

DBASIC Commands

New Facilities

There are four new commands and six new statements available in DBASIC. They are (with example filenames):

DSAVE "n:FRED" <CR>

DLOAD "n:JOE" <CR>

DELFNP "n:PROG"

DELFND "n:DATA"

DOPENR "n:DATAIN" (program mode only)

DOPENW "n:DATOUT"

DCLOSER

DCLOSEW

DINPUT

DPRINT

All filename specifications must be expressed as a valid BASIC string, either by enclosing the filenames in quotes or by specifying a string variable containing the filename e.g.

A\$="2: FRED"

DSAVE A\$

will work correctly. String expressions may also be used.

DBASIC also provides a facility for executing TANDOS Commands from within BASIC simply by prefixing the TANDOS command with DOS, eg:

DOSDIR 0:

DSAVE

This command is used to save the entire BASIC program as a disc file. It requires a disc filename specification (and optional drive number) expressed in the usual format except that the extension is given by the system and will always be '.BAS'. You will get a syntax error if you try to specify your own file extension. DSAVE may also be used from within a program.

A typical example would be:

DSAVE "2:MYFILE" <CR>

DLOAD

This command is the complement of the DSAVE command and is used to load a new program into memory from disc. Like DSAVE it needs a file specification in order to know which file to load.

DLOAD may also be used from within a program. e.g. DLOAD "0:PIRATE"

DELFNx

There are two versions of DELFN, DELFNP and DELFND. These are used to delete Programs or Data files created by BASIC (in fact any file having the '.BAS' or '.BDT' extension).

e.g.

```
DELFND "2:DATA"      or
A$="2:DATA"
DELFND A$
```

```
DELFNP "2:PROG"     or
B$="2:PROG"
DELFNP B$
```

An open file is automatically closed before being deleted. DELFNx may be issued in immediate mode or may be included in a program.

DOPENX

There are two versions of the DOPENX statement - DOPENR and DOPENW

The DOPENx statements are used to indicate to the system which files will be read from/written to by the DINPUT and DPRINT statements.

e.g.

```
DOPEN "2:ADDRES"    or
A$="2:ADDRES"
DOPENR A$
```

indicates that DINPUT statements should read the contents of the file 2:ADDRES.BDT

```
DOPENW "0:EXAMPL"  or
A$= "0:EXAMPL"
DOPENW A$
```

will cause DPRINT statements to write to the file 0:EXAMPL.BDT

As with DLOAD and DSAVE, only the filename, not the extension, is needed. The system supplies the extension '.BDT' (Basic DaTa) for you. Any attempt to specify an extension will result in a syntax error.

DCLOSEX

Again there are two versions of this statement — DCLOSER and DCLOSEW. They are the complements of DOPENX statements. The correct use of DCLOSE is especially important for files being written by the DPRINT statement as the DCLOSEW causes the system to write out any remaining data left unwritten in the output buffer and then tidies up the disc file and enters the relevant information into the disc directory. The effect of failing to issue the DCLOSEW will be

that all the data written to the file will be lost and the file will not exist on the disc (this does not harm the other information on the disc). DCLOSEX will return successfully even if a file has not been Opened. It is not possible to issue a second DOPENR or DOPENW until the appropriate DCLOSEX has been issued. The effect of this is that there can be only two disc files open simultaneously; one for input and one for output.

On input you can effectively 'rewind' a file back to the beginning by closing and then re-opening the same file. DOPEN R, DCLOSE R and DOPENW do not write to the disc. DCLOSEW will only write to the disc if a DRINT statement has been issued since the DOPENW (it is not possible to generate empty files). Both of the DOPENX statements read from disc.

DINPUT and DPRINT

These two statements operate in exactly the same way as INPUT and PRINT with the sole exception that I/O is with the disc and not the keyboard/VDU. Reference should be made to the BASIC manual for more information. Note that the same requirements regarding separators (commas) apply to disc I/O as they do to cassette I/O.

As for cassette I/O only sequential files are currently supported by TANDOS 65 v1.0.

The DPRINT statement makes use of the user programmable output facility (Tanbug V2 onwards) whereby the address of a user provided output routine is placed in the slow interrupt vector. Consequently, slow interrupts and user programmed output are not available from Disc BASIC.

Example Programs

The first program generates a file with a single column of numbers from 1 to 10.

```
10 DOPENW "0:EX01"  
20 FOR I = 1 TO 10  
30 DPRINT I  
40 NEXTI  
50 DCLOSEW
```

If you want to see the numbers as they are written to disc then insert the line 35 PRINT I.

This second example reads the column of numbers from the first program and adds a second column of numbers equal to 100 times the first column. This is intended to demonstrate, in very simple terms, the common 'input - process - output' type of application.

```
NEW  
10 DOPENR "0:EX01"  
20 DOPENW "0:EX02"  
30 FOR I = 1 to 10  
40 DINPUT J  
50 DPRINT J,"",100*I  
60 NEXTI  
70 DCLOSEW  
80 DCLOSER
```

Adding line 45 PRINT J will display the values read from file.

This last program reads the second data file we generated (with two columns of data) and displays them on the screen.

```
NEW
10 DOPENR "0:EX02"
20 FOR I = 1 to 10
30 DINPUT J,K
40 PRINT J,K
50 NEXTI
60 DCLOSER
```