BRADLEY'S BASIC

PROGRAMMER'S REFERENCE GUIDE

by
Matthew Desmond

FactorOfMatt
factorofmatt.com

A companion to BRADLEY'S BASIC V1.0

SECOND EDITION
FIRST PRINTING-November 2023
SECOND PRINTING-January 2024

TABLE OF CONTENTS

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 synax (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:

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

EXAMPLES OF ACTUAL STATEMENTS:

```
COPY D0, "TEST" TO D1, "TEST PROG"
COPY D0, "STUFF" TO D1, "STUFF"
COPY D0 TO D1
COPY "WORK.PROG" TO "BACKUP"
```

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 under- standing as to how syntax examples are presented.

SYMBOL	EXAMPLE	DESCRIPTION
<logical-file-number>	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 '80s 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 witten by me sometime between 1983 and 1987. I did take some time to make the code better and more compatible with the actual BASIC 4.0/7.0 versions.

The code for **BRADLEY'S BASIC** operates in the $8000 8K ROM address-block ($8000-$9FFF). The extra-commands make extensive use of the cassette buffer area ($033C-$03FB) which may interfere with any small programs that also happen to use that space.

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.

Please visit the Factor Of Matt website at factorofmatt.com. If you find a bug or a documentation error please drop me a line (details on the website).

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.

Additionally some aspects of the existing BASIC2.0 language have been modified slightly to integrate with the new capabilities.

BRADLEY'S BASIC Keywords

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╳	
HEADER	H SHIFT E	H▔	

COMMAND	ABBREVIATION	SCREEN	FUNCTION TYPE
HELP	HE SHIFT L	HE└	
INIT	IN SHIFT I	IN╮	
OLD	O SHIFT L	O└	
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

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 calied "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:
- start address is the starting address where the program is LOADed to.

Action: Load a binary file into memory starting either at the address it was saved-from or as specified.

EXAMPLES of BLOAD Statement:

BLOAD "SPRITES"	LOADS the binary file "SPRITES" to the address it was saved from.
BLOAD "DATA1", D0, U8, P4096	LOADS the binary file "DATA1" into address 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.

> NOTE: If the first character of the filename is @ then the file will replace an existing file with the same name (known as save-and-replace). This can be dangerous in certain situations.

EXAMPLES of BSAVE Statement:

BSAVE "SPRITE DATA",P3584 TO P4096

Saves the binary file named "SPRITE DATA", starting at address-range 3584 through 4095.

BSAVE "PROGRAM.SCR",DO,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

TYPE: I/O Statement (BASIC4.0)
FORMAT: CATALOG [Ddrive number][<ON|,>Udevice number][,wildcard string]

Action: Display the disk directory. The CATALOG command is the same as the DIRECTORY command.

EXAMPLE of CATALOG Statement:

CATALOG	Displays the disk directory on drive 0.

COLD

TYPE: Statement (Bradley's BASIC only)
FORMAT: COLD

Action: Restart the Commodore-64. This command performs the same operation as the traditional SYS 64738 does. This command may only be run from direct-mode (not running a program). Additionally, the "ARE YOU SURE?" prompt will be displayed and the machine will only restart if the response is Y

EXAMPLE of COLD Statement:

COLD	Restarts the computer.

COLLECT

TYPE: I/O Statement (BASIC4.0)
FORMAT: COLLECT [Ddrive number][<ON|,>Udevice]

Action: Free inaccessible disk space.

EXAMPLE of COLLECT Statement:

COLLECT D0	Free all available space which has been incorrectly allocated to improperly closed files. Such files are indicated with an asterisk on the disk directory.

CONCAT

TYPE: I/O Statement (BASIC4.0)
FORMAT: CONCAT "file 2" [,Ddrive number] TO "file 1"[,Ddrive number][<ON|,>Udevice]

Action: Concatenate two data files.

NOTE: CONCAT only works with sequential (SEQ) or user (USR) file-types.

EXAMPLES of CONCAT Statement:

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).

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.

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
```

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 l.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.

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 "N0: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.

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
- record length is the length of records in a relative file only.
- W = indicates a write operation (otherwise 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          Displays the most recent disk error code.
   73
```

DS$

TYPE: String Function (BASIC4.0)
FORMAT: DS$

Action: This function will return the disk error string 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$         Displays the most recent disk error string.
   73,CBM DOS V2.6 1541,00,00
```

DSAVE

TYPE: I/O Statement (BASIC4.0)
FORMAT: DSAVE "filename" [,Ddrive number] [<ON|,>Udevice number]

Action: Save a BASIC program file to disk

> NOTE: If the first character of the filename is @ then the file will replace an existing file with the same name (known as save-and-replace). This can be dangerous in certain situations.

EXAMPLES of DSAVE Statement:

DSAVE "BANKRECS"	Saves the program "BANKRECS" to disk.
DSAVE A$	Saves the disk program named in the variable A$.
DSAVE "PROG3",D1,U9	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.

> NOTE: To verify binary data, use the usual BASIC 2.0 command: VERIFY "filename",device,1

EXAMPLES of DVERIFY Statement:

DVERIFY "C64"	Verifies program "C64" on drive 0, device 8.
DVERIFY "SPRITES",D0,U9	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 "MYDISK" 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 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-NEW a program.

RECORD

TYPE: I/O Statement (BASIC4.0) [NOT IMPLEMENTED]
FORMAT: RECORD# logical file number, record number [,byte number]

Where:

- record number is the desired record number to access.
- byte-number is the offset into the desired record.

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"	open existing relative-file called FILE as file 2.
20 RECORD#2,10,1	position relative-file pointer to first byte in record number 10.
30 PRINT#2,A$	write the data from A$ into the record.
40 DCLOSE#2	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
Change the name of the file "TEST" to "FINALTEST".

RENAME A$ TO B$,D0,U9
Change the filename specified in A$ to the filename specified in B$ on drive 0, device number 9.

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

Modifications to BASIC 2.0

RUN

TYPE: Command (BASIC7.0)
FORMAT: RUN [line number]
RUN "filename" [,Ddrive number] [<ON|,>Udevice number]

Action: Execute BASIC program.

EXAMPLES of RUN Statement:

RUN	Starts execution from the beginning of the program.
RUN 100	Starts program execution at line 100.
RUN "PRG1"	DLOADs "PRG1" from drive 0, device 8 and runs it from the beginning of the program.
RUN A$	DLOADs the program named in the variable A$ and runs it from the beginning of the program.

Modified auto-load and run

TYPE: Keyboard Shortcut (BASIC2.0)

Action: Pressing the RUN key (aka SHIFT-STOP aaka SHIFT-RUN/STOP) will now load and run the first program from device 8, unit 0. This is equivalent to entering:

```
RUN "*"
```

Differences from BASIC 4.0/7.0

Every attempt has been made to make **BRADLEY'S BASIC** as compatible with BASIC 4.0/7.0 as possible.

The BASIC token-codes for the BASIC 4.0-extensions should be the same as the real BASIC 4.0 token-codes. BASIC programs written in BASIC 4.0 have a decent chance of working directly in **BRADLEY'S BASIC** (aside from those that use the unimplemented commands).

Rather annoyingly, BASIC 7.0 doesn't maintain this compatibility so token-code wise, **BRADLEY'S BASIC** isn't compatible. Programs will have to be modified slightly to correct these differences. See the appendices for a comparison of keywords and their tokens for BRADLEY'S BASIC, BASIC 4.0 and BASIC 7.0

Deletions:

The following BASIC4.0/7.0 commands are not (currently) implemented:

APPEND, **DCLEAR**, **DCLOSE**, **DOPEN** and **RECORD**.

Honestly I couldn't be bothered to implement these just yet. I'll likely implement these in the next major release.

Differences:

Generally speaking, the Commodore BASIC 4.0/7.0 disk-related commands require that string-variables for filenames be placed in parenthseses. The implementation of these commands in **BRADLEY'S BASIC** does not require the parentheses.

NOTE: I'm sure COMMODORE had a good reason for their choice here, but for now I haven't figured-out (or remembered) why. Likely it has something to do with some ambiguity when parsing the command.

The DS and DS$ variables are not aware of disk operations performed with non-**BRADLEY'S BASIC** file commands (**SAVE**, **LOAD**, **OPEN**, **CLOSE** etc). I'll likely correct this in the next major release.

DOS WEDGE

Introduction

A dos-wedge is a traditional way to expand the capabilities of COMMODORE 8-bit computers reaching all the way back to the PET line of computers. **BRADLEY'S BASIC** gladly continues this tradition. Attention has been paid to help ensure that this implementaion of the dos-wedge is compatible with other versions: if you are familiar with an existing one then this one should work for you too.

NOTE: The **BRADLEY'S BASIC** dos-wedge commands only work in direct-mode – they cannot be used within BASIC programs. Fortunately, however, this is not a problem since **BRADLEY'S BASIC** comes complete with BASIC4.0/7.0 disk commands which do work inside programs.

Wedge Filenames

The LOAD, SAVE and VERIFY dos-wedge commands all require a filename. Unlike the BASIC4.0/7.0 commands, the dos-wedge does not require the filename to be surrounded in quotes. If, however, the filename is surrounded in quotes, then anything else on the input-line is ignored. This is particularly useful when you wish to load a file directly after listing a directory. Simply cursor-up to the beginning of the line containing the filename and type the load-command character and press enter. Suppose **$** shows:

```
0 "EXCELLENT DISK" ED 2A
20  "PROG1"          PRG
10  "PROG2"          PRG
634 BLOCKS FREE.
```

Then placing a **/** at the start of the line containing **PROG2** and pressing enter will load the file:

```
/0  "PROG2"          PRG
```

Wedge Commands

$, >$ or @$ (Directory)

FORMAT: $[drive:][wildcard]
>$[drive:][wildcard]
@$[drive:][wildcard]

Action: List the disk directory to the screen.

EXAMPLES of Directory Statement:

$	Display the directory from the default device.
@$BB*	Display all files beginning with "BB".
>$1:	Display the directory from drive 1 on the default device.

> or @ (Status)

FORMAT: >
@

Action: Read and display the disk error channel from the default device.

EXAMPLES of Status Statement:

>	Display error status.
@	Display error status.

> or @ (Command)

FORMAT: >[command]
@[command]

Action: Send the command to the default-drive's command channel.

NOTE: No processing is performed upon the command (such as adding the drive number) so the command is sent exactly as typed.

EXAMPLES of Command Statement:

>S:FILE1	Scratch "FILE1" from drive 0 on default device.
@V	Send a validate (collect) command to the default device.

## ># or @# (Device)

FORMAT: >#[device number]
@#[device number]

Action: Select the device number to be used for dos-wedge commands. Device numbers may range from 4 to 30

EXAMPLES of Device Statement:

>#9	Select device 9 for future wedge operations.
@#	Select device 8 for future wedge operations.

>O or @O (Old)
FORMAT: >O
@O

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

NOTE: The **>O or @O** 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:

>O	Un-NEW a program.
@O	Un-NEW a program.

>Q or @Q (Quit)
FORMAT: >Q
@Q

Action: Disable the dos-wedge functionality of **BRADLEY'S BASIC**.

NOTE: If you wish to re-enable the dos-wedge, then simply enter the **BRADLEY'S BASIC** command:
WEDGE ON

EXAMPLES of Quit Statement:

>Q	Disable the dos-wedge.
@Q	Disable the dos-wedge.

/ (Load BASIC program)
FORMAT: /"filename"
/filename

Action: Load a BASIC program from the default device. This command is similar to **DLOAD"filename"**.

EXAMPLES of Load BASIC Statement:

/"PROGRAM"	LOAD the BASIC program, "PROGRAM".
/PROGRAM	LOAD the BASIC program, "PROGRAM".

' (Verify BASIC program)

FORMAT: ' "filename"
'filename

Action: Verify the BASIC program from the default device. This command is similar to **DVERIFY"filename".**

EXAMPLES of Verify BASIC Statement:

' "PROGRAM"	VERIFY the BASIC program, "PROGRAM".
'PROGRAM	VERIFY the BASIC program, "PROGRAM".

← (Save BASIC program)

FORMAT: ←"filename"
←filename

Action: Save a BASIC program to the default device. This command is similar to **DSAVE"filename".**

NOTE: If the first character of the filename is @ then the file will replace an existing file with the same name (known as save-and-replace). This can be dangerous in certain situations.

EXAMPLES of Save BASIC Statement:

←"MYPROG"	SAVE the BASIC program as "MYPROG".
←@MYPROG	SAVE-and-REPLACE the BASIC program as "MYPROG".

↑ (Load and RUN BASIC program)

FORMAT: ↑"filename"
↑filename

Action: Load and RUN a BASIC program from the default device. This command is similar to **DLOAD"filename"** followed by **RUN**.

EXAMPLES of Load and RUN BASIC Statement:

↑"PROGRAM"	LOAD and RUN the BASIC program, "PROGRAM".
↑PROGRAM	LOAD the RUN BASIC program, "PROGRAM".

% (Load binary program or file)

FORMAT: %"filename"
%filename

Action: Load a binary (machine code) program from the default device. The binary program is loaded to the address the program was saved from. This command is similar to **BLOAD"filename"**.

EXAMPLES of Load binary Statement:

%"SPRITES"	LOAD the binary file, "SPRITES".
%SPRITES	LOAD the binary file, "SPRITES".

£ (Load and execute binary program or file)

FORMAT: £"filename"
£filename

Action: Load a binary (machine code) program from the default device and execute the loaded binary. The binary program is loaded to the address the program was saved from and executed starting at that address. This command is similar to **BLOAD"filename"** followed by **SYS load-location**.

EXAMPLES of Load and execute binary Statement:

£"MCODE-PROG"	LOAD then execute binary file, "MCODE-PROG".
£MCODE-PROG	LOAD then execute binary file, "MCODE-PROG".

APPENDICES

KEYWORD TO TOKEN-CODE NUMBERS COMPARISON

A comparison of BRADELY'S BASIC, BASIC 4.0 and BASIC 7.0 keywords and their token-code numbers:

KEYWORD	BRADLEY'S BASIC	BASIC 4.0	BASIC 7.0
CONCAT	204	204	254+145
DOPEN	205	205	254+139
DCLOSE	206	206	254+141
RECORD	207	207	254+144
HEADER	208	208	241
COLLECT	209	209	243
BACKUP	210	210	246
COPY	211	211	244
APPEND	212	212	254+140
DSAVE	213	213	239
DLOAD	214	214	240
CATALOG	215	215	254+138
RENAME	216	216	245
SCRATCH	217	217	242
DIRECTORY	218	218	238
BSAVE	219	-	254+142
BLOAD	220	-	254+143

KEYWORD	BRADLEY'S BASIC	BASIC 4.0	BASIC 7.0
DVERIFY	221	-	254+146
DCLEAR	222	-	254+147
DISK	223	-	-
INIT	224	-	-
COLD	225	-	-
OLD	226	-	-
WEDGE	227	-	-
HELP	228	-	234 (unrelated)

TOKEN-CODE NUMBERS TO KEYWORD COMPARISON

A comparison of command token-code numbers and their BRADELY'S BASIC, BASIC 4.0 and BASIC 7.0 keywords.

TOKEN	BRADLEY'S BASIC	BASIC 4.0	BASIC 7.0
204	CONCAT	CONCAT	RGR
205	DOPEN	DOPEN	RCLR
206	DCLOSE	DCLOSE	RLUM
207	RECORD	RECORD	JOY
208	HEADER	HEADER	RDOT
209	COLLECT	COLLECT	DEC
210	BACKUP	BACKUP	HEX$
211	COPY	COPY	ERR$
212	APPEND	APPEND	INSTR

TOKEN	BRADLEY'S BASIC	BASIC 4.0	BASIC 7.0
213	DSAVE	DSAVE	ELSE
214	DLOAD	DLOAD	RESUME
215	CATALOG	CATALOG	TRAP
216	RENAME	RENAME	TRON
217	SCRATCH	SCRATCH	TROFF
218	DIRECTORY	DIRECTORY	SOUND
219	BSAVE	-	VOL
220	BLOAD	-	AUTO
221	DVERIFY	-	PUDEF
222	DCLEAR	-	GRAPHIC
223	DISK	-	PAINT
224	INIT	-	CHAR
225	COLD	-	BOX
226	OLD	-	CIRCLE
227	WEDGE	-	GSHAPE
228	HELP	-	SSHAPE