**Interactive Microware, Inc.**
**P.O. Box 771**
**State College, Pa 16801**
**(814) 238-8294**

## * * * D O S + F O R N O R T H S T A R D I S K * * *

DOS+ enables any program to execute all North Star disk commands and all Meca tape commands.

DOS+ provides a simple protocol for transfer of ASCII data between programs and I/O devices.

DOS+ allows your BASIC programs to list the directory and create or delete disk files.

DOS+ also works with assembler language programs or high level languages with only minor changes.

DOS+ permits any program to create a list of operations in memory and then execute them once or repeatedly. The list of commands can also be passed to another program.

DOS+ requires no changes in DOS or your present I/O routines. It runs in less than 1K bytes of memory, starting at 1C00, 5C00, 6C00 or 7C00H.

If you are using the North Star floppy disk, DOS+ will significantly increase its capabilities. Any program that uses the DOS routines for character input and output can take advantage of the power of DOS+. In order to execute a DOS+ command,you merely type or print a control S (here denoted <S>), followed by one of the following DOS+ commands:

<S>N... means execute any North Star disk command (...).
<S>M... means execute any Meca tape command (...).
<S>A... defines the Address (...) of a data list in memory.
<S>W... means Write output data (...) into a data list.
<S>J... means Jump over a number (...) of lines.
<S>P means Print the next line from a data list.
<S>R means Read input data from a data list.
<S>T Terminates the W and R commands and resumes normal I/O.
<S>S sets the Scratch area address for DOS.
<S>D means interpret numbers as Decimal.
<S>H means interpret numbers as Hexadecimal.
<S>E means Exit from the DOS+ extension.

(OVER...)

**Interactive Microware, Inc.**
**P.O. Box 771**
**State College, Pa 16801**
**(814) 238-8294**

# SYSKIT

Would you like to develop programs in a manageable modular form so you could re-use the modules with a minimum of rewriting?

Would you like to be able to choose the modules that do the job and then link them together with two commands?

Would you like to let your computer do the messy job of keeping track of its own memory and free you up to think and create?

Would you like all of this power in 2½k bytes of memory?

Then you would like Syskit.

Syskit is a sophisticated, but easy to use, development tool. With it you can put together very large programs with a minimum of hardware, if you have an 8080 or Z-80 based system that allows RST 1,RST 2, and RST 3. Syskit does not require a special assembler or loader, so you don't have to throw out the ones that you are familiar with.

FEATURES

**Syskit is based on a concept of modules that may be selected at will. You use our table of functions to choose those that do the job. You can even write your own modules.

**Syskit offers a linkage function that lets you add, delete, or change not only the modules that we supply, but also those that you write, as long as you write them to Syskit's easy conventions.

**Syskit offers Dynamic Memory Management, which keeps track of what module is using what memory for what purpose. This feature is normally found on Mainframes and has only recently become common on Minis. Now, fro the first time, it is available for Micros.

**Syskit's monitor lets you examine, alter, and list memory either from absolute address or relative to the start of a module. Syskit will keep track of which modules are linked; where they are in memory; and how long they are. But best of all, the monitor is particularily well suited for top down program development.

**Syskit will fit in 4k of memory and still leave 1.5k for your own programs.

DOS+ UPDATE #1--APPENDIX I

Please replace page 16 of the DOS+ manual with this one. With these minor changes, you will be able to use DOS+ with the new North Star DOS version 4.

## Modified Versions of DOS+

The diskette supplied with DOS+ includes four different versions which are compiled for locations 7C00(DOS+7C), 6C00(DOS+), 5C00(DOS+5C) and 1C00(DOS+1C). If none of these versions are satisfactory, you should contact Interactive Microware for information about custom versions. If you are using DOS version 3, you can use any of these versions of DOS+ by typing "GO DOS+XX", where XX stands for the last two letters of the name.

If you are using DOS version 2, you will have to alter the following locations (X is the first digit of the start of DOS+):

XDAD=C5 XDCF=C7 XDD2=FD XE4B=21 XE4D=22 XE4E=48 XE4F=23 XE50=22 XE51=5B XE52=23 XE53=22 XE54=B5 XE55=23

For DOS version 4, the following changes are required:

XDC0=5D XDC6=05 XDC7=21 XDCC=CF XDD2=FA XDD5=AF XDDD=94 XDE0=CF XDE3=05 XDE4=21 XDED=94 XE4B=8B XE52=EF.

If you have located your Meca TOS anywhere other than 7000H, you should change the following locations in DOS+ (X is the first digit of the start of DOS+ and Y is the first digit of the start of your TOS):

XE95=YF XEA8=YF XEAB=YF XEAE=Y3

Feel free to call me at 814-863-0074 or 814-238-8294 if you have any questions or comments on DOS+.

Paul K. Warme

```
LIST

1!"NORTH STAR DISK/MECA TAPE BACKUP SYSTEM"
2!"COPYRIGHT (C) 1978 BY INTERACTIVE MICROWARE, INC."
3REM  BEFORE RUNNING THIS PROGRAM, INITIALIZE THE TAPE OPERATING
4REM SYSTEM AND GO DOS+, GO BASIC AND LOAD THIS PROGRAM.
5REM  TO TRANSFER FROM DISK TO TAPE, MOUNT THE TAPE ON DRIVE Ø,
6REM AND INITIALIZE IF IT IS A NEW TAPE "^SMNE :Ø".  THEN RUN
7REM THIS PROGRAM WITH THE DISK TO BE SAVED ON DRIVE 1.
8REM  TO RESTORE FROM TAPE TO DISK, MOUNT A FRESH DISK ON DRIVE 1
9REM AND THE BACKUP TAPE ON DRIVE Ø "^SMMO :Ø" AND THEN RUN
10REM THIS PROGRAM.
11REM  TWO FULL DISKS CAN BE STORED ON EACH SIDE OF A CASSETTE.
15S$=CHR$(19)
20N$=S$+"N"\M$=S$+"M"
30INPUT "DISK # ?",N\F$=STR$(N)
35IF N=Ø THEN END
40L=LEN(F$)\IF L<2 THEN F$="Ø"+F$
50F$(1,1)="D"
100INPUT "DISK TO TAPE(1) OR TAPE TO DISK(-1)?",D
110IF D=1 THEN !N$,"LI"
120FOR I=1 TO 12
130B=30\IF I=12 THEN B=20
140B0=INT(30*(I-1))
150B$=STR$(B0/10)\L=LEN(B$)
160IF L<3 THEN B$(1,1)="Ø" ELSE B$=B$(2,3)
170IF D=-1 THEN 210
175!N$,"RD",B0," Ø",B
180IF B=30 THEN 200
190!M$,"SAVE ",F$,B$," :Ø Ø 13FF"
195GOTO 300
200!M$,"SAVEQ ",F$,B$," :Ø Ø 1DFF"
205GOTO 300
210!M$,"LOAD ",F$,B$," :Ø Ø"
220!N$,"WR",B0," Ø",B
300NEXT I
310IF D=-1 THEN !\!N$,"LI"
320GOTO 20
READY
```

DOS+ for North Star Disk

TABLE OF CONTENTS

# DOS+ for North Star Disk

by

William LaForge and Paul Warme

## INTRODUCTION

DOS+ will significantly expand the capabilities of your North Star floppy Disk Operating System (DOS). It allows any program to control the floppy disk by merely printing two special command letters, followed by the usual DOS command. If you own a Meca tape system, DOS+ gives you the same sort of control over the Tape Operating System (TOS). In fact, DOS+ provides a simple protocol for transferring ASCII data from any mass storage device or program to any other mass storage device or program.

When the DOS+ extension is activated, all input and output characters are "read" by DOS+, but nothing unusual happens until a control S (here denoted <S>) comes along. The letter after the <S> tells DOS+ what you want to do with subsequent input or output to your terminal.

Although each DOS+ command will be explained in detail later, here is a brief rundown of some of them. If the command letter is N, the rest of the line is interpreted as a command for the North Star DOS. If the command letter is M, the rest of the line is interpreted as a command for the Meca TOS.

The command A defines the Address of a data list stored in memory. A data list consists of one or more lines of ASCII characters, ending with a carriage return (here denoted <CR>).

Each line may contain either data or a command to be passed to another program. The command letter W means Write the subsequent output characters into a data list, and the R command causes subsequent input to be Read from a data list. If you want to Print the next line from the data list, the P command will do this. The J command Jumps over lines in the data list and the T command Terminates an R or W command.

When you type <S>, your terminal will print a "$" and DOS+ will "listen" for the rest of the command. All characters that are typed as part of the command are printed by your terminal, but they are ignored by your program. If you make an error while typing a DOS+ command, you can erase one character at a time by typing one or more underlines (left arrow on some terminals). You can erase the entire command line by typing "@". Any undefined command letter after a <S> will be ignored and the bell on your terminal will signal the error.

It is very easy to activate DOS+. Just bootstrap the disk operating system at location E900, mount the DOS+ diskette and type GO DOS+. Appendix I contains special instructions for users of DOS version 2 and for loading DOS+ at addresses other than the standard address (6C00 to 6FFF). DOS+ can be deactivated at any time by typing control S (<S>), the letter "E" for Exit and then a carriage return (<CR>). All DOS+ commands are available to any program that uses your DOS I/O routines for terminal input (CALL 2010H) and output (CALL 200DH).

## DESCRIPTION AND EXAMPLES OF DOS+ COMMANDS

### The N Command (North Star DOS)

If you want to execute a DOS command, you should type or print <S>N, followed by the DOS command and a <CR>. Like all other DOS+ commands, you can either type this command while your program is waiting for input or you can have your program print the command. Since DOS+ "listens" to both input and output data, either form of command is permitted.

You should not use the N command while you are working in DOS because this will probably cause the system to bomb. However, you can use the regular form of DOS command in this case.

If an error occurs during execution af the N command, your program will abort to DOS. The DOS error routine will print "?" and DOS will then wait for a new command after the "*" prompt.

EXAMPLE: While your program is waiting for input, list the directory of the disk on drive 1 (the default drive).

```
<S>NLI<CR>
```

EXAMPLE: In your BASIC program, load file DATA at location 1000. Notice that you don't have to print a <CR> after the command, since the PRINT statement automatically prints <CR> at the end of each line. CHR$(19) is the same as <S>.

```
PRINT CHR$(19),"NLF DATA 1000"
```

The M Command (Meca TOS)

Any Meca tape operating system command can be executed by typing or printing <S>M, the normal TOS command, and then a <CR>. In conjunction with other DOS+ commands, you can transfer files from disk to tape, or vice versa, and you can also write programs in BASIC that can read and write data on tapes (see the R, W and P commands).

If an error occurs while the TOS is in control, the standard TOS error message will be printed. At this point, you should type <CR> to continue; your program will be stopped and TOS will gain control.

EXAMPLE: While your program is waiting for input, list the directory of the tape on drive Ø.

```
<S>MDIR :Ø<CR>
```

EXAMPLE: In your BASIC program, save file DATA from locations 1ØØØ to 2ØØØ on drive 1.

```
PRINT CHR$(19),"MSAVE DATA :1 1ØØØ 2ØØØ"
```

The A Command (Address of data list)

Before using the W, P, R and J commands (see below), you must define the address for a data list. To do this, you should type or print <S>A, followed by the desired address. This address will be updated automatically after each transfer to or from the data list, thus allowing your programs to access each line of the data list in sequence. The address will be interpreted as either

a hexadecimal number or a decimal number, in accordance with the last H or D command issued (see below). If you want to access lines randomly within a data list, you will find the J command very useful.

EXAMPLE: You can define the address of a data list to be location 1000 by typing:

<S>A1000<CR>

or by including the following statement in your BASIC program:

PRINT CHR$(19),"A1000"

The D Command (Decimal number mode)

The numbers typed or printed after the A, J and S commands will be interpreted as decimal numbers if you type <S>D<CR> or insert the statement PRINT CHR$(19),"D" in your BASIC program.

The H Command (Hexadecimal number mode)

This is similar to the D command, but causes subsequent numbers to be interpreted as hexadecimal numbers. The most recent D or H command determines the number base; when DOS+ is initialized, the hexadecimal mode is set.

The W Command (Write output into a data list)

After defining the address of your data list with the A command, you can write information into memory by typing or printing <O>W. Everything that is output after the W is stored

in memory, instead of being printed on your terminal. The W command terminates when a control T (<T>) is output or when you issue the T command (see below). When you type <T> on your terminal, it will be printed as "%". The <T> is stored in memory, but all subsequent output will be printed on your terminal. Bear in mind that the W command will be terminated if any other DOS+ command is issued, as well. The W command is useful for temporary storage of data which will later be read by another program or will be written on a "foreign" mass storage device. It is also capable of creating macro commands (see Application Ideas).

EXAMPLE: Start a data list at location 1000 and write one line.

```
S$=CHR$(19)
PRINT S$,"A1000"
PRINT S$,"WThis is the first line of the data list."
```

EXAMPLE: Write another line in the data list and then return to normal output. Suppose that N denotes the quantity, C the catalog number and P the price of peaches. The data will be formatted for later INPUT using the R command.

```
N$=STR$(N)\N$=N$(2)+","
C$=STR$(C)\C$=C$(2)+","
P$=STR$(P)\P$=P$(2)+","
PRINT N$,C$,P$,"PEACHES"
PRINT CHR$(20),"This line will be printed on the terminal."
```

The P Command (Print the next line from a data list)

If you want to print one line from a data list on your terminal, you should type or print <S>P<CR>. If you want to print the first line, you should use the A command to define the starting address; otherwise, the address is automatically advanced to the beginning of the next line after each line is printed. If the data list is properly terminated with a <T>, the P command will print just a carriage return when it reaches the end of the list.

EXAMPLE: Print the first two lines from a data list at location 1000. You will recall that "!" is a shorthand for PRINT and "\" separates multiple statements on a line. An extra PRINT is included because the P command doesn't print a carriage return.

```
S$=CHR$(19)\PRINT S$,"A1000"
FOR I=1 TO 2\!S$,"P"\!\NEXT I
```

The R Command (Read input from a data list)

After you have defined the address of a data list using the A command, you can read the data into your program by typing or printing <S>R<CR>. Thereafter, all input will be read from the data list, instead of coming from the keyboard of your terminal. As the data is read from memory, each character will be printed by your program, just as if you had typed it on your terminal. Normal input from your keyboard will resume when a <T> is found in the data list or when your program sends the T command (see below) to DOS+. The R command will also be terminated if any

other DOS+ command is issued.

The R command is very useful for reading data stored by other programs or data input from a "foreign" mass storage device. It is also capable of executing macro commands (see Application Ideas). Together, the R and W commands provide a simple and effective mechanism for transferring ASCII data among various programs and I/O devices.

EXAMPLE: Read and print the first line of the data list at location 1000 that was created earlier by the W command.

```
S$=CHR$(19)\!S$,"A1000"
!S$,"R"
INPUT T$
```

EXAMPLE: Read the next line of the data list as four separate items: the number of items (N), catalog number (C), price (P) and description (D$).

```
INPUT N,C,P,D$
```

Notice that subsequent input will be from the terminal, since the next character in the data list was <T>.

The J Command (Jump over lines in a data list)

When you are reading lines from a data list, it is often desireable to be able to access any line (i.e., random access) conveniently. Of course, this could be done by setting the address to the first line and then reading lines until the right line is found. Another way this could be done is to have your program remember the starting address of each line and then use the A command to address the desired line (this is the preferred method when each line is the same length).

The J command provides a quick way to address any line; it is particularly useful when lines are unequal in length. The number typed or printed after the J determines how many lines will be skipped, starting from the present position in the list. If the number after the J is Ø or is omitted, the address is set to the beginning of the previous line. This feature is useful when you want to read a line, alter it and then write it back in the same position in the data list. If you try to jump over more lines than there are in the data list, the address will be set to the location of the <T> at the end of the data list. To avoid an error condition, you should not use the JØ command after reading the first line of the data list. Instead, you should use the A command to return to the start of the first line.

One method for accessing any line in a data list is to set the address to the beginning of the data list and then jump over the required number of lines. If the desired line is beyond the previous line read from the data list, it will be faster to

subtract the previous line number from the desired line number and then jump over that number of lines minus one.

EXAMPLE: Read a line from a data list, alter it and then write it back in the same position.

```
S$=CHR$(19)\!S$,"A1000"\!S$,"J1"
!S$,"R"\INPUT T$\T$(1,4)="1978"
!S$,"J0"\!S$,"W",T$
```

EXAMPLE: Read the 10th line of a data list starting at 1000.

```
S$=CHR$(19)\!S$,"A1000"
!S$,"J9"\!S$,"R"
INPUT T$
```

EXAMPLE: Read the 20th line of the data list after the previous example.

```
N=20-10-1\N$=STR$(N)\N$=N$(2)
!S$,"J",N$\!S$,"R"
INPUT U$\!S$,"T"
```

The T Command (Terminate the R or W command)

If you want to terminate I/O from a data list during a Read or Write command, you should insert (PRINT CHR$(19),"T") in your BASIC program. After this statement is executed, I/O will communicate with your terminal, rather than with the data list.

The T command does not affect the current address in the data list, so you can return to the data list later by printing <S>R<CR> or <S>W<CR>. You will recall that a <T> in the data

list will also terminate the R or W command, but the T command gives your program an additional means of terminating I/O from a data list.

EXAMPLE: Read the first item in the data list at location 1000 and then resume input from the keyboard.

```
S$=CHR$(19)\!S$,"A1000"
!S$,"R"\INPUT T$
!S$,"T"
INPUT "What next, Captain Nemo?",C$
```

The S Command (Define a Scratch area for DOS)

Certain DOS commands (IN,CO,DT and CD) require a scratch area of 2,560 bytes for temporary storage of information from the disk. Normally, the scratch area begins at location 2A00 (hex). Unfortunately, this area overlaps North Star BASIC and therefore, BASIC will be destroyed if you execute one of the four commands listed above. DOS+ gives you the capability of relocating this scratch area by typing or printing <S>S...<CR>. This will set the address of the scratch area to any location (...) that you specify.

In most cases, you can use locations 1600 to 1FFF (hex) as the DOS scratch area. Note that the copy of DOS in memory is altered by the S command, so you should not copy it on another disk without changing the scratch area back to 2A00.

EXAMPLE: Set the DOS scratch area to 1600 (hex)

<S>1600<CR>

<S>S1600<CR>

EXAMPLE: In your BASIC program, initialize a diskette and then change the scratch area back to the normal location.

```
S$=CHR$(19)\!S$,"H"\!S$,"S1600"
INPUT "INSERT A NEW DISK AND TYPE <CR>",A$
!S$,"NIN"\!S$,"S2A00"
```

The E Command (Exit from DOS+)

If you type or print <S>E<CR>, the DOS+ extension will be deactivated and DOS+ will exit to DOS. Since DOS+ alters several locations in DOS, it is generally necessary to deactivate DOS+ before copying DOS onto another diskette. However, Appendix II tells how you can alter DOS so that it will automatically load DOS+ each time you bootstrap the disk. If you have made this modification to DOS, it is not necessary to deactivate DOS+ before copying DOS and DOS+ to another diskette.

## APPLICATION IDEAS

### A. Creating and Executing Macro Commands

Using the W command, it is possible to create a list of commands in memory that will subsequently be executed as a batch job. The first step is to set the address of the command list by outputting <S>A.... You can type the list of commands on your terminal if you precede them with <S>W or they can be printed by your program (PRINT CHR$(19),"W", ... ). At the end of the list of commands, type or print <T> to terminate the list.

Now, you can execute this list of commands by setting the address to the start of the list (<S>A...) and typing or printing <S>R<CR>. You can do this as many times as you wish. When your program asks for input, the data will be read from the command list in memory, rather than from your terminal. As the input is taken from memory, each character is printed on your terminal so that you can see what command is being executed. When the <T> at the end of the command list is encountered, the input from memory will end and subsequent input will be accepted from the terminal, as usual.

Using this approach, any program that uses the DOS routines for character input (i.e., it calls 2010H to get each character) can execute a macro command!

### B. Using DOS+ with Other Programs and Languages

New and exciting programs for 8080-type computers are becoming available at a rapid pace. Unfortunately, most of them do not allow you to use your North Star disk. With DOS+, you

won't become outmoded because it is usually quite easy to make programs compatible with your disk.

Generally, the documentation for Program "X" will tell you how to modify it to work with your I/O devices. If you simply insert your DOS routines for input (CALL 2Ø1ØH) and output (CALL 2ØØDH), you can use DOS+. It's as easy as that. After this adaptation, you can type DOS+ commands whenever the program is waiting for input. You can also modify the program to include print statements that will cause DOS+ to load and save disk files, list the directory, etc.

C. Using DOS+ to Back Up Disks on "Foreign" I/O Devices

Have you ever wanted to back up your disks on an inexpensive tape recorder? With limited memory, this is a painfully slow process of reading one file, jumping to the tape handler, saving that file, and so on.

How about writing a BASIC program to do this for you? First, list the directory (PRINT CHR$(19),"NLI") to determine how many blocks to transfer. Allocate a buffer in memory (say, 2K or 4K) and use the RD command (PRINT CHR$(19),"NRD",D,R,N) to read the appropriate number of blocks into the buffer. Here, D is the disk address, R is the buffer address and N is the number of blocks.

In order to transfer this batch of data to tape, you could set up a DOS+ macro command containing the appropriate commands for your tape handler, print the <S>R command to read from memory and then CALL the tape handler to execute the commands. After returning to your BASIC program, you can increment the disk

address and read in the next batch of data from the disk, etc.

To read the data back from the tape to the disk, you can reverse this process by first executing a macro to read a batch of data from the tape, and then use the WR command to write the data on the disk.

D. Interfacing "Foreign" I/O Devices with BASIC Programs

Using the DOS+ R and W commands, it is an easy matter to make your BASIC programs read and write ASCII data lists at any location in memory. In this form, the data can also be read or written on any I/O device. You should consider using a macro command to pass instructions to the foreign I/O device.

E. Passing Data Between BASIC Programs

When you chain together a series of BASIC programs, intermediate results usually have to be saved on the disk. This is necessary because all variables and arrays are wiped clean as each new program is loaded. In many cases, it would be easier and faster to write the intermediate results into a data list stored outside the memory range of the BASIC programs. The data can then be read conveniently by any other BASIC program.

F. Making Multiple Copies of Disks or Tapes

It often requires quite a few commands to initialize a disk or tape and then write several files on it. With DOS+, you can create a macro command to do this job and then execute it repeatedly to create multiple copies. You can even save the macro command on the disk or tape for future use.

DOS+ UPDATE #1--APPENDIX I

Please replace page 16 of the DOS+ manual with this one. With these minor changes, you will be able to use DOS+ with the new North Star DOS version 4.

Modified Versions of DOS+

The diskette supplied with DOS+ includes four different versions which are compiled for locations 7CØØ(DOS+7C), 6CØØ(DOS+), 5CØØ(DOS+5C) and 1CØØ(DOS+1C). If none of these versions are satisfactory, you should contact Interactive Microware for information about custom versions. If you are using DOS version 3, you can use any of these versions of DOS+ by typing "GO DOS+XX", where XX stands for the last two letters of the name.

If you are using DOS version 2, you will have to alter the following locations (X is the first digit of the start of DOS+):

XDAD=C5 XDCF=C7 XDD2=FD XE4B=21 XE4D=22 XE4E=48 XE4F=23 XE5Ø=22 XE51=5B XE52=23 XE53=22 XE54=B5 XE55=23

For DOS version 4, the following changes are required:

XDCØ=5D XDC6=Ø5 XDC7=21 XDCC=CF XDD2=FA XDD5=AF XDDD=94 XDEØ=CF XDE3=Ø5 XDE4=21 XDED=94 XE4B=8B XE52=EF.

If you have located your Meca TOS anywhere other than 7ØØØH, you should change the following locations in DOS+ (X is the first digit of the start of DOS+ and Y is the first digit of the start of your TOS):

XE95=YF XEA8=YF XEAB=YF XEAE=Y3

Feel free to call me at 814-863-0074 or 814-238-8294 if you have any questions or comments on DOS+.

Paul K. Warme

```
LIST

1!"NORTH STAR DISK/MECA TAPE BACKUP SYSTEM"
2!"COPYRIGHT (C) 1978 BY INTERACTIVE MICROWARE, INC."
3REM  BEFORE RUNNING THIS PROGRAM, INITIALIZE THE TAPE OPERATING
4REM SYSTEM AND GO DOS+, GO BASIC AND LOAD THIS PROGRAM.
5REM  TO TRANSFER FROM DISK TO TAPE, MOUNT THE TAPE ON DRIVE Ø,
6REM AND INITIALIZE IF IT IS A NEW TAPE "^SMNE :Ø".  THEN RUN
7REM THIS PROGRAM WITH THE DISK TO BE SAVED ON DRIVE 1.
8REM  TO RESTORE FROM TAPE TO DISK, MOUNT A FRESH DISK ON DRIVE 1
9REM AND THE BACKUP TAPE ON DRIVE Ø "^SMMO :Ø" AND THEN RUN
10REM THIS PROGRAM.
11REM  TWO FULL DISKS CAN BE STORED ON EACH SIDE OF A CASSETTE.
15S$=CHR$(19)
20N$=S$+"N"\M$=S$+"M"
30INPUT "DISK # ?",N\F$=STR$(N)
35IF N=Ø THEN END
40L=LEN(F$)\IF L<2 THEN F$="Ø"+F$
50F$(1,1)="D"
100INPUT "DISK TO TAPE(1) OR TAPE TO DISK(-1)?",D
110IF D=1 THEN !N$,"LI"
120FOR I=1 TO 12
130B=30\IF I=12 THEN B=20
140BØ=INT(3Ø*(I-1))
150B$=STR$(BØ/1Ø)\L=LEN(B$)
160IF L<3 THEN B$(1,1)="Ø" ELSE B$=B$(2,3)
170IF D=-1 THEN 210
175!N$,"RD",BØ," Ø",B
180IF B=30 THEN 200
190!M$,"SAVE ",F$,B$," :Ø Ø 13FF"
195GOTO 300
200!M$,"SAVEQ ",F$,B$," :Ø Ø 1DFF"
205GOTO 300
210!M$,"LOAD ",F$,B$," :Ø Ø"
220!N$,"WR",BØ," Ø",B
300NEXT I
310IF D=-1 THEN !\!N$,"LI"
320GOTO 20
READY
```

## APPENDIX II

### • Automatic Initialization of DOS+

You can easily modify DOS to load DOS+ each time you bootstrap the disk. You will recall that when you ran DOS for the first time, you defined a subroutine TINIT that is called to initialize your terminal. If you save DOS+ on your disk directly after DOS (i.e., block 14), you can automatically read it in by including the following code in your TINIT routine:

```
TINIT   LXI B,101H      Read from drive 1
        LXI D,XC00H     X=first digit of DOS+ address
        LXI H,14        Disk address
        MVI A,4         Read 4 blocks
        CALL 2022H      DCOM entry
        ...             Other instructions
        JMP XC00H       Initialize DOS+ and return to DOS
```

NOTE: You should generally exit from DOS+ (type<S>E) before saving DOS or DOS+ on a fresh diskette. However, this is unnecessary if you install the above modification and change the last instruction (JMP XC00H) to RET.

DOS+ UPDATE #1--APPENDIX I

Please replace page 16 of the DOS+ manual with this one. With these minor changes, you will be able to use DOS+ with the new North Star DOS version 4.

Modified Versions of DOS+

The diskette supplied with DOS+ includes four different versions which are compiled for locations 7CØØ(DOS+7C), 6CØØ(DOS+), 5CØØ(DOS+5C) and 1CØØ(DOS+1C). If none of these versions are satisfactory, you should contact Interactive Microware for information about custom versions. If you are using DOS version 3, you can use any of these versions of DOS+ by typing "GO DOS+XX", where XX stands for the last two letters of the name.

If you are using DOS version 2, you will have to alter the following locations (X is the first digit of the start of DOS+):

XDAD=C5 XDCF=C7 XDD2=FD XE4B=21 XE4D=22 XE4E=48 XE4F=23 XE5Ø=22 XE51=5B XE52=23 XE53=22 XE54=B5 XE55=23

For DOS version 4, the following changes are required:

XDCØ=5D XDC6=Ø5 XDC7=21 XDCC=CF XDD2=FA XDD5=AF XDDD=94 XDEØ=CF XDE3=Ø5 XDE4=21 XDED=94 XE4B=8B XE52=EF.

If you have located your Meca TOS anywhere other than 7ØØØH, you should change the following locations in DOS+ (X is the first digit of the start of DOS+ and Y is the first digit of the start of your TOS):

XE95=YF XEA8=YF XEAB=YF XEAE=Y3

Feel free to call me at 814-863-0074 or 814-238-8294 if you have any questions or comments on DOS+.

Paul K. Warme

```
LIST

1!"NORTH STAR DISK/MECA TAPE BACKUP SYSTEM"
2!"COPYRIGHT (C) 1978 BY INTERACTIVE MICROWARE, INC."
3REM  BEFORE RUNNING THIS PROGRAM, INITIALIZE THE TAPE OPERATING
4REM SYSTEM AND GO DOS+, GO BASIC AND LOAD THIS PROGRAM.
5REM  TO TRANSFER FROM DISK TO TAPE, MOUNT THE TAPE ON DRIVE Ø,
6REM AND INITIALIZE IF IT IS A NEW TAPE "^SMNE :Ø".  THEN RUN
7REM THIS PROGRAM WITH THE DISK TO BE SAVED ON DRIVE 1.
8REM  TO RESTORE FROM TAPE TO DISK, MOUNT A FRESH DISK ON DRIVE 1
9REM AND THE BACKUP TAPE ON DRIVE Ø "^SMMO :Ø" AND THEN RUN
10REM THIS PROGRAM.
11REM  TWO FULL DISKS CAN BE STORED ON EACH SIDE OF A CASSETTE.
15S$=CHR$(19)
20N$=S$+"N"\M$=S$+"M"
30INPUT "DISK # ?",N\F$=STR$(N)
35IF N=Ø THEN END
40L=LEN(F$)\IF L<2 THEN F$="Ø"+F$
50F$(1,1)="D"
100INPUT "DISK TO TAPE(1) OR TAPE TO DISK(-1)?",D
110IF D=1 THEN !N$,"LI"
120FOR I=1 TO 12
130B=30\IF I=12 THEN B=20
140B0=INT(30*(I-1))
150B$=STR$(B0/10)\L=LEN(B$)
160IF L<3 THEN B$(1,1)="Ø" ELSE B$=B$(2,3)
170IF D=-1 THEN 210
175!N$,"RD",B0," Ø",B
180IF B=30 THEN 200
190!M$,"SAVE ",F$,B$," :Ø Ø 13FF"
195GOTO 300
200!M$,"SAVEQ ",F$,B$," :Ø Ø 1DFF"
205GOTO 300
210!M$,"LOAD ",F$,B$," :Ø Ø"
220!N$,"WR",B0," Ø",B
300NEXT I
310IF D=-1 THEN !\!N$,"LI"
320GOTO 20
READY
```

DOS+ for North Star Disk

TABLE OF CONTENTS

DOS+ for North Star Disk
by
William LaForge and Paul Warme

## INTRODUCTION

DOS+ will significantly expand the capabilities of your North Star floppy Disk Operating System (DOS). It allows any program to control the floppy disk by merely printing two special command letters, followed by the usual DOS command. If you own a Meca tape system, DOS+ gives you the same sort of control over the Tape Operating System (TOS). In fact, DOS+ provides a simple protocol for transferring ASCII data from any mass storage device or program to any other mass storage device or program.

When the DOS+ extension is activated, all input and output characters are "read" by DOS+, but nothing unusual happens until a control S (here denoted <S>) comes along. The letter after the <S> tells DOS+ what you want to do with subsequent input or output to your terminal.

Although each DOS+ command will be explained in detail later, here is a brief rundown of some of them. If the command letter is N, the rest of the line is interpreted as a command for the North Star DOS. If the command letter is M, the rest of the line is interpreted as a command for the Meca TOS.

The command A defines the Address of a data list stored in memory. A data list consists of one or more lines of ASCII characters, ending with a carriage return (here denoted <CR>).

Each line may contain either data or a command to be passed to another program. The command letter W means Write the subsequent output characters into a data list, and the R command causes subsequent input to be Read from a data list. If you want to Print the next line from the data list, the P command will do this. The J command Jumps over lines in the data list and the T command Terminates an R or W command.

When you type <S>, your terminal will print a "$" and DOS+ will "listen" for the rest of the command. All characters that are typed as part of the command are printed by your terminal, but they are ignored by your program. If you make an error while typing a DOS+ command, you can erase one character at a time by typing one or more underlines (left arrow on some terminals). You can erase the entire command line by typing "@". Any undefined command letter after a <S> will be ignored and the bell on your terminal will signal the error.

It is very easy to activate DOS+. Just bootstrap the disk operating system at location E900, mount the DOS+ diskette and type GO DOS+. Appendix I contains special instructions for users of DOS version 2 and for loading DOS+ at addresses other than the standard address (6C00 to 6FFF). DOS+ can be deactivated at any time by typing control S (<S>), the letter "E" for Exit and then a carriage return (<CR>). All DOS+ commands are available to any program that uses your DOS I/O routines for terminal input (CALL 2010H) and output (CALL 200DH).

## DESCRIPTION AND EXAMPLES OF DOS+ COMMANDS

### The N Command (North Star DOS)

If you want to execute a DOS command, you should type or print <S>N, followed by the DOS command and a <CR>. Like all other DOS+ commands, you can either type this command while your program is waiting for input or you can have your program print the command. Since DOS+ "listens" to both input and output data, either form of command is permitted.

You should not use the N command while you are working in DOS because this will probably cause the system to bomb. However, you can use the regular form of DOS command in this case.

If an error occurs during execution af the N command, your program will abort to DOS. The DOS error routine will print "?" and DOS will then wait for a new command after the "*" prompt.

EXAMPLE: While your program is waiting for input, list the directory of the disk on drive 1 (the default drive).

```
<S>NLI<CR>
```

EXAMPLE: In your BASIC program, load file DATA at location 1000. Notice that you don't have to print a <CR> after the command, since the PRINT statement automatically prints <CR> at the end of each line. CHR$(19) is the same as <S>.

```
PRINT CHR$(19),"NLF DATA 1000"
```

The M Command (Meca TOS)

Any Meca tape operating system command can be executed by typing or printing <S>M, the normal TOS command, and then a <CR>. In conjunction with other DOS+ commands, you can transfer files from disk to tape, or vice versa, and you can also write programs in BASIC that can read and write data on tapes (see the R, W and P commands).

If an error occurs while the TOS is in control, the standard TOS error message will be printed. At this point, you should type <CR> to continue; your program will be stopped and TOS will gain control.

EXAMPLE: While your program is waiting for input, list the directory of the tape on drive Ø.

```
<S>MDIR :Ø<CR>
```

EXAMPLE: In your BASIC program, save file DATA from locations 1ØØØ to 2ØØØ on drive 1.

```
PRINT CHR$(19),"MSAVE DATA :1 1ØØØ 2ØØØ"
```

The A Command (Address of data list)

Before using the W, P, R and J commands (see below), you must define the address for a data list. To do this, you should type or print <S>A, followed by the desired address. This address will be updated automatically after each transfer to or from the data list, thus allowing your programs to access each line of the data list in sequence. The address will be interpreted as either

a hexadecimal number or a decimal number, in accordance with the last H or D command issued (see below). If you want to access lines randomly within a data list, you will find the J command very useful.

EXAMPLE: You can define the address of a data list to be location 1000 by typing:

<S>A1000<CR>

or by including the following statement in your BASIC program:

PRINT CHR$(19),"A1000"

The D Command (Decimal number mode)

The numbers typed or printed after the A, J and S commands will be interpreted as decimal numbers if you type <S>D<CR> or insert the statement PRINT CHR$(19),"D" in your BASIC program.

The H Command (Hexadecimal number mode)

This is similar to the D command, but causes subsequent numbers to be interpreted as hexadecimal numbers. The most recent D or H command determines the number base; when DOS+ is initialized, the hexadecimal mode is set.

The W Command (Write output into a data list)

After defining the address of your data list with the A command, you can write information into memory by typing or printing <S>W. Everything that is output after the W is stored

in memory, instead of being printed on your terminal. The W command terminates when a control T (<T>) is output or when you issue the T command (see below). When you type <T> on your terminal, it will be printed as "%". The <T> is stored in memory, but all subsequent output will be printed on your terminal. Bear in mind that the W command will be terminated if any other DOS+ command is issued, as well. The W command is useful for temporary storage of data which will later be read by another program or will be written on a "foreign" mass storage device. It is also capable of creating macro commands (see Application Ideas).

EXAMPLE: Start a data list at location 1000 and write one line.

```
S$=CHR$(19)
PRINT S$,"A1000"
PRINT S$,"WThis is the first line of the data list."
```

EXAMPLE: Write another line in the data list and then return to normal output. Suppose that N denotes the quantity, C the catalog number and P the price of peaches. The data will be formatted for later INPUT using the R command.

```
N$=STR$(N)\N$=N$(2)+","
C$=STR$(C)\C$=C$(2)+","
P$=STR$(P)\P$=P$(2)+","
PRINT N$,C$,P$,"PEACHES"
PRINT CHR$(20),"This line will be printed on the terminal."
```

The P Command (Print the next line from a data list)

If you want to print one line from a data list on your terminal, you should type or print <S>P<CR>. If you want to print the first line, you should use the A command to define the starting address; otherwise, the address is automatically advanced to the beginning of the next line after each line is printed. If the data list is properly terminated with a <T>, the P command will print just a carriage return when it reaches the end of the list.

EXAMPLE: Print the first two lines from a data list at location 1000. You will recall that "!" is a shorthand for PRINT and "\" separates multiple statements on a line. An extra PRINT is included because the P command doesn't print a carriage return.

```
S$=CHR$(19)\PRINT S$,"A1000"
FOR I=1 TO 2\!S$,"P"\!\NEXT I
```

The R Command (Read input from a data list)

After you have defined the address of a data list using the A command, you can read the data into your program by typing or printing <S>R<CR>. Thereafter, all input will be read from the data list, instead of coming from the keyboard of your terminal. As the data is read from memory, each character will be printed by your program, just as if you had typed it on your terminal. Normal input from your keyboard will resume when a <T> is found in the data list or when your program sends the T command (see below) to DOS+. The R command will also be terminated if any

other DOS+ command is issued.

The R command is very useful for reading data stored by other programs or data input from a "foreign" mass storage device. It is also capable of executing macro commands (see Application Ideas). Together, the R and W commands provide a simple and effective mechanism for transferring ASCII data among various programs and I/O devices.

EXAMPLE: Read and print the first line of the data list at location 1000 that was created earlier by the W command.

```
S$=CHR$(19)\!S$,"A1000"
!S$,"R"
INPUT T$
```

EXAMPLE: Read the next line of the data list as four separate items: the number of items (N), catalog number (C), price (P) and description (D$).

```
INPUT N,C,P,D$
```

Notice that subsequent input will be from the terminal, since the next character in the data list was <T>.

The J Command (Jump over lines in a data list)

When you are reading lines from a data list, it is often desireable to be able to access any line (i.e., random access) conveniently. Of course, this could be done by setting the address to the first line and then reading lines until the right line is found. Another way this could be done is to have your program remember the starting address of each line and then use the A command to address the desired line (this is the preferred method when each line is the same length).

The J command provides a quick way to address any line; it is particularly useful when lines are unequal in length. The number typed or printed after the J determines how many lines will be skipped, starting from the present position in the list. If the number after the J is Ø or is omitted, the address is set to the beginning of the previous line. This feature is useful when you want to read a line, alter it and then write it back in the same position in the data list. If you try to jump over more lines than there are in the data list, the address will be set to the location of the <T> at the end of the data list. To avoid an error condition, you should not use the JØ command after reading the first line of the data list. Instead, you should use the A command to return to the start of the first line.

One method for accessing any line in a data list is to set the address to the beginning of the data list and then jump over the required number of lines. If the desired line is beyond the previous line read from the data list, it will be faster to

subtract the previous line number from the desired line number and then jump over that number of lines minus one.

EXAMPLE: Read a line from a data list, alter it and then write it back in the same position.

```
S$=CHR$(19)\!S$,"A1000"\!S$,"J1"
!S$,"R"\INPUT T$\T$(1,4)="1978"
!S$,"J0"\!S$,"W",T$
```

EXAMPLE: Read the 10th line of a data list starting at 1000.

```
S$=CHR$(19)\!S$,"A1000"
!S$,"J9"\!S$,"R"
INPUT T$
```

EXAMPLE: Read the 20th line of the data list after the previous example.

```
N=20-10-1\N$=STR$(N)\N$=N$(2)
!S$,"J",N$\!S$,"R"
INPUT U$\!S$,"T"
```

The T Command (Terminate the R or W command)

If you want to terminate I/O from a data list during a Read or Write command, you should insert (PRINT CHR$(19),"T") in your BASIC program. After this statement is executed, I/O will communicate with your terminal, rather than with the data list.

The T command does not affect the current address in the data list, so you can return to the data list later by printing <S>R<CR> or <S>W<CR>. You will recall that a <T> in the data

list will also terminate the R or W command, but the T command gives your program an additional means of terminating I/O from a data list.

EXAMPLE: Read the first item in the data list at location 1000 and then resume input from the keyboard.

```
S$=CHR$(19)\!S$,"A1000"
!S$,"R"\INPUT T$
!S$,"T"
INPUT "What next, Captain Nemo?",C$
```

The S Command (Define a Scratch area for DOS)

Certain DOS commands (IN,CO,DT and CD) require a scratch area of 2,560 bytes for temporary storage of information from the disk. Normally, the scratch area begins at location 2A00 (hex). Unfortunately, this area overlaps North Star BASIC and therefore, BASIC will be destroyed if you execute one of the four commands listed above. DOS+ gives you the capability of relocating this scratch area by typing or printing <S>S...<CR>. This will set the address of the scratch area to any location (...) that you specify.

In most cases, you can use locations 1600 to 1FFF (hex) as the DOS scratch area. Note that the copy of DOS in memory is altered by the S command, so you should not copy it on another disk without changing the scratch area back to 2A00.

EXAMPLE: Set the DOS scratch area to 1600 (hex)

<S>S1600<CR>

EXAMPLE: In your BASIC program, initialize a diskette and then change the scratch area back to the normal location.

```
S$=CHR$(19)\!S$,"H"\!S$,"S1600"
INPUT "INSERT A NEW DISK AND TYPE <CR>",A$
!S$,"NIN"\!S$,"S2A00"
```

The E Command (Exit from DOS+)

If you type or print <S>E<CR>, the DOS+ extension will be deactivated and DOS+ will exit to DOS. Since DOS+ alters several locations in DOS, it is generally necessary to deactivate DOS+ before copying DOS onto another diskette. However, Appendix II tells how you can alter DOS so that it will automatically load DOS+ each time you bootstrap the disk. If you have made this modification to DOS, it is not necessary to deactivate DOS+ before copying DOS and DOS+ to another diskette.

## APPLICATION IDEAS

### A. Creating and Executing Macro Commands

Using the W command, it is possible to create a list of commands in memory that will subsequently be executed as a batch job. The first step is to set the address of the command list by outputting <S>A.... You can type the list of commands on your terminal if you precede them with <S>W or they can be printed by your program (PRINT CHR$(19),"W", ... ). At the end of the list of commands, type or print <T> to terminate the list.

Now, you can execute this list of commands by setting the address to the start of the list (<S>A...) and typing or printing <S>R<CR>. You can do this as many times as you wish. When your program asks for input, the data will be read from the command list in memory, rather than from your terminal. As the input is taken from memory, each character is printed on your terminal so that you can see what command is being executed. When the <T> at the end of the command list is encountered, the input from memory will end and subsequent input will be accepted from the terminal, as usual.

Using this approach, any program that uses the DOS routines for character input (i.e., it calls 2010H to get each character) can execute a macro command!

### B. Using DOS+ with Other Programs and Languages

New and exciting programs for 8080-type computers are becoming available at a rapid pace. Unfortunately, most of them do not allow you to use your North Star disk. With DOS+, you

won't become outmoded because it is usually quite easy to make programs compatible with your disk.

Generally, the documentation for Program "X" will tell you how to modify it to work with your I/O devices. If you simply insert your DOS routines for input (CALL 2010H) and output (CALL 200DH), you can use DOS+. It's as easy as that. After this adaptation, you can type DOS+ commands whenever the program is waiting for input. You can also modify the program to include print statements that will cause DOS+ to load and save disk files, list the directory, etc.

C. Using DOS+ to Back Up Disks on "Foreign" I/O Devices

Have you ever wanted to back up your disks on an inexpensive tape recorder? With limited memory, this is a painfully slow process of reading one file, jumping to the tape handler, saving that file, and so on.

How about writing a BASIC program to do this for you? First, list the directory (PRINT CHR$(19),"NLI") to determine how many blocks to transfer. Allocate a buffer in memory (say, 2K or 4K) and use the RD command (PRINT CHR$(19),"NRD",D,R,N) to read the appropriate number of blocks into the buffer. Here, D is the disk address, R is the buffer address and N is the number of blocks.

In order to transfer this batch of data to tape, you could set up a DOS+ macro command containing the appropriate commands for your tape handler, print the <S>R command to read from memory and then CALL the tape handler to execute the commands. After returning to your BASIC program, you can increment the disk

address and read in the next batch of data from the disk, etc.

To read the data back from the tape to the disk, you can reverse this process by first executing a macro to read a batch of data from the tape, and then use the WR command to write the data on the disk.

D. Interfacing "Foreign" I/O Devices with BASIC Programs

Using the DOS+ R and W commands, it is an easy matter to make your BASIC programs read and write ASCII data lists at any location in memory. In this form, the data can also be read or written on any I/O device. You should consider using a macro command to pass instructions to the foreign I/O device.

E. Passing Data Between BASIC Programs

When you chain together a series of BASIC programs, intermediate results usually have to be saved on the disk. This is necessary because all variables and arrays are wiped clean as each new program is loaded. In many cases, it would be easier and faster to write the intermediate results into a data list stored outside the memory range of the BASIC programs. The data can then be read conveniently by any other BASIC program.

F. Making Multiple Copies of Disks or Tapes

It often requires quite a few commands to initialize a disk or tape and then write several files on it. With DOS+, you can create a macro command to do this job and then execute it repeatedly to create multiple copies. You can even save the macro command on the disk or tape for future use.

DOS+ UPDATE #1--APPENDIX I

Please replace page 16 of the DOS+ manual with this one. With these minor changes, you will be able to use DOS+ with the new North Star DOS version 4.

## Modified Versions of DOS+

The diskette supplied with DOS+ includes four different versions which are compiled for locations 7CØØ(DOS+7C), 6CØØ(DOS+), 5CØØ(DOS+5C) and 1CØØ(DOS+1C). If none of these versions are satisfactory, you should contact Interactive Microware for information about custom versions. If you are using DOS version 3, you can use any of these versions of DOS+ by typing "GO DOS+XX", where XX stands for the last two letters of the name.

If you are using DOS version 2, you will have to alter the following locations (X is the first digit of the start of DOS+):

XDAD=C5 XDCF=C7 XDD2=FD XE4B=21 XE4D=22 XE4E=48 XE4F=23 XE5Ø=22 XE51=5B XE52=23 XE53=22 XE54=B5 XE55=23

For DOS version 4, the following changes are required:

XDCØ=5D XDC6=Ø5 XDC7=21 XDCC=CF XDD2=FA XDD5=AF XDDD=94 XDEØ=CF XDE3=Ø5 XDE4=21 XDED=94 XE4B=8B XE52=EF.

If you have located your Meca TOS anywhere other than 7ØØØH, you should change the following locations in DOS+ (X is the first digit of the start of DOS+ and Y is the first digit of the start of your TOS):

XE95=YF XEA8=YF XEAB=YF XEAE=Y3

Feel free to call me at 814-863-ØØ74 or 814-238-8294 if you have any questions or comments on DOS+.

Paul K. Warme

```
LIST

1!"NORTH STAR DISK/MECA TAPE BACKUP SYSTEM"
2!"COPYRIGHT (C) 1978 BY INTERACTIVE MICROWARE, INC."
3REM  BEFORE RUNNING THIS PROGRAM, INITIALIZE THE TAPE OPERATING
4REM SYSTEM AND GO DOS+, GO BASIC AND LOAD THIS PROGRAM.
5REM  TO TRANSFER FROM DISK TO TAPE, MOUNT THE TAPE ON DRIVE Ø,
6REM AND INITIALIZE IF IT IS A NEW TAPE "^SMNE :Ø".  THEN RUN
7REM THIS PROGRAM WITH THE DISK TO BE SAVED ON DRIVE 1.
8REM  TO RESTORE FROM TAPE TO DISK, MOUNT A FRESH DISK ON DRIVE 1
9REM AND THE BACKUP TAPE ON DRIVE Ø "^SMMO :Ø" AND THEN RUN
10REM THIS PROGRAM.
11REM  TWO FULL DISKS CAN BE STORED ON EACH SIDE OF A CASSETTE.
15S$=CHR$(19)
20N$=S$+"N"\M$=S$+"M"
30INPUT "DISK # ?",N\F$=STR$(N)
35IF N=Ø THEN END
40L=LEN(F$)\IF L<2 THEN F$="Ø"+F$
50F$(1,1)="D"
100INPUT "DISK TO TAPE(1) OR TAPE TO DISK(-1)?",D
110IF D=1 THEN !N$,"LI"
120FOR I=1 TO 12
130B=3Ø\IF I=12 THEN B=2Ø
140BØ=INT(3Ø*(I-1))
150B$=STR$(BØ/1Ø)\L=LEN(B$)
160IF L<3 THEN B$(1,1)="Ø" ELSE B$=B$(2,3)
170IF D=-1 THEN 21Ø
175!N$,"RD",BØ," Ø",B
180IF B=3Ø THEN 2ØØ
190!M$,"SAVE ",F$,B$," :Ø Ø 13FF"
195GOTO 3ØØ
2ØØ!M$,"SAVEQ ",F$,B$," :Ø Ø 1DFF"
205GOTO 3ØØ
21Ø!M$,"LOAD ",F$,B$," :Ø Ø"
22Ø!N$,"WR",BØ," Ø",B
3ØØNEXT I
31ØIF D=-1 THEN !\!N$,"LI"
32ØGOTO 2Ø
READY
```

## APPENDIX II

### Automatic Initialization of DOS+

You can easily modify DOS to load DOS+ each time you bootstrap the disk. You will recall that when you ran DOS for the first time, you defined a subroutine TINIT that is called to initialize your terminal. If you save DOS+ on your disk directly after DOS (i.e., block 14), you can automatically read it in by including the following code in your TINIT routine:

```
TINIT   LXI B,101H      Read from drive 1
        LXI D,XC00H     X=first digit of DOS+ address
        LXI H,14        Disk address
        MVI A,4         Read 4 blocks
        CALL 2022H      DCOM entry
        ...             Other instructions
        JMP XC00H       Initialize DOS+ and return to DOS
```

NOTE: You should generally exit from DOS+ (type<S>E) before saving DOS or DOS+ on a fresh diskette. However, this is unnecessary if you install the above modification and change the last instruction (JMP XC00H) to RET.