action: Read and return disk status channel or send an arbitrary command. If a disk-command is sent then DISK will wait for the operation to finish then automatically display the disk-error status.

NOTE: The disk-command is sent to the disk command channel as-is, no processing is performed first. If you wish to include a drive number then you must encode it in the disk-command.

EXAMPLES of DISK Statement:

DISK "NO:BRADLEY'S BASIC"	Quick-formats the disk in device 8, unit 0 with the name "BRADLEY'S BASIC"
DISK ON U9	Reads the disk error channel from device 9 and display it on the screen.

DLOAD

TYPE: I/O Statement (BASIC4.0)

FORMAT: DLOAD "filename" [,Ddrive number][<ON|,>Udevice number]

Action: Load a BASIC program from the disk drive, device 8.

EXAMPLES of DLOAD Statement:

DLOAD "BANKRECS"	Searches the disk for the program "BANKRECS" and LOADS it.
DLOAD (A\$)	LOADS a program from disk in which the name is stored in the variable A\$. An error message is given if A\$ is null. Remember, when a variable is used as a filename, it must be enclosed in parentheses.

DOPEN

TYPE: I/O Statement (BASIC4.0) [NOT IMPLEMENTED]

FORMAT: DOPEN # logical file number,"filename[,<type>"][,Lrecord length][,Ddrive number][<ON|,>Udevice number][,W]

where type is:

S = Sequential File Type

P = Program File Type

U = User File Type

R = Relative File Type

L = Record Length = the length of records in a relative file only

W = Write Operation (if not specified a read operation occurs)

Action: Open a disk file for a read and/or write operation

EXAMPLES of DOPEN Statement:

DOPEN#1, "ADDRESS",W

Create the sequential file number 1 (ADDRESS) for a write operation.

DOPEN#2 "RECIPES",DI,U9

Open the sequential file number 2 (RECIPES) for a read operation on device number 9, drive 1.

DS

TYPE: Numeric Function (BASIC4.0)

FORMAT: DS

Action: This function will return the disk error code that resulted from the most recent disk action (on whichever device it was).

NOTE: While **DS** acts like a variable it may not be assigned to and will only change if a disk operation or command is issued.

EXAMPLE of DS Statement:

PRINT DS
73

Displays the most recent disk error code.

DSS

TYPE: String Function (BASIC4.0)

FORMAT: DSS

Action: This function will return the disk error string that resulted from the most recent disk action (on whichever device it was).

NOTE: While **DSS** acts like a variable it may not be assigned to and will only change if a disk operation or command is issued.

EXAMPLE of DSS Statement:

PRINT DSS
73,CBM DOS V2.6 1541,00,00

Displays the most recent disk error string.

DSAVE

TYPE: I/O Statement (BASIC4.0)

FORMAT: DSAVE "filename" [,Ddrive number][<ON|,>Udevice number]

Action: Save a BASIC program file to disk

EXAMPLES of DSAVE Statement:

DSAVE "BANKRECS"
DSAVE (A\$)
DSAVE "PROG3",D1,U9

Saves the program "BANKRECS" to disk.
Saves the disk program named in the variable A\$.
Saves the program "PROG3" to disk on device number 9, drive 1.

DVERIFY

TYPE: I/O Statement (BASIC7.0)

FORMAT: DVERIFY "filename" [,Ddrive number] [<ON|,>Udevice number]

Action: Verify the program in memory against the one on disk.

To verify Binary data, see VERIFY "filename",8,1 format, under VERIFY command description.

EXAMPLES of DVERIFY Statement:

DVERIFY "C64"
DVERIFY "SPRITES",D0,U9

Verifies program "C64" on drive 0, device 8.
Verifies program "SPRITES" on drive 0, device 9.

HEADER

TYPE: I/O Statement (BASIC4.0)

FORMAT: HEADER "diskname" [,I i.d.][,Ddrive number] [<ON|,>Udevice number]

Action: Format a diskette. Before a new disk can be used for the first time, it must be formatted with the HEADER command. The HEADER command can also be used to erase a previously formatted disk, which can then be reused. When you enter a HEADER command in direct mode, the prompt ARE YOU SURE? appears. In program mode, the prompt does not appear. The HEADER command is analogous to the BASIC 2.0 command:

OPEN 1,8,15,"N0:diskname,i.d."

EXAMPLES of HEADER Statement:

HEADER "MYDISK",I23, D0

This headers " M Y D I S K " using i.d. 23 on drive 0, (default) device number 8.

HEADER "RECS", I45, D1 ON U9

This headers "RECS" using i.d. 45, on drive 1, device number 9.

HEADER "C64 PROGRAMS", D0

This is a quick header on drive 0, device number 8, assuming the disk in the drive was already formatted. The old i.d. is used.

HEADER (A\$),I76,D0,U9

This example headers the diskette with the name specified by the variable A\$, and the i.d. 76 on drive 0, device number 9.

HELP

TYPE: Statement (Bradley's BASIC only)

FORMAT: HELP

Action: Display a brief summary of BRADLEY'S BASIC and DOS-WEDGE commands.

EXAMPLES of INIT Statement:

HELP Displays a summary of BRADLEY'S BASIC commands.

INIT

TYPE: I/O Statement (Bradley's BASIC only)

FORMAT: INIT [Ddrive-number][<ON|,>Udevice number]

Action: Send an initialize command to the specified disk device and unit. This command will automatically read the disk-error channel and display it once the initialize command completes.

EXAMPLES of INIT Statement:

INIT Sends a disk-initialize command to drive 0, device 8.
INIT D1 ON U9 Sends a disk-initialize command to drive 1, device 9.

OLD

TYPE: Statement (Bradley's BASIC only)

FORMAT: OLD

Action: Attempt to un-NEW a program. In the event that you are forced reset your machine (via a reset-button or **COLD** statement) or if you just accidentally type **NEW** then the **OLD** statement may be able to restore your program.

NOTE: The **OLD** statement may not work correctly if any variables have been defined or new program-lines entered since the restart or **NEW** occurred.

EXAMPLES of OLD Statement:

OLD Un-NEWs a program.

RECORD

TYPE: I/O Statement (BASIC4.0) [NOT IMPLEMENTED]

FORMAT: RECORD# logical file number, record number [,byte number]

Action: Position relative file pointers. This statement positions a relative file pointer to select any byte (character) of any record in the relative file. When the record number value is set higher than the last record number in the file, the following occurs:

For a write (**PRINT#**) operation, additional records are created to expand the file to the desired record number.

For a read (**INPUT#**) operation, a null record is returned and a "RECORD NOT PRESENT ERROR occurs". See your disk drive manual for details about relative files.

EXAMPLE of RECORD Statement:

```
10 DOPEN#2,"FILE"  
20 RECORD#2,10,1
```

open existing relative-file called FILE as file 2.
position relative-file pointer to first byte in record number 10.

```
30 PRINT#2,A$  
40 DCLOSE#2
```

write the data from A\$ into the record.
close the relative-file #2.

RENAME

TYPE: I/O Statement (BASIC4.0)

FORMAT: RENAME "old filename" TO "new filename" [,Ddrive number] [<ON|,>Udevice number**]**

Action: Change the name of a file on disk.

EXAMPLES of RENAME Statement:

```
RENAME "TEST" TO "FINALTEST",D0  
RENAME (A$) TO (B$),D0,U9
```

Change the name of the file "TEST" to "FINALTEST".
Change the filename specified in A\$ to the filename specified in B\$ on drive 0, device number 9.
Remember, whenever a variable name is used as a filename, it must be enclosed in parentheses.

SCRATCH

TYPE: I/O Statement (BASIC4.0)

FORMAT: SCRATCH "filename" [,Ddrive number] [<ON|,>Udevice number**]**

Action: Delete file from the disk directory.

EXAMPLE of SCRATCH Statement:

```
SCRATCH "MY BACK", D0
```

This erases the file MY BACK from the disk in drive 0.

WEDGE

TYPE: Statement (Bradley's BASIC only)

FORMAT: WEDGE **<ON|OFF>**

Action: Enable or disable the DOS-WEDGE functionality.

EXAMPLES of WEDGE Statement:

WEDGE ON
WEDGE OFF

Enables the DOS-WEDGE functionality.
Disables the DOS-WEDGE functionality.

DOS WEDGE