TALL TREE SYSTEMS

JETDRIVE/JSPOOL COMBO DISK

1120 San Antonio Road, Palo Alto, CA 94303
Sales and Customer Service (415)964-1980
Technical Support (415)964-1397

Tall Tree Systems produces hardware peripherals and accompanying software for use on the IBM PC, XT, AT and other compatible machines.

HARDWARE: We pioneered bank-switching megamemory boards and now feature several JRAM multifunction boards. Our boards can be populated with either 64K or 256K chips. Fully populated with 256K chips, the JRAM provides 2 Mb of memory in just one slot. All boards are guaranteed for one year.

JRAM-3 is a multifunction memory board for IBM PCs and XTs, designed to work both with conventional software and Expanded Memory Specification software. It comes with software for a RAM disk and printspooler, as well as a driver to control EMS software and the JLaser interface. Any of our piggyback modules can fit on this board, including our JLaser interface for Canon based laser printers.

JRAM-2 is our earlier version bank-switching board for PCs, XTs and compatibles. It comes with JETDRIVE/JSPOOL software for a RAM disk and printspooler and can be mounted with a piggyback module for additional ports, clock/calendar, or a diskette controller. This board does not support the LOTUS/INTEL MICROSOFT Expanded Memory specification.

JRAM-AT is our 16 bit board for ATs and compatibles. It can be used either as a standard memory expansion board or a bank switching board with our JETDRIVE/JSPOOL software.

JRAM-AT3 can do everything the JRAM-AT can do PLUS work with EMS software, JPager, and it has even more flexible bank-switching capabilities.

JLASER is a hardware interface that revolutionizes the speed, quality, and flexibility of laser printers, giving you full page graphics and custom fonts at eight pages per minute. Add it to either our JRAM-3 or JRAM-AT3 board, with recommended minimum of 1 Mb of RAM for full page graphics.

The JDISKETTE controller, available either as a module (for JRAM-2 and JRAM-3 boards) or a stand-alone board, allows you to put 1.2 megabyte diskette drives in the IBM PC or XT (the JDISKETTE is not recommended for compatibles) for the ultimate in diskette storage and IBM AT compatibility.

SOFTWARE: The source code for each of these installable device drivers is provided.

JETDRIVE provides up to four RAM disks plus JET -- the industry's fastest file transfer program for DOS 2.0 and above.

JFORMAT-2 lets you use quad (96 tpi), 8", or 1.2Mb drives in a PC, XT, or compatible. 8" and 1.2Mb drives also require a special diskette controller. Put 800 K, 1.2 Mb, or more on one diskette.

JSPOOL, a redirectable printspooler, works with serial and parallel ports.

This manual was written by Janet Bein and Bruce Owen.

JETDRIVE/JSPOOL
TABLE OF CONTENTS

These sections are designed to be included in the JRAM User's Manual, which accompanies our memory boards.

PAGE

JETDRIVE

JETDRIVE ERRATA

August 8, 1985

33.5: JET:TRANSFERRING LISTS OF FILES (page 33-4) Files in file lists must be typed in 11-character format (8 characters for the file name and 3 characters for the extension, with no dot or space between them). File names without extensions and file names with extensions of less than three characters must be padded out to 11 characters in length with spaces. The example below is clearer than the one in your manual.

```
A>COPY CON WSFILES
WS      COM
WSMSGS  OVR
WSOVLY1 OVR
JET     COM
<F6> (Screen will show ^Z)
```

33.6: JET: HANDLING SUBDIRECTORIES (page 33-6) JET syntax requires a \ before both subdirectory names and file names in subdirectories. All the \ marks were left out of your manual.

line 4 of 33.6: The file name...must be separated from the last subdirectory name by a \ not a space.

line 7 of 33.6: Do not specify the root directory with "\" (empty backslash).

Page 33-7, line 5: `A>JET A:AFILE B:\SUBDIR/256`

SUMMARY - HANDLING SUBDIRECTORIES (page 33-7)

Syntax: JET [<source>]\<subdir>..[\<filename>] <target>:\<subdir>..[/<option>..]

Examples:

```
JET A:\SUB1\MYFILE B:\BSUB
JET A:\SUB1\SUB2\SUB3 B:
JET \ASUB B:\BSUB1\BSUB2/V
```

Do not specify the root directory with an empty backslash (like C:\).

SUMMARY - BACKING UP YOUR HARD DISK (page 33-9)

Syntax for backing up:
JET <source>:[\<subdir>...] <target>:[\<subdir>...]</D or /S>

Example:

```
JET D:\SUB1 A:/D
```

Syntax for restoring:
JET <source>:[\<subdir>...] <target>:[\<subdir>...]</R or /A>

Example: `JET A: D:\SUB1/R`

34.2: THE JET OPTIONS (page 34-1)

line 3 of 34.2: `A>JET A: B:\SUBDIR/256`

If you want to use MAKEDOS to change the size of DOS without changing CONFIG.SYS, use this line in CONFIG.SYS:

```
DEVICE=JBOOT.BIN 0/O=11b
```

(use "zero" before the slash and the letter "O" after the slash)

PROGRAM LICENSE

You should read carefully the following terms and conditions before opening this diskette package. Opening this diskette package indicates your acceptance of these terms and conditions. If you do not agree with them, you should promptly return the package unopened; your money will be refunded.

You may:

A. Use the JETDRIVE and JSPOOL programs on a single machine.

B. Use and copy the JBOOT.BIN, JDRIVE.BIN, MAKEDOS.COM, JET.COM, DUMDISK.BIN, JBOOT.ASM, JDRIVE.ASM, JPAGER.BIN, JPAGER.ASM, JLASER.BIN, JLASER.ASM, JSPOOL.BIN, SPOOLCOM.COM, and MAKEFILE.COM programs as many times on as many diskettes as desired for use on a single machine. Do not use JPAGER.BIN or JLASER.BIN unless you have a JRAM-3 or JRAM-AT3 memory board.

You may not use, copy, modify, or transfer the program, or any copied, modified, or merged portion, in whole or in part, except as expressly provided for in this license.

If you transfer possession of any copy, modification, or merged portion of the programs to another party, your license is automatically terminated. You agree, upon such termination, to destroy the program together with all copies, modifications, and merged portions in any form.

PROGRAM LIMITED WARRANTY

These programs are provided "as is" without warranty of any kind, either expressed or implied, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the programs is with you. Should the programs be defective, you (not Tall Tree Systems) assume the entire cost of all necessary servicing, repair, or correction. In no event will Tall Tree Systems be liable to you for any damages, including any lost profits, lost savings, or other incidental or consequential damages. Some states do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

Tall Tree Systems does not warrant that the functions contained in these programs will meet your requirements or that the operation of the programs will be uninterrupted or error free. However, Tall Tree Systems warrants the diskettes on which the program is furnished to be free from defects in materials and workmanship, and will replace defective diskettes free of charge.

JETDRIVE
SECTION THIRTY
INTRODUCTION TO JETDRIVE

30.1: WHAT IS JETDRIVE?

JETDRIVE is a software package that creates one to four RAM disks and helps you to transfer files quickly and conveniently. A RAM disk is a part of memory set up to simulate a diskette drive. A RAM disk acts exactly like a diskette drive, with four important exceptions:

1. Speed. Physical diskette and hard disk drives are mechanical devices, so they are extremely slow by computer standards. A RAM disk, on the other hand, operates at electronic speed. Loaded onto a RAM disk, programs like text editors and data bases that access a drive frequently run two to fifty times faster than they do from a diskette drive, and up to five times faster than from a hard disk.

2. Capacity. If you use JETDRIVE with Tall Tree System's JRAM boards, the RAM disks can have up to sixteen megabytes of storage capacity.

3. Safety. You will do all your computer work from the RAM disks, working on files copied from originals stored on physical diskettes. If an accident should happen, you still have the original files on the physical diskette. When you work from a RAM disk, you can never lose or damage more than one session's work.

4. Volatility. RAM disks are for temporary storage only. When you turn on the computer, you must transfer the programs and data files you want to use to the RAM disks. When you finish, you must copy your completed work onto a physical diskette or hard disk for safekeeping. If the power fails, you lose the contents of the RAM disks. Use the JET utility to transfer files quickly and conveniently.

30.2: COMPATIBILITY

JETDRIVE is designed for the IBM PC, XT, and will also work with many compatible machines that are running PC DOS or MS DOS. The machine must have a double-sided diskette drive in order to read the distribution diskette.

A system using JETDRIVE must have at least 192K of RAM, and we suggest at least 256K for a useful system. JETDRIVE works well with ordinary memory boards, including those by AST and Quadram. The real power of JETDRIVE, however, is that it can manage the enormous add-on memory capacity of Tall Tree Systems' JRAM boards. Unless you plan to write your own bank-switching software, JETDRIVE is required to use these boards.

30.3: THE RAM DISKS: FEATURES AND OPTIONS

Unlike other popular RAM disk programs, the JETDRIVE programs are "installable device drivers". Installable device drivers are special programs allowed for by the PC Disk Operating System (PC-DOS). These programs actually become part of DOS, without altering the way DOS functions. Therefore, the JETDRIVE programs are completely compatible with applications and other software, and do not affect any of the useful features of DOS.

JETDRIVE allows non-destructive rebooting. You can reboot by pressing <Ctrl><Alt><Del> without losing the contents of the RAM disks. No other RAM disk can survive a warm boot.

With JETDRIVE, you can make up to four RAM disks of any size you want, leaving from 128K to 640K of memory for use by DOS and your applications programs. If you have a JRAM board, you can even allocate an extra 64K to DOS, for a total of 704K. You can specify or change the sizes of the RAM disks from your keyboard, without setting any internal switches.

30.4:JET: THE HIGH-SPEED FILE TRANSFER UTILITY

JET helps you transfer files to and from your RAM disks at the beginning and end of every computer session. JET is also useful for many other kinds of copying, comparing, updating, and backing-up functions for diskette and hard disk drives.

30.5: WHAT YOU GET

Your Tall Tree Systems distribution diskette contains a number of files, including programs for JSPOOL as well as JETDRIVE. The essential JETDRIVE files are:

JBOOT.BIN

This file is an installable device driver that takes control when the system boots up in order to initialize JRAM boards, to set up the DOS memory allocation, and to redefine certain commands such as <Ctrl><Alt><Del>.

JDRIVE.BIN

This file is also an installable device driver. It creates and manages the RAM disks and the JRAM boards.

JET.COM

This is the high speed file transfer utility.

MAKEDOS.COM

This program allows you to change the amount of memory allocated to DOS directly from the keyboard.

DUMDISK.BIN

This program allows you to skip drive letters before or between the RAM disks.

JBOOT.ASM

This file is the source code for JBOOT.BIN. If you are a programmer and want to see the source code, TYPE this file.

JDRIVE.ASM

This file is the source code for JDRIVE.BIN. If you are a programmer and want to see the source code, TYPE this file.

JETDRIVE
SECTION THIRTY
INTRODUCTION TO JETDRIVE

30.1: WHAT IS JETDRIVE?

JETDRIVE is a software package that creates one to four RAM disks and helps you to transfer files quickly and conveniently. A RAM disk is a part of memory set up to simulate a diskette drive. A RAM disk acts exactly like a diskette drive, with four important exceptions:

1. Speed. Physical diskette and hard disk drives are mechanical devices, so they are extremely slow by computer standards. A RAM disk, on the other hand, operates at electronic speed. Loaded onto a RAM disk, programs like text editors and data bases that access a drive frequently run two to fifty times faster than they do from a diskette drive, and up to five times faster than from a hard disk.

2. Capacity. If you use JETDRIVE with Tall Tree System's JRAM boards, the RAM disks can have up to sixteen megabytes of storage capacity.

3. Safety. You will do all your computer work from the RAM disks, working on files copied from originals stored on physical diskettes. If an accident should happen, you still have the original files on the physical diskette. When you work from a RAM disk, you can never lose or damage more than one session's work.

4. Volatility. RAM disks are for temporary storage only. When you turn on the computer, you must transfer the programs and data files you want to use to the RAM disks. When you finish, you must copy your completed work onto a physical diskette or hard disk for safekeeping. If the power fails, you lose the contents of the RAM disks. Use the JET utility to transfer files quickly and conveniently.

can specify up to 704K for DOS.

To create a CONFIG.SYS file that specifies 192K for DOS, and establishes a single RAM disk, type:

```
A>COPY CON CONFIG.SYS
DEVICE=JBOOT.BIN 192K
DEVICE=JDRIVE.BIN
<F6> (Screen will show ^Z)
```

You will see the message: 1 file(s) copied

The "COPY CON CONFIG.SYS" line tells the computer to create a file called CONFIG.SYS that will contain whatever is typed in from the keyboard (CONsole) until the <F6> key is pressed.

If you want to specify a different amount of memory for DOS, just type a different number after DEVICE=JBOOT.BIN. This number is the number of kilobytes you want to allocate to DOS. It must be separated from JBOOT.BIN by a space, and must be followed by an upper or lower case K. This number should be a multiple of 64K. If you do not specify a multiple of 64K, DOS will be made to the size you specify, but the first RAM disk will start at the next higher multiple of 64K. The memory in between will be wasted. If you expect to be using several different sizes of DOS, we suggest making a separate boot diskette with an appropriate CONFIG.SYS file for each application.

To create more than one RAM disk, use a separate JDRIVE.BIN line for each one, specifying the size of the RAM disk in Kilobytes on each line. We recommend not specifying a size for the last RAM disk; it will use up all the memory that is left over after DOS and the previous RAM disks have been installed. For example, to allocate 192K to DOS and create two RAM disks, one of 128K and another that will use up the rest of the memory in your machine, use a CONFIG.SYS file like:

```
A>COPY CON CONFIG.SYS
DEVICE=JBOOT.BIN 192K
DEVICE=JDRIVE.BIN 128K
DEVICE=JDRIVE.BIN
<F6>
```

The RAM disks may be of any size, in multiples of 64K. If you specify a size that is not a multiple of 64K, it will be rounded up.

5. Reboot the computer by pressing <Ctrl><Alt><Del>. This installs everything in CONFIG.SYS and displays a message like:

```
JBOOT  Ver 2.51 Ser.#123456 Copyright(c)1984,
Tall Tree Systems 192K
128 Kilobytes  now in DOS.
192 Kilobytes  wanted in DOS.
Sizing memory, RAM =
```

There will be a pause as the computer measures how much memory is really installed in the computer (as opposed to what is set on the motherboard memory switches). Then the message will continue:

```
Sizing memory, RAM = 128 Kilobytes
JRAM = 512 Kilobytes
192 Kilobytes  now in DOS
JDRIVE RAM Disk  Ver 2.51  128K
Boot+Fat+Dir = 11K, Data = 117K, * Init *
JDRIVE RAM Disk  Ver 2.51
Boot+Fat+Dir = 11K, Data = 309K, * Init *
```

Of course, if your memory switches are set differently or you do not have a JRAM board, the numbers in the message will be different. The RAM disks can hold the amount of data shown in "Data = ", while the "Boot+Fat+Dir =" figure is the amount of space used to keep track of the files stored on the RAM disk. You will see these messages every time you boot up. You will also see a similar message when you reboot by pressing <Ctrl><Alt><Del>.

6. Make a backup copy of your new boot diskette.

7. Determine what drive letters refer to your RAM disks. DOS assigns drive letters to the RAM disks, usually starting with C: or D:, depending on your particular system. To find out which drive letters are assigned to your RAM disks, put a diskette with at least one file on it in every drive you have, and make sure that at least one file is stored on your hard disk, if you have one. Then type: **A>DIR C:**

If you see the "File not found" message, then you have found a drive with no files stored on it: one of your RAM disks. If you do not see this message, try DIR for drive letter D:, then E:, F:, and so on until you see the "File not found" message. You have not used up all the assigned drive letters until you see the message: **Invalid drive specification.** You will notice that some of your physical diskette and hard disk drives may respond to more than one drive letter. This is just a quirk of DOS.

Note: DOS provides a command (ASSIGN) that lets you rename your drives. Do not use it. ASSIGN causes unpredictable side effects.

To see how large each RAM disk is, type:

A>CHKDSK <the drive letter of one of your RAM disks>:

The first line of CHKDSK report tells you how large the RAM disk is. If any of the RAM disk is filled up with files, they will be listed in several more lines, ending with the line showing how much space is still available on the disk.

SECTION THIRTY-TWO USING THE RAM DISKS

32.1: INTRODUCTION TO USING THE RAM DISKS

To take advantage of the speed and capacity of your RAM disks, you should transfer your programs and data to them, log onto a RAM disk, and run your programs from there. When you are done with your work, transfer your new and changed files back to a physical diskette for permanent storage. The JET high speed file transfer utility helps you do these things quickly and conveniently. (If you prefer,you can use COPY.)

Just to get started, you can transfer the entire contents of the diskette in drive A: to a RAM disk. If one of your RAM disks is drive C:, you might put a diskette that has your text editor and your text files in drive A: and type: **A>JET A: C:** and you would see a message like:

```
153600 bytes copied JET 3.20
(c)1982,83,84,85 Tall Tree Systems
```

If the RAM disk is very small, and you have a lot of data on the diskette in drive A:, you will see a message like:

```
Output Volume too small
0 bytes copied JET 3.20 (c)1982,83,84,85 Tall Tree Systems
```

No harm has been done. There just was not enough space on the RAM disk. For now, just transfer the files you will need individually or with wild cards, as you would with COPY:

A>JET A:MYFILE.TXT C: **or** A>JET A:*.TXT C:

Once you have transferred the files you will need to a RAM disk, you can log onto the RAM disk and get to work. If the RAM disk is drive C:, type: A>C: The prompt will change to C>. You can now use the computer exactly as you used to. Just remember: when you are done with your files, transfer your work back to a physical diskette for permanent storage. Anything that is "saved" on a RAM disk will be lost if you turn the computer off. It is a good idea to transfer your files from the RAM disk to a physical diskette frequently during your work session. This way, if the power fails or your computer is accidentally turned off, you will not lose too much work.

A good way use the RAM disks is with an AUTOEXEC.BAT file. An AUTOEXEC.BAT file causes the computer to AUTOmatically EXECute all the commands it finds in the AUTOEXEC.BAT file every time it boots up or reboots. (For more information about AUTOEXEC.BAT files, see the DOS manual.) If one of your RAM disks is drive C:, to create a simple AUTOEXEC.BAT file, you type:

```
A>COPY CON AUTOEXEC.BAT
JET A: C:
C:
<F6>
```

This AUTOEXEC.BAT file will automatically transfer all the files in drive A: to the RAM disk, and then log the computer onto the RAM disk. Once you know more about JET, you will want to make a more selective AUTOEXEC.BAT file. If you want DOS to read the Command.com file from the RAM drive, add two lines to the end of your Autoexec.bat file after you have copied Command.com to drive C:

```
SET COMSPEC = C: \ COMMAND.COM
COMMAND /C COMMAND
```

Note: When you reboot by pressing <Ctrl><Alt><Del> and there is a non-system diskette in drive A:, two things may happen. If the diskette was made with DOS FORMAT, you will see a message asking you to insert a bootable diskette and press any key. The RAM disks will be restored, provided you insert a bootable diskette with the JETDRIVE programs and CONFIG.SYS on it. However, if the non-system diskette in drive A: was made with JFORMAT-2, the contents of the RAM disk will be lost. So, if you use JFORMATted diskettes in drive A:, make sure they are all bootable system diskettes with JETDRIVE and CONFIG.SYS on them.

SECTION THIRTY-THREE
USING JET

33.1: INTRODUCTION TO JET

JET was created to quickly and conveniently transfer many large programs and data files to and from the RAM disks. However, it can also be used to transfer files from any diskette or hard disk drive to any other drive, regardless of the drives' sizes and formats, and it can be used to compare and update diskettes in a variety of ways. One way to understand JET is to compare it to the roughly analogous COPY and DISKCOPY commands in DOS.

COPY is primarily intended for transferring single files, although it can be made to transfer multiple files with similar names by using wild cards. JET is primarily intended to copy multiple files or whole diskettes, although it can also transfer single files and multiple files with similar names by using wild cards. COPY can not transfer operating system files (the hidden files that allow a diskette to be used for booting up). JET can. COPY lists each file as it is transferred. JET gives you the option of listing the files it is about to transfer, and can even pause to let you change your mind and cancel the transfer. COPY can rename the files it transfers. JET cannot. COPY tries to transfer files into contiguous regions of the target diskette. JET does too. Contiguous files can be accessed more rapidly than files that are fragmented and scattered on the diskette; in applications that step sequentially through files, such as text editors, this can noticeably speed up performance. Both COPY and JET check for bad tracks on the target diskette, and avoid placing data on them. Finally, JET is much faster than COPY.

DISKCOPY is for duplicating entire diskettes. Unlike

JET, DISKCOPY does not check for bad tracks on the target diskette, so it can occasionally lose data. Unlike JET, DISKCOPY cannot transfer files to a diskette of a different format from the source diskette. This means that DISKCOPY can only make a single-sided diskette from a single-sided original, and that it cannot transfer files to special formats such as quad density or hard disks. And, finally, DISKCOPY cannot selectively transfer particular files. However, DISKCOPY can format a blank diskette, while COPY and JET require the target diskette to be previously formatted. DISKCOPY is faster than COPY, but still not as fast as JET.

JET looks at the time and date of each file, and can transfer just those files whose times and dates meet your criteria: those that are more recent on the source diskette, those more recent than a certain date, and so on.

33.2: USING JET

You can JET files from any drive to any other drive. If you type: **A>JET B: C:** the files on the diskette in drive B: will be transferred to the diskette, RAM disk, or hard disk in drive C:.

Every use of JET allows the specification of various options at the end of the command line. These options are signified by a slash followed by a letter. For example, the command: **JET A: B:/P** specifies that JET is to list all the files to be transferred, and pause for you to indicate whether you want those files to be transferred or whether you want to cancel the command. Each of these options is fully explained in section 34.2.

Here are a few details about how JET operates:

* JET does not transfer a file if it finds a file with the same name, date, and time on the target diskette unless the /M option is specified. Files of zero length, such as subdirectory names and volume labels, are not transferred.

* Files already on the target diskette are not erased unless the /E, /ERA, /X, or /XCL options are specified.

* Hidden files are transferred, except for the DOS programs and BADTRACK. JET can transfer the DOS files and boot sector if the /O and /B options are specified.

* Since COMMAND.COM is almost never needed on a RAM disk, JET does not transfer it unless /O or /C is specified.

33.3: JET: TRANSFERRING EVERYTHING ON THE DISKETTE

The simplest way to use JET is to transfer the entire contents of a diskette to another diskette, hard disk, or RAM disk. To copy the entire contents of drive A: to drive B:, you type: **A>JET A: B:**

Drive A: will whirr, and you will see a message like:

```
153600 bytes copied JET 2.70
(c)1982,83,84 Tall Tree Systems
```

SUMMARY--
TRANSFERRING EVERYTHING ON THE DISKETTE

Syntax:

```
JET [<source>:] <target>:[/<option>...]
```

Examples:

```
JET A: C:
JET B:
JET B:/P
```

Notes: If only one drive letter is specified, the source drive is assumed to be the currently logged drive and the target drive is the drive specified. JET does not normally transfer COMMAND.COM because it is almost never needed on a RAM disk. A few programs, such as VOLKSWRITER, require force JET to transfer COMMAND.COM by specifying the /C option.

33.4: JET: TRANSFERRING SPECIFIC FILES

You can use JET with specific file names and wild cards to transfer files, much as you would use COPY. If you use WORDSTAR, you might type any or all of these commands:

```
A>JET A:TEXTFILE.DOC B:
A>JET WS*.* C:
A>JET B:????.DOC C:
```

These commands would cause TEXTFILE.DOC, all of your WORDSTAR programs, and all your working files that have names four characters long and the extension ".DOC" to be quickly transferred to the specified target diskette.

SUMMARY -- TRANSFERRING SPECIFIC FILES

Syntax:
JET <source>:<file name> <target>:[/<option>...]

Examples:
JET A:MYFILE.TXT B:
JET A:*.TXT B:/L

Notes: Unlike COPY, JET cannot rename files. No file name on the target diskette is allowed. If you do not specify a source diskette, JET will default to the currently logged drive.

33.5: JET: TRANSFERRING LISTS OF FILES

If you will be using a particular set of programs or data files frequently, you may want to use JET to transfer a list of files. If you use WORDSTAR, you might make a file list like this on drive A:

```
A>COPY CON WSFILES
WS      COM
WSMSGS  OVR
WSOVLY1 OVR
JET     COM
WORKFILEDOC
<F6>
```

Then, whenever you wanted to use those files with a RAM disk, you could type: **JET A: C: A:WSFILES/F**

All the files listed in the file WSFILES would be transferred to the RAM disk. You can have several file lists for different applications, so you can load your RAM disks with spreadsheet programs and text files when you need them, or with database programs and data files when you need those.

SUMMARY -- TRANSFERRING FILE LISTS

File list syntax: COPY CON <file list> <file name in 11-character format> <more file names in 11-character format> <F6>

Example: COPY CON MYLIST.LST JOHNLET DOC EDITOR COM L7 508 WORKFILEDOC <F6>

JET syntax: JET [<source>:] <target>: <source>:<file list>/ <F, T, X, or XCL> [/<option>...]

Example: JET A: C: A:MYLIST.LST/F

Notes: The files in a file list must be listed in 11-character format. This means that spaces are added between the file name and the extension to make the file name a full 8 characters long. Spaces are added to the file extension, if any, to make it a full 3 characters long. No dot is used, and no wild cards are allowed. The file names must be separated by two characters. The best way to do this is to press the carriage return key (which sends the two characters carriage return and line feed) after each file name. This will make them appear on separate lines as shown above.

The file list must be terminated by pressing <F6>. A single file list can contain up to 39 file names.

The name of the file list must be fully specified in the JET command without wild cards. Only one file list can be transferred at a time. One of the four options /F, /T, /X, or /XCL must be specified in the JET command. /F simply transfers the files in the file list. /T transfers everything on the source **except** the files in the file list. /X transfers the files in the file list and deletes **everything** on the target diskette that is not in the file list, including subdirectories and hidden files, with the single exception of the volume label, after displaying a message and pausing to let you cancel the command. /XCL is the same as /X, but does not display a message or pause. /XCL is useful for batch files, but be careful with it. /XCL makes it very easy to wipe out a lot of data. /T may be used together with /X or /XCL.

33.6: JET: HANDLING SUBDIRECTORIES

To transfer files to and from subdirectories, you must specify the "path", much as you do with COPY. The file name, if any, must be separated from the last subdirectory name by a , not a space. You never need CHDIR, CD, MKDIR, or MD in a JET command. If no subdirectory is specified, JET defaults to the root directory. Do not specify the root directory with " " (empty backslash).

If the target subdirectory does not exist, JET will create a subdirectory of the specified name, with space for two fewer file names than there is in the new subdirectory's parent directory. The root directory on double sided PC diskettes can hold 112 files and subdirectory names, the root directory on single-sided PC diskettes can hold 61 files and subdirectories, and the root directory on a PC XT's hard disk holds 512 files and subdirectories. The RAM disk's root directories hold 256 files and subdirectories unless you have specified otherwise in CONFIG.SYS (see section 35.4 for how to specify the size of the RAM disk directory). If you use JET to create a subdirectory on a RAM disk, for example, the subdirectory will hold 254 files and subdirectories. If you use the DOS command MKDIR or MD to create a subdirectory, it will have space for only 32 files.

Once a subdirectory is created, JET will not enlarge it unless you tell it to with the /# option (see section 34.2 for details). COPY enlarges subdirectories automatically. When JET finds that it is about to transfer more files than there is space for, it displays this error message:

```
Output Volume too small' 0 bytes copied JET 2.70 (c)1982,83,84 Tall Tree Systems'
```

and does not transfer any files at all. If you JET files to a subdirectory created by MKDIR, you will only be able to transfer files until the total number of files in the subdirectory is 32. We recommend routinely creating

subdirectories with JET, so that you will not have to enlarge them often.

If you find that a subdirectory is not big enough, use the /# option to expand it. Type:

```
A>JET A:AFILE B: SUBDIR/256
```

The subdirectory SUBDIR will be expanded to hold a total of 256 entries. Subdirectories may be as large as 512 entries.

JET does not allow subdirectory names that include a period (.). Use subdirectory names of eight characters or less, without extensions separated from the name by a period.

For your protection, the /E and /ERA options do not erase files in subdirectories.

SUMMARY -- HANDLING SUBDIRECTORIES

Syntax:

```
JET [<source>:] <subdir>...[ <file name>]
    <target>:[ <subdir>...] [/<option>...]
```

Examples:

```
JET A: SUB1 MYFILE B: BSUB
JET A: SUB1 SUB2 SUB3 B:
JET  ASUB B: BSUB1 BSUB2/V
```

Notes: Do not specify the root directory with " " (empty backslash). If you do not specify a source drive, JET will default to the logged drive. Subdirectory names must be fully specified without wild cards.

33.7: JET: BACKING-UP AND RESTORING YOUR HARD DISK

33.7.1: BACKING-UP YOUR HARD DISK

JET can be used to back up a hard disk by transferring its contents to multiple diskettes.

Make sure that JET.COM is on the currently logged drive. The easiest way to do this is to put JET.COM on your hard disk, and issue your commands from there. If your hard disk is drive D:, and the first formatted blank diskette onto which you want to back up your hard disk is in drive A:, you type: D>JET D:

A:/D
The /D option causes JET to pause and prompt you to put in fresh target diskettes as needed.

If you are using the /P or /L options to preview the files about to be transferred, you will see a list of the files that are about to be transferred to the diskette currently in the target drive. When you insert another diskette and press any key, you will see a list of the files about to be transferred to the new diskette.

Since JET can only transfer to or from a single subdirectory per command, you have to back up each subdirectory with a separate command.

33.7.2: BACKING UP VERY LARGE FILES

If you have large files on your hard disk, you can make JET split them for more efficient use of diskette space. Type: D>JET D: A:/S

This will work exactly like JET with /D, except that it will fill up all the space on the diskette with files. If a file is only partially copied when space on the first diskette runs out, the rest of it will be copied onto the second diskette. The fragment on the second diskette will have an ! in its file extension, and will be the first entry in the directory. JET does not number the diskettes. You should number them as you make them, so that you can restore them in order.

33.7.3: RESTORING YOUR HARD DISK

To restore the contents of your hard disk, put one of your backup diskettes in drive A: and type: D>JET A: D:/R

JET will restore the files from the first diskette and prompt you to insert the next one. Put the next backup diskette in drive A: and press the carriage return key. When you have used up all your backup diskettes in this way, press any key other than <Enter>, and the DOS drive letter prompt will reappear. You have restored all your files to your hard disk.

JET will fail if the source directory has more than one split file tail (marked by a file extension ending in !) in it. This situation will not arise in normal backing up. As a rule, do not transfer tail files except when restoring with /R or /A. (See below for details about /A.)

JET does not back up files in any special format, so all the backed up files are usable directly from the diskettes, exactly is if they had been COPYed--except the split files, of course. If you want to use a split file without restoring the entire hard disk, you can put it back together by using the /A option. If the split file is called BIGFILE, start by JETting the first part of BIGFILE to a diskette, hard disk, or RAM disk that has space for the whole file. Type: D>JET A:BIGFILE C:

Then, put in the backup diskette that has the next part of BIGFILE on it (usually there are only two, but you could have an enormous file spread over several diskettes), and type: D>JET A: C:/A

JET will transfer only the file with the ! extension, and will append it onto the partial file already on the target diskette. The split file is now back together and ready to use.

33.7.4: HARD DISK BACKUP TECHNIQUES

JET can be made to check and change the archive bit of a file, but not in exactly the same way as BACKUP and RESTORE. When DOS writes to a file, it sets the archive bit to one. BACKUP only transfers files that have their archive bit set to one--that is, BACKUP only transfers files that have been changed since their archive bits were last reset to zero. JET /J does the same. BACKUP also resets the archive bit of both the source and the target file to zero, to signify that it has been backed up. JET /K changes the archive bit, but of the target file only. If you intend to use JET /J later to back up only the files that have been changed since the last backup with JET /K, you have to transfer the backed up files back to the source disk in order to change the archive bits on the source disk.

Another way to keep track of which files need to be backed up is by date. For example, if the last time you backed up your hard disk was May 30, 1984, you would type: D>JET D: A:/I053184

JET would transfer only those files that had dates equal to or later than May 31, 1983. In this way, you would avoid backing up the files that had not changed since the last time you backed up, since DOS updates the date every time it writes to a file.

SUMMARY -- BACKING UP YOUR HARD DISK

Syntax for backing up: JET <source>:[<subdir>...] <target>:[<subdir>...]/[D or S]

Example: JET D: SUB1 A:/D

Syntax for restoring: JET <source>:[<subdir>...] <target>:[<subdir>...]/[R or A]

Example: JET A: D: SUB1/R

Notes: See the explanation above about restoring split files. Subdirectories must be backed up and restored one at a time, with a separate JET command for each. JET does not back up files into a subdirectory unless a target subdirectory name is specified. Diskettes can be used to restore the hard disk in any order, except when restoring split files. These must be restored in order, starting with the first diskette created.

SECTION THIRTY-FOUR
JET OPTIONS

34.1: INTRODUCTION TO JET OPTIONS

After every JET command, you may specify one or more options. All JET options are specified by a letter or group of letters preceded by a "/". There are no spaces between the previous part of the command and the /<option>, nor are there spaces between options. Options may be entered in any order, and may be upper or lower case. For example, you could type:

JET A: B:/P/U/N' **or** jet A: b:/P/u/n

Both of these commands would have exactly the same effect.

34.2: THE JET OPTIONS

### /# Expand

This option causes JET to expand the target subdirectory to hold the total number of entries you specify. You can expand any subdirectory to hold up to 512 entries. To expand a subdirectory, type a command like: A>JET A: B: SUBDIR/256. The subdirectory SUBDIR will be expanded to hold 256 entries.

/A Append

This option is used to restore individual files split by the /S option.

/C COMMAND.COM

This option allows JET to transfer COMMAND.COM.

/D Divide

This option is used to divide the output of a single JET command over multiple target diskettes. It is most commonly used for backing up a hard disk. When the target diskette is filled, JET will pause and prompt you to put a freshly formatted diskette in the target drive, and to press any key to continue. Files transferred with /D are complete, and do not need to be put back together as do files split by /S.

/E Erase

This option causes JET to erase everything that is already on the target

diskette: all files, hidden files, read-only files, and subdirectories. The volume label is the only thing that is not erased. Before doing this, JET asks you: Erase files? (Y/N). You can change your mind if you do not want to take such a drastic step. For your protection, /E does not erase anything when transferring files to a subdirectory.

/ERA ERAse

This option is the same as /E, except that it does not display a message or pause to let you change your mind. /ERA is useful for batch files where you do not want a pause or message.

/F File list

This option instructs JET to transfer only the files listed in a specified file list. /F, /T, /X, or /XCL must be specified when you use a file list. See section 33.5 for details about creating and transferring file lists.

/H Hold

This option holds the file transfer and prompts you to insert a different source diskette before proceeding.

/I sInce

This option lets you transfer only files created or changed since a specified date. The date is specified without spaces, hyphens, or slashes. To transfer files dated May 31, 1984 and later, you would type: A>JET A: C:/IO53184. This option is useful for backing up a hard disk. Just specify one day after your last backup, and only the files that have been created or changed since then will be transferred to your new backup diskettes.

/J chanJed

This option makes JET transfer only files that have their archive bit set to one. Whenever DOS writes to a file, it sets the file's archive bit to one. When the file is backed up using BACKUP, the archive bit is reset to zero. JET/K can also be used to reset the archive bit (see /K). Both BACKUP and JET can use this bit to selectively transfer only files that have been changed since the last time the archive bit was reset.

/K arKive

This option causes JET to reset the archive bit of every transferred file to zero (see /J above). Without /K, JET does not affect the archive bit. Only the copy of the file on the target diskette is changed. If you want to use the archive bit to show whether the file has been changed since it was last backed up with JET/K, you first back up the source with JET/K, then JET/M the files with the reset archive bits from the backup diskette back to the source. The /M is necessary to make JET transfer the backed up files with reset archive bits over the original files, since both have the same time and date.

/L List

This option causes JET to list the files it is about to transfer.

/M Mandatory transfer

This option forces JET to transfer every specified file on the source diskette to the target diskette, even if some files of the same name, time, and date are already present on the target diskette. Normally, JET does not transfer files that are present and have the same time and date on both the source and target diskettes. The /M option is useful if you are using a program that modifies files without resetting the time and date, such as BASIC Random I/O or CONDOR, or if your system does not have a clock and you do not set the time and date when you boot up.

/N New

With this option, a file that is present on both diskettes will only be transferred if the file on the source diskette has the more recent date and time. This prevents you from wiping out newer versions of files by copying older versions over them.

/P Pause

This option causes JET to list all the files it is about to transfer, then pause and prompt you to indicate whether to transfer the files or abort the command. JET will do nothing until you decide whether or not to transfer the listed files. This is a very useful feature when you are using other options that affect what files will be transferred, since it lets you see what files have been selected and lets you change your mind if you do not like the choices. /P is also useful for comparing diskettes when used in conjunction with other options, since it allows you to see the selected files without necessarily transferring them.

/R Restore

This option is used to JET files from multiple source diskettes to a single target disk. The most common use of /R is to restore a hard disk from multiple backup diskettes. See section 33.7.3 for details about restoring a hard disk.

/S Split

This option is used for transferring large files from a large source, such as a hard disk, onto multiple diskettes. It causes JET to split the last file transferred onto each target diskette, so that the first part of the file is on the first diskette, and the second part of the file is on the second diskette. Files split in this way are automatically put back together when the hard disk is restored with JET /R. Individual split files can be put back together (on a hard disk, diskette, or RAM disk) with the /A option.

/T excepT

This option is used when transferring file lists. It causes JET to transfer everything **except** the files listed in the file list. This option can be specified together with the /E or /ERA options.

/U Update

With this option, a file is transferred only if the file name is already present on the target diskette. As usual, the file is only transferred if the times and/or dates of the file are different on the two diskettes. Note that /U allows an older version of a file to be written over a newer version. To prevent this, use /U with /N.

/V Verify

This option makes JET read the transferred files after writing them and compare what it reads to the source files. If there has been an error in the transfer, you will see an error message.

/X eXclusive

This option is used when transferring file lists. It causes JET to erase everything on the target diskette that is not listed in the file list, including hidden files and subdirectories but excepting the volume label, and then to transfer all the files listed in the file list. Before erasing any files, JET will display the message: Erase files? (Y/N) and pause to let you change your mind. /X, /XCL, /T, or /F must be specified when JETting a file list. /T may be used with /X. See section 33.5 for details on using file lists.

/XCL eXCLusive

This option is the same as /X, except that it does not display a message or pause to let you change your mind. /XCL, /X, /T, or /F must be specified when JETting a file list. /T may be used with /XCL. See section 33.5 for details on using file lists.

/Z Zero-length files

This option allows JET to transfer zero-length files other than the volume label and subdirectory names. This can be useful for some programs that create zero-length files as markers or place holders.

SECTION THIRTY-FIVE
ADVANCED FEATURES

35.1: INTRODUCTION TO ADVANCED FEATURES

The JETDRIVE programs give you many special features and options not available with other RAM disk packages. Most of these work with any memory boards; some apply only to systems using JRAM memory boards. Most of these features are controlled, enabled, or disabled with appropriate commands in the CONFIG.SYS file. Here is an overview of the more important advanced features.

* You can silence the warm and cold boot sound effects.

* With MAKEDOS, you can change the size of DOS memory from the keyboard, without setting internal switches or rewriting your CONFIG.SYS file, or test all memory above 64K, including JRAM memory, without damaging the contents of the memory.

* You can allocate a special section of "user" memory in between the end of DOS memory and the beginning of the first RAM for special uses such as holding assembly language programs under development.

* You can specify the sector size, cluster size, and directory size of each RAM disk.

* You can add comments after any Tall Tree Systems driver line in CONFIG.SYS, as long as they are separated from the driver and parameters by a semicolon. Comments following JBOOT.BIN, JDRIVE.BIN, and JSPOOL will be displayed in the boot up message.

* Parity and divide by zero errors are trapped and do not crash the computer.

* You can skip drive letters before and between the RAM disks, to make the RAM disks on different machines have the same letters.

* You can turn off certain features of JBOOT.BIN, such as its special handling of divide by zero errors and parity errors, its memory testing at boot up, and the special <Ctrl><Alt> commands described below.

* Special <Ctrl><Alt> commands:

* <Ctrl><Alt><Del> is redefined to preserve the RAM disks through a warm reboot and play a little sound effect. It is otherwise exactly the same as in DOS.

* <Ctrl><Alt><-> does exactly what <Ctrl><Alt><Del> used to do. This command will either wipe out the RAM disks or corrupt them with about 1 to 5 errors per 100K, depending on the system.

* <Ctrl><Alt><K> (for "Kill" or "Kold boot") causes a cold boot and a sound effect directly from the keyboard. This is useful for COMPAQs and other machines with a time delay built into the power switch. A cold boot destroys all the RAM disks.

* <Ctrl><Alt><F> (for "Flush") allows you to flush the JSPOOL print spooler buffer from within applications programs. You must have JSPOOL to use this command, and the F option must be specified in the CONFIG.SYS file for this command to work.

35.2: ADVANCED JBOOT.BIN SYNTAX

The complete syntax for the JBOOT.BIN line in the CONFIG.SYS file is:

```
DEVICE=JBOOT.BIN <dos>K/[U=<#>K][0=<#>B]
[S=<#>B][R=<#>][X=<#>][<option>...]
```

where the values and options after the / may be in any order, and where:

* <dos> is the number of kilobytes of memory to be allocated to DOS. This number should be a multiple 64K. If it is not a multiple of 64K, DOS will be made to the size specified, but the first RAM disk will start at the next higher multiple of 64K. The memory in between DOS and the beginning of the first RAM disk will be inaccessible and wasted. Note: if you change this parameter and reboot, the sizes and locations of the RAM disks will be changed and their contents will be lost.

* U=<#>K sets the number of kilobytes of memory to be allocated to "user" memory. This memory lies between DOS and the beginning of the first RAM disk, and is not used by either. User memory can be useful for assembly language program development. The sum of DOS memory and user memory cannot exceed 704K. If this number is not an even multiple of 64K, it will be rounded up. Note: if you change this parameter and reboot, the sizes and locations of the RAM disks will be changed and their contents will be lost.

* O=<#>B applies only to systems with JRAM boards. This parameter specifies the maximum number of 64K JRAM banks that can be "turned on" in address space at the same time. Normally, enough JRAM banks are turned on to completely fill the address space used by DOS, and the banks used for the RAM disks are always off except when they are actually being accessed by JDRIVE.BIN. As long as a JRAM bank is "off", it is physically inaccessible to DOS or any other programs, and cannot be accidentally overwritten by errant software under development. The contents of the RAM disk are protected. Only JRAM memory can be switched in and out of address pages in this way. For this reason, if any ordinary memory is used by the RAM disk, it cannot be protected in this way. To fully protect the RAM disk, you must use all your ordinary memory for DOS and user memory, so that the RAM disk is entirely in JRAM memory. In normal use, there is no reason to alter JBOOT.BIN's automatic setting of this value. This value is specified starting at 0, not 1. O=0B would leave one bank on, and O=5B would leave six banks on.

* S=<#>B also applies only to systems that have JRAM boards. Normally, all the JRAM RAM banks that are not used for DOS or user memory are switched in and out of address page D. Very rarely, people have some other piece of hardware or software that makes use of reserved address page D. JBOOT.BIN automatically checks to see if there is ordinary memory installed in the default (or specified) swap bank. If there is, it selects a bank that does not have ordinary memory installed, and uses that as the swap bank. You see a message to this effect at boot up. If address page D is used by another piece of bank- switching software, JBOOT.BIN will not know that, and the two programs will conflict by swapping in the same bank. To prevent this problem, or to prevent JBOOT.BIN from searching for a free swap bank and displaying its message, you can specify a swap bank that you know is free. Page C is often free except in XTs, and page A is free unless you make DOS 704K or place the JSPOOL print spooler buffer on the RAM disk. You can go into lower address pages if you reduce the size of DOS appropriately. This parameter is specified in

decimal, not hexadecimal. That is, to specify page C, use S=12. * R=<#>K allows you to make JBOOT.BIN ignore any regular memory above the specified value. This ordinary memory will not be tested or used for anything.

* X=<#> allows you to make specific address pages off-limits for use as swap banks or ordinary memory. Pages B, C, E, and F (hex) default to being off-limits. (B is used for monitor memory; E is usually used by the JRAM bank select registers, and F is occupied by IBM ROM.) To make a page off-limits, specify it in decimal, as in X=10. To make a page available (page C if you do not have a JRAM board, page E if you set the JRAM-2 or JRAM-2X page select shunt to F), use -X, like -X=13.

35.3: JBOOT.BIN OPTIONS

C <Ctrl><Alt> commands

Enables three of the special <Ctrl><Alt> commands: <Ctrl><Alt><Del>, <Ctrl><Alt><->, and <Ctrl><Alt><K>. (The JSPOOL command <Ctrl><Alt><F> is enabled by option F.) You do not have to specify this option; it is a default value. To disable the special <Ctrl><Alt> commands, specify -C. -C also disables <Crtl><Alt><F>.

D Divide by zero error trap

Enables the special handling of divide by zero errors (see section 35.8). You do not have to specify this option; it is a default value. To disable the divide by zero error trap, specify -D.

E Eagle PC

Disables all parity checking, for use with the Eagle PC or other machines that do not support parity checking. Please do not use this option with any machine that does support parity checking; parity is your only defense against random, dangerous RAM errors.

F Flush the JSPOOL buffer

Enables the special command <Ctrl><Alt><F> to flush the JSPOOL buffer from within applications programs. See the JSPOOL manual. This option will not work if -C is specified.

J Test for JRAM boards

Causes JBOOT.BIN to test for JRAM boards and initialize them if present. You do not have to specify this option; it is a default value. To cause JRAM boards to be ignored and unused, specify -J.

M Test memory after cold boot

Causes JBOOT.BIN to test all memory after a cold boot. You do not have to specify this option; it is a default value. To prevent the memory test and speed up booting up, specify -M.

P Parity error trap

Enables the special handling of NMI parity errors (see section 35.8). You do not have to specify this option; it is a default value. To disable the parity error trap, you specify -P.

Q Quiet

Silences the warm boot and cold boot sound effects. Does not change the short beeps that signal errors and ROM reset (the final stage of a cold boot).

Y Restore date after reboot

Causes JBOOT.BIN to restore the system date after a reboot (DOS automatically restores the system time). You do not have to specify this option; it is a default value. To prevent JBOOT.BIN from restoring the system date, specify -Y.

Z Zero user-defined interrupts at warm boot

Causes JBOOT.BIN to reset all non-DOS, user-redefined interrupts to zero when it reboots. This is the expected behavior for a warm boot. You do not have to specify this option; it is a default value. To preserve redefined interrupts through a warm boot, specify -Z.

35.4: ADVANCED JDRIVE.BIN SYNTAX

The complete syntax for each JDRIVE.BIN line in the CONFIG.SYS file is:

```
DEVICE=JDRIVE.BIN [<size>K]/[B=<#>][C=<#>]
[D=<#>][S=<#>][N=<#>][R or I]
```

Where the letters and numbers after the / may be in any order, and where:

* <size>K is the size of the RAM disk. If no size is specified, the RAM disk will take up all the available memory after DOS and user memory are allocated. You must specify this value to leave room for subsequent RAM disks. If the specified value is not a multiple of 64K, it will be rounded up.

* B=<#> specifies the bank (also called address page or segment) at which the RAM disk will start. This option is useful if you do not have a JRAM board, but do have an ordinary memory board that lets you set banks to pages C, D, and E. You can use this memory as a 64K to 192K RAM disk by specifying B=12 (12 is C in decimal). You will also have to use the JBOOT.BIN X=<#> option.

* C=<#> specifies the RAM disk's cluster size (number of sectors per cluster). The choices are 1, 2, 4, 8, and 16. Larger clusters allow slightly faster access times, but waste more RAM space per file stored. DOS allows a maximum of 4094 clusters per disk, which limits how small you can make the sectors and clusters on a large RAM disk. The default value is C=1 for most RAM disks, but is automatically raised for larger RAM disks or smaller sectors so that the

total number of clusters stays below 4094.

* D=<#> specifies the number of files, including subdirectory names, that can be stored in the RAM disk's directory, from 1 to 512. The default value is 256. Higher values allow you to store more files, but make the directory take up slightly more room on the RAM disk. Each directory entry takes up 32 bytes. The directory space is allocated in units of one cluster, so you must reduce the directory by a cluster's-worth of entries to free any additional space on the RAM disk. Increasing the entries by one will take up a whole additional cluster of RAM disk space.

* S=<#> specifies the RAM disk's sector size. The default value is 512 bytes. The choices are 128, 256, 512, and 1024 bytes. DOS must be patched to use the last choice (see the JFORMAT-2 manual). Larger sectors allow slightly faster access times, but waste more space per file stored. To store a large number of small files, for example batch files, you might use a 64K RAM disk with 128K sectors. JET cannot transfer files between volumes with differing sector sizes, so you must use COPY to transfer files from standard 512 byte sector floppies or hard disks to non-standard RAM disks.

* N=<#> specifies the number of boot sectors in the RAM disk's boot record. The default value is 1. Do not change this value unless you know exactly what you are doing; some DOS functions depend on the boot record being only one sector long.

* R causes JBOOT.BIN to restore an old RAM disk after a warm boot if it is found at the location where the specified one would be installed. Normally, if JBOOT.BIN finds an old RAM disk after rebooting that has a different size or other different parameters from the specified RAM disk (due to a change of boot disks, altering CONFIG.SYS, or a hardware failure), it displays a warning message and makes the disputed drive unavailable, without destroying it. To restore the old RAM disk so that you can salvage its contents, specify /R in CONFIG.SYS and reboot. To install the new RAM disk, cold boot by pressing <Ctrl><Alt><K>. Do not use /R routinely, or you will never know when such a mismatch has occurred. If you have an Eagle PC, you must put /R after every JDRIVE.BIN line in CONFIG.SYS in order to allow the RAM disks to be properly restored after a warm reboot.

* I causes JBOOT.BIN to initialize the RAM disk after a warm reboot. Any previous RAM disk at the same location will not be restored, but will be wiped out and replaced with the new, empty one.

35.5: SKIPPING DRIVE LETTERS

You can cause your RAM disks to have higher drive letters by adding dummy RAM disks to your CONFIG.SYS file. These dummy disks do not function and take up almost no memory, but do use up a drive letter. This feature allows you to set up RAM disks with the same drive letters on computers with different numbers of physical drives. You can specify up to 16 dummy disks in a single line, and you can use any number of dummy disk lines at any point in CONFIG.SYS after JBOOT.BIN. DUMDISK.BIN must be on the boot disk. For example, to cause two drive letters to be skipped before the first RAM disk, you could use a CONFIG.SYS file like:

```
A>COPY CON CONFIG.SYS
DEVICE=JBOOT.BIN 192K
DEVICE=DUMDISK.BIN N=2
DEVICE=JDRIVE.BIN
<F6>
```

35.6: MAKEDOS

MAKEDOS is a command that lets you change the size of DOS memory without having to change your CONFIG.SYS file. To use MAKEDOS, you must have MAKEDOS.COM on the currently logged drive, and <dos> must be specified as 0 in CONFIG.SYS. (Specifying <dos> as 0 causes DOS to be allocated as much memory as is shown on the motherboard memory switches, or as much as has been subsequently specified by MAKEDOS). To change the size of DOS, just type a command like:

```
A>MAKEDOS 256K
```

The computer will automatically reboot, and the size of DOS will be changed to 256K. This procedure will destroy the contents of the RAM disks (unless you have requested a "change" to the size of DOS that is already there). Save everything of value onto a physical diskette before changing the size of DOS.

35.7: PARITY AND DIVIDE BY ZERO ERROR HANDLING

JBOOT.BIN includes two useful features that are not directly related to the RAM disks: special handling of parity and divide by zero errors. Normally, these errors can crash the machine, hang up the keyboard, or stop your application program in mid-stride like a <Ctrl><C> command.

JBOOT.BIN intercepts the interrupts signaling these two errors, prints a warning message, and gives you the option of ignoring the error, warm rebooting, or cold rebooting from the keyboard. If a divide by zero error is detected, you will see:

```
*ERROR* DIV OVERFLOW **
(I=ignore, W=warm, Q=coldboot)
```

If a parity error is detected, you will see:

```
+++ CS:IP,SS:SP,DS:SI,ES:DI =
xxxx:xxxx,xxxx:xxxx,xxxx:xxxx,xxxx:xxxx
*ERROR* << NMI PARITY ERROR >>
(I=ignore, W=warm, Q=coldboot)
```

In the case of a parity error, you not only have the three choices of what to do, but you also see the contents of the various registers and pointers at the time of the error. DS:SI tells you the logical address of the parity error.

Usually, the best thing to do is to ignore the error, save your work onto a blank physical diskette, and then reboot. Be sure you do not save over your old copies of the files, because the ones you are saving from the RAM disks may be damaged. Check your saved files carefully before trusting them. If the keyboard is hung up, you will have to say goodbye to your work and cold boot by flipping the power switch.

SECTION THIRTY-SIX
ERROR MESSAGES

These error messages come from the JETDRIVE programs. If you see an error message that is not on this list, check the DOS manual. Messages that do not involve errors are not included in this list. Many of these messages are preceded by the standardized warning *ERROR*.

Algorithm failure

This message can be caused by using JET with /D/S, an illegal and meaningless combination. It can also be caused by trying to JET from a directory with more than one split file tail in it (see section 33). Failing these possibilities, this message indicates a severe error, usually due to a damaged copy of JET, other damaged data, or a hardware failure. Use COMP to compare your copy of JET to the original on your Tall Tree Systems distribution diskette. If they match, run CHKDSK to see if there is any damage on the logged drive. If the volume in the logged drive is OK, use MAKEDOS 0 to run the Tall Tree Systems memory diagnostic, and cold boot to run the IBM memory diagnostic. If you do this several times without finding any memory errors, use COMP to check for damage in your other files. If you still find no errors, suspect a hardware failure.

x = bad entry

This message tells you that part of the command you typed was illegal, and it points out the illegal character. Retype the command correctly, replacing the indicated character with a legal one.

-Bad JRAM, board=xx, bank=xx, quad=xx

This message means that the memory diagnostic in JBOOT.BIN has found a bad bank of JRAM memory. The board number refers to the board select shunt settings: boards 0-3 have their page select shunt set to position E, board 4-7 are set to page F. The bank refers to the vertical columns of chips on the JRAM, where bank 0 is closest to the gold plated edge connector, and bank 7 is at the end of the board. The quad refers to the 64K virtual bank within a bank of 256K chips. Since you have to replace the whole 256K chip even when only one quadrant fails, this information is usually not important. 64K chips have only quad 0.

-Bad Mem bank=

This message means that the memory diagnostic in JBOOT.BIN has found a bad bank of ordinary memory. The number (0 to F, in hex) specifies which bank is bad.

Bad SWAP page not free: changed

The swap bank that you specified after JBOOT.BIN is already occupied by ordinary memory or is "off limits". JBOOT.BIN has selected a different page. See section 35.

Bad values (try "makedos 64k")

You have specified an illegal parameter. Retype the command correctly.

++ CS:IP,SS:SP,DS:SI,ES:DI = xxxx:xxxx,xxxx:xxxx,xxxx:xxxx << NMI PARITY ERROR >> (I=ignore, W=warm, Q=coldboot)

A parity error has been found. This means that at least one byte of data somewhere in memory or on some drive has been damaged, and the location of that byte, as well as the contents of the various registers and pointers, are shown in the first line of the message. The best thing to do is to press I, save the contents of the RAM disk to a blank physical diskette, and then cold boot. Be aware that you may have a damaged byte in your saved data somewhere; a message will appear warning you that you take this risk by ignoring such an error. Check your saved data carefully before you trust it. Sometimes when you see this message, the keyboard will be hung up, that is, the computer will not respond to anything you type. In that case, you have to say goodbye to the data on your RAM disk and cold boot by flipping the power switch.

DIV OVERFLOW ** (I=ignore, W=warm, Q=coldboot)

Without JBOOT.BIN, a divide by zero error will crash the computer, forcing you to cold boot and lose the contents of the RAM disk. Divide by zero errors are usually caused by faulty logic in a program. JBOOT.BIN gives you three choices when a divide by zero error occurs. If you ignore the error, control goes back to the program that caused it, and the faulty division is never executed. A warning message will appear, reminding you that you are in for trouble as soon as the program tries to use the value it intended to divide by zero. If you warm boot, you abort the faulty program and have the chance to save the contents of your RAM disk before anything can happen to it.

Error appending to partially full sector

You are restoring files with JET /R or /A. The first backup diskette was formatted with 512 byte sectors, and the second was formatted with 1K sectors. This is possible if you backed up onto eight inch drives and 5 inch diskettes, or if you have specified 1K sector formats with JFORMAT-2 (another Tall Tree Systems product). Try deleting everything you already restored, copying a very small file to the target disk, then restoring with JET /R again exactly as you did before.

Error opening file list file

You did not include /F, /X, or /XCL in a JET command with a file list, or you specified one of these options without specifying a file list. Retype the command correctly.

Error opening Output Drive

Either you specified an incorrect target drive letter, or something is wrong with the target drive. Check to make sure that there is a diskette in the target drive, that the diskette is formatted, and that the door of the drive is closed.

Invalid drive specification

You have tried to JET files to or from a drive that does not exist. Retype the command with legal drive letters.

JET unable to transfer 8" to 5" singles

JET cannot transfer files between diskettes with different size sectors. When it tries, you see this message and no files are transferred. The only time you will see sectors that are not the standard 512 bytes is if you are using certain formats of eight inch diskettes or if you have specified an odd sector size for a RAM disk. Use COPY to transfer the files.

Max subdirectory entries = 512

You have tried to expand or create a subdirectory using JET /# to hold more than 512 entries. This is not possible. Retype the command specifying 512 entries or less.

** No JBOOT in CONFIG.SYS **

You cannot create a RAM disk without JBOOT.BIN in the CONFIG.SYS file. Reread section 31 and fix your CONFIG.SYS file according to the instructions.

No action needed

MAKEDOS returns this message when you try to change the size of DOS to the size that it already is.

No room in output directory for a subdirectory

The target directory or subdirectory is completely full. (Directories hold a finite number of entries, independent of the size of the files. See section 33.) Delete a file in order to make room for the subdirectory name. If the target is a subdirectory, you can enlarge it with the /# option to make room for the new subdirectory.

<<No space avail>>

There is not enough memory left to create the specified RAM disk at all, even truncated. Change CONFIG.SYS to reflect the amount of memory you actually have, and reboot.

NOT 64K multiple, changed

If you specify a DOS size and "user" memory size that do not add up to a multiple of 64K, the values you specified will be used for DOS and "user" memory. However, the first RAM disk will start at the next higher multiple of 64K. The memory in between the end of DOS or "user" memory and the beginning of the first RAM disk will be inaccessible for use. No damage is done, but memory is wasted.

Not enough Memory: DOS or USERK changed

You have specified more DOS and user memory than is installed in the machine. All your available memory (which will be less than what you specified) will be allocated to DOS. No memory will be left over for RAM disks, the JSPOOL buffer, or "user" memory.

Number too large

A parameter was specified with an illegally large number. Retype the command correctly.

Output Directory Overflow

The target directory is not large enough to hold all the file names to be transferred. Transfer fewer files, delete some files from the target directory, or use the /E option to delete all the data on the target diskette. If the target is a subdirectory, you can enlarge to up to 512 entries with the /# option.

Output Volume Allocation Problems - Run CHKDSK

The target diskette is severely screwed up. Usually, this message means that the File Allocation Table does not correspond to the directory. Sometimes the diskette can be fixed by running CHKDSK/F. Try transferring your files to a different diskette.

Output Volume too small

This message means that the target diskette does not have enough space for the files to be transferred by JET. Either delete some files on the target diskette, put in a diskette with more free space, or specify fewer files to be transferred, and try again. If you are transferring an enormous file, you may want to use the /S option to split it between two target diskettes.

Parameter Error

You have typed an illegal JET command. Retype it correctly.

Previous clusters, size, or version different.

(Cntrl-Alt-K will clear it, or use /R option.)

This message is displayed when JBOOT.BIN tries to create a RAM disk, but finds one already in memory with different parameters or that was created by a different version of JETDRIVE. This can happen when you reboot with a different diskette in drive A: than the one you first booted up with, but it can also be due to altering CONFIG.SYS or to a hardware failure. An old RAM disk will only be recognized if it starts at the same place in memory as the new one trying to be installed--that is, if the old DOS plus "user" memory equals the new DOS plus "user" memory. If an old RAM disk is found, the new one is not installed, and the old one is made unavailable but not destroyed. No data is lost. To restore the old RAM disk, temporarily put /R after JDRIVE.BIN in your CONFIG.SYS file and reboot. To get rid of the old RAM disk and install the new one, cold boot by pressing <Ctrl><Alt><K>.

RAM Disk found. DANGER: check if corrupted

You are rebooting (possibly by calling MAKEDOS), and JBOOT.BIN has found an old RAM disk in memory left over from some previous work on the computer. It may or may not be corrupted with random errors at the rate of about 1 to 5 errors per 100K. Unless you want to try to save something stored on the old RAM disk, it is best to cold boot by pressing <Ctrl><Alt><K> and get rid of the possibly damaged disk and its contents.

Read Error - Abort, Retry, Ignore

Some kind of failure has occurred in the file transfer. If some files have already been written to the target diskette, the data on the diskette has been changed, but the File Allocation Table and the directory have not yet been updated. First press R to retry. If that fails a few times, press I to ignore the error and continue. If that also fails, you will have to press A to abort. If you have to use I or A, be sure to check the target diskette when you are done. Run CHKDSK, and look at a few files. The diskette may be badly scrambled.

==ROM RESET==

This message will flash briefly on the screen when control passes from JBOOT.BIN to the IBM ROM bootup routines. Then the screen will go blank. You will probably see this message only when you press <Ctrl><Alt><K>. This means you are about to lose the contents of the RAM disks.

SWAP page not free: changed

JBOOT.BIN's default swap page (D hex, or 13 decimal) is occupied by installed ordinary memory, and JBOOT.BIN has found a different bank to use as a swap bank. To prevent this message, use the S=<#>B option in CONFIG.SYS to specify a bank you know is free in your system. See section 35.

Too many RAM disks

You have specified more than four RAM disks in CONFIG.SYS. Change CONFIG.SYS to specify four or fewer, and reboot.

Truncated

There was not enough memory to install the specified size of RAM disk. JDRIVE.BIN has installed a smaller RAM disk.

Unable to find input directory or file name

You specified a subdirectory or file name that does not exist on the source diskette. Retype the command with the correct file name and subdirectory path.

User takes RISK!

If you choose to ignore a parity error or a divide by zero error, this message reminds you that a potentially dangerous error is hiding somewhere in your programs or data.

Write Error - Abort, Retry, Ignore See "Read Error".

SECTION FORTY
INTRODUCTION TO JSPOOL

40.1: WHAT IS JSPOOL?

JSPOOL is a software print spooler that lets your computer print files, graphics, or any other type of printed output while running any program at the same time. The printer will print at full speed or close to it; the program running at the same time will typically run with only a four to seven percent loss in speed.

Unlike other print spoolers, JSPOOL does not slow the computer down except when it is actually printing. JSPOOL is an installable device driver, so it is automatically installed when you boot up. No AUTOEXEC.BAT file is needed. Because it is an installable device driver, JSPOOL is compatible with virtually all other software.

JSPOOL lets you put the spooler buffer on the JETDRIVE RAM disk. If you have a JRAM board, this means that you can have a large spooler buffer and still use up to 640K for DOS.

JSPOOL can be "smart"--it can set custom tab stops, adjust margins, truncate lines at any length, and so on--or it can be just as dumb as any other spooler, simply passing data to the printer without altering it at all.

With JSPOOL, you can send commands to the printer (such as compressed mode, line truncate, pause, and so on) straight from the keyboard. You can also embed these commands directly in your files. For example, you could embed a command to cause a single word to be printed in compressed mode, or to cause a particular file to automatically print out on the desired printer.

JSPOOL lets you select between all the serial and parallel printers connected to your computer, including those using COM3 and COM4, which are initialized by JBOOT.BIN for people who have JRAM serial-serial modules or other multiple serial port hardware.

40.2: COMPATIBILITY

JSPOOL is designed for the IBM PC and XT, and will work with many compatible computers that are running PC DOS or MS DOS. It can also be used in an IBM AT. The computer must have a single or double-sided diskette drive to read the distribution diskette.

A system using JSPOOL must have at least 64K of RAM, although we suggest a minimum of 128K for a useful system. JSPOOL will work with any serial or parallel printer, including serial printers that use the XON/XOFF protocol.

40.3: WHAT YOU GET

Your Tall Tree Systems distribution diskette contains a number of files, including programs for JETDRIVE as well as JSPOOL. The essential JSPOOL files are:

JSPOOL.BIN

This file is the installable device driver that creates and maintains the JSPOOL print spooler.

SPOOLCOM.COM

This file allows you to send commands to the spooler and printer directly from the keyboard. It also provides a status report including what type of buffer is in use, and how many characters remain to be printed.

MAKEFILE.COM

This file allows you to put the JSPOOL buffer on the JETDRIVE RAM disk. It is also a generalized utility for investigating and creating files, and adding or removing the "hidden" and "read only" file attributes.

JSPOOL.ASM

This file is the source code for JSPOOL.BIN.

40.4: WHAT JSPOOL DOES

JSPOOL establishes a reserved area of RAM memory called a buffer. JSPOOL intercepts all output to the printer and sends it to the buffer instead. Because the buffer can accept data at electronic speeds, the computer can dump all its printer output into the buffer very rapidly, without having to wait for the the printer to actually type each character. Once the output is all in the buffer, the computer is free to do other things.

Eighteen times every second, JSPOOL interrupts whatever the computer is doing for a brief time and sends a few characters from the buffer to the printer. JSPOOL runs "in the background", which means that it occupies only a small fraction of the computer's time, and takes second priority to whatever program is running "in the foreground".

JSPOOL lets you select among three different types of buffers. In its default mode, JSPOOL reserves a buffer of any size you choose in the memory normally used by DOS to hold programs and data currently being used. If you are using the JETDRIVE RAM disk package, you can also put the JSPOOL buffer on the RAM disk. This option allows you to change the size of the buffer without changing CONFIG.SYS and rebooting, and it allows you to reboot if necessary without destroying the contents of the spooler buffer. If you have a JRAM board, putting the spooler buffer on the RAM disk also allows you to have a large spooler buffer and still allocate up to 640K to DOS. Finally, you can define an existing file on the JETDRIVE RAM disk as the buffer. In this way, you can print the file and be able to run another program at the same time, without using up any additional RAM space for a buffer.

SECTION FORTY-ONE
INSTALLING JSPOOL

41.1: INSTALLING JSPOOL

1. Make a copy of your usual boot diskette, and put it away as a backup. If you have a hard disk, ignore this step.

2. Copy JSPOOL.BIN, SPOOLCOM.COM, and MAKEFILE.COM from your Tall Tree Systems distribution diskette onto your new boot diskette.

3. Since you have already installed JETDRIVE, you already have a CONFIG.SYS file. To install JSPOOL, you will have to add one more line to CONFIG.SYS. Display your current CONFIG.SYS file by typing:

```
A>TYPE CONFIG.SYS
```

You will see something like:

```
A>TYPE CONFIG.SYS
DEVICE=JBOOT.BIN 192K/F
DEVICE=JDRIVE.BIN
```

If you have installed other installable device drivers, your CONFIG.SYS file will include them, too. To add JSPOOL to CONFIG.SYS without losing the devices that you have already installed, just retype the old CONFIG.SYS file before adding JSPOOL.BIN. Continuing the example above, you would type:

```
A>COPY CON CONFIG.SYS
DEVICE=JBOOT.BIN 192K/F
DEVICE=JDRIVE.BIN
DEVICE=JSPOOL.BIN
<F6>(Screen will show ^Z)
```

The COPY CON CONFIG.SYS line tells the computer to create a file called CONFIG.SYS that will contain whatever is typed in until function key <F6> is pressed.

The CONFIG.SYS file above will install JSPOOL with a buffer in DOS memory (as opposed to a buffer on the JETDRIVE RAM disk). The buffer will take up one quarter of the available DOS memory. To specify a larger or smaller buffer, specify the size buffer you want (in Kilobytes) after JSPOOL.BIN. For example, you could use:

```
    DEVICE=JSPOOL.BIN 50K
```

When you are deciding how large to make the buffer, remember that it has to be big enough to hold the files or other output you want to print. If you want to set up the buffer on a RAM disk, follow the instructions in the section on MAKEFILE. For more details about buffer sizes and types, see section forty-four.

4. Reboot the computer by pressing <Ctrl><Alt><Del>. This installs JSPOOL and displays all the usual messages plus the lines:

```
JSPOOL    Ver 2.10  Ser.#060132
Copyright(c)1983 Tall Tree Systems
16K. Buffer size=16384 bytes.
```

5. Make a backup copy of your new boot diskette.

SECTION FORTY-TWO
SIMPLE PRINTING WITH JSPOOL

42.1: INTRODUCTION TO PRINTING WITH JSPOOL

You can send text or graphics to the printer in any way that you have used before, and JSPOOL will intercept the data, dump it into the spooler buffer, and free the computer to do anything else you want it to.

42.2: PREPARING JSPOOL TO PRINT

42.2.1: SIMPLE JSPOOL COMMANDS

Before you print, you generally have to tell JSPOOL which printer you want to use, and, if it is a serial printer, what kind of serial printer it is. The simplest way to give such spooler commands is to use SPOOLCOM. Only the essential, basic SPOOLCOM commands are covered in these sections. For more details about SPOOLCOM and special spooler and printer commands, see section 43.

SPOOLCOM commands replace the DOS MODE commands that you may have been using. Do not use MODE with JSPOOL.

If you are planning to print graphics or microjustified text, issue the command:

```
    A>SPOOLCOM D1
```

before starting to print. For details about this command and why it is important, see section 43.5.

42.2.2: SELECT A PRINTER

42.2.2.1: THE DEFAULT PARALLEL PRINTER

If you do not tell it otherwise, JSPOOL assumes that you want to use a parallel printer connected to your first parallel port (LPT1). If you have a parallel printer and only one parallel port (a common configuration), then you can use JSPOOL without giving it any further instructions. Section 42.3 describes how to start printing.

42.2.2.2: A DIFFERENT PARALLEL PRINTER

JSPOOL identifies up to three parallel printers with the numbers 4, 5, and 6 (0 through 3 are serial printers). The default printer is number 4. To select a different parallel printer, type a command like: **A>SPOOLCOM S5**.

This command will Select the parallel printer connected to your second parallel port (LPT2). The "S" specifies that the number following it will Select a printer, and the number refers to the particular printer. Printer number 4 is the printer connected to your first parallel port (LPT1); number 5 is connected to LPT2, and number 6 is connected to LPT3. If you have only one parallel port, the printer connected to it will always be printer 4 (JSPOOL's default printer). If you have more parallel ports, the easiest way to determine which printers are which is by trial and error. One hint: the numbers are allocated in order, so if you have two parallel ports, for example, they will be numbers 4 and 5.

After you enter any SPOOLCOM command, you will see a message like:

```
JSPOOL Buffer: -Inside DOS-,
Type=0, Size=16383 bytes
Number of chrs not yet printed= 8 bytes.
```

For now, you can ignore this message. The "characters not yet printed" value refers to the SPOOLCOM command itself, which does not print. The buffer is empty again in a few milliseconds.

Now that you have selected a parallel printer, you are ready to print. See section 42.3 for the basics of printing.

42.2.2.3: A SERIAL PRINTER

Before you select a serial printer, you have to tell JSPOOL at what speed to send data to the serial printer. Different

serial printers accept data at different speeds, or "baud" rates. JSPOOL defaults to 9600 baud, the fastest commonly used baud rate. Check your serial printer manual to see what baud rate it uses. To select a different baud rate, type a command like: **A>SPOOLCOM B2**. This command would set JSPOOL to send data to a serial printer at 300 baud. The number in the command is one of the

Code:	B0	B1	B2	B3	B4	B5	B6	B7
Baud rate:	110	150	300	600	1200	2400	4800	9600

After you enter this command, you will see a message like:

```
JSPOOL Buffer: -Inside DOS-, Type=0, Size=16383 bytes Number of chrs not
yet printed= 7 bytes.
```

For now, you can ignore this message. The "characters not yet printed" value refers to the SPOOLCOM command itself, which does not print. The buffer is empty again in a few milliseconds.

JSPOOL identifies up to four serial printers with the numbers 0, 1, 2, and 3. After you have set the baud rate for the serial printer you want to use, type a command like: **A>SPOOLCOM S0.**

This command will Select the serial printer connected to your first serial port (COM1). Printer number 0 is the serial printer connected to your first serial port (COM1); number 1 is connected to COM2, and 2 and 3 are connected to COM3 and COM4. If you have only one serial port, the printer connected to it will be printer 0. If you have more serial ports, the easiest way to determine which printers are which is by trial and error. One hint: the numbers are allocated in order, so if you have two serial ports, for example, they will be numbers 0 and 1.

After you enter this command, you will see a message like:

```
JSPOOL Buffer: -Inside DOS-, Type=0, Size=16383 bytes
Number of chrs not yet printed= 8 bytes.
```

Again, you can ignore this message.

Now that you have selected a serial printer, you are ready to print. The next section describes the basics of printing.

42.3: PRINTING WITH JSPOOL

42.3.1: THE AUTOMATIC METHOD

After JSPOOL is set to use the desired printer as explained above, you can forget about it. No matter how you send text to the printer, JSPOOL will intercept it, quickly dump it into the spooler buffer, and free the computer to do anything else you want. JSPOOL works automatically with any application program that creates printed output. Just use the same print commands that you always have with your programs, and JSPOOL will take the printing from there. JSPOOL even automatically spools output from <Shift><PrtSc>, <Ctrl><PrtSc>, and <Ctrl><P>. (For details about these printing methods, see section 42.3.1.)

The only limitation is that the JSPOOL buffer should be big enough to hold all the data that you want printed. If the buffer is not large enough to hold all the printer output, you lose most of the advantages of JSPOOL. You will have to wait for the printer to physically type the first part of the text or graphic before there is room in the buffer for the last part of the printer output. No data will be lost, but the computer will not be free to do anything else until the last of the printer output has been stuffed into the buffer.

42.3.1: <SHIFT><PRTSC>, <CTRL><PRTSC>, AND <CTRL><P>

JSPOOL works with the three special print commands supplied by DOS: <Shift><PrtSc>, <Ctrl><PrtSc>, and <Ctrl><P>. Since these commands usually send only a small amount of data to the printer at a time, JSPOOL

will not enhance them much. However, JSPOOL does allow you to use them, unlike PRINT. With JSPOOL, the output of these commands is simply added to the buffer. JSPOOL will print the output from any of these commands after it has finished printing whatever is already in the buffer.

Pressing <Shift> and <PrtSc> simultaneously will cause your printer to print whatever is showing on the screen. You can use this command from within any application program, as well as from the DOS prompt.

Pressing <Ctrl> and <PrtSc> simultaneously causes your printer to print everything that the computer displays, starting at the time you issue the command. The computer will not print whatever is already on the screen. In effect, <Ctrl><PrtSc> makes the printer behave like a teletype, printing both what you type and whatever the computer outputs to the screen. The printer will continue printing everything that appears on the screen, line by line, until you press <Ctrl><PrtSc> again. As with <Shift><PrtSc>, you can issue this command directly from DOS, or from within an application program.

Finally, <Ctrl><P> is exactly the same as <Ctrl><PrtSc>. It is not a documented feature of DOS, but it works.

42.4: STOPPING JSPOOL FROM PRINTING

Eventually you may send something to the spooler buffer that, on second thought, you do not really want to print after all. To stop the printing, type: **A>SPOOLCOM FLUSH.** This command instantly flushes (empties) the spooler buffer. JSPOOL has nothing to print, and printing stops immediately. If your printer has an internal buffer, the printer will print out anything that is already in its internal buffer, and then stop.

SECTION FORTY-THREE
SPOOLER COMMANDS

43.1: INTRODUCTION TO SPOOLER COMMANDS

There are a variety of things you can instruct JSPOOL to do, from flushing printed output by setting margins and tabs, truncating lines, pausing, and so on. The more of the features your printer, text editor, and application programs already have, the fewer of these JSPOOL commands you will need. Still, no matter how clever your system is, you will always need to have a certain amount of control over the spooler: you will want to know if the buffer is getting full and how much remains to be printed, you will occasionally want to flush the spooler buffer to prevent unwanted material from being printed, you will want to select various printers, and you will have to fortify JSPOOL to withstand the onslaught of weird and confusing characters used by graphics printing programs.

There are three ways to issue commands to the spooler. First, you can enter spooler commands directly from the keyboard by using SPOOLCOM. These commands take effect immediately.

Second, you can embed the same commands directly in the text to be printed. These embedded commands will not print, but will take effect automatically when the printer comes to them. For example, you could embed spooler commands to make a single word print in compressed mode, or you could embed a command that would make the printer stop anywhere in your file to enable you to change to a different type element or ribbon.

Finally, you can put spooler commands in your CONFIG.SYS file. For example, you might add commands to your CONFIG.SYS file to automatically set JSPOOL to use a particular printer. In this way, you can customize JSPOOL without cluttering up your AUTOEXEC.BAT file with spooler commands.

43.2: SPOOLCOM

To enter spooler commands directly from the keyboard, use SPOOLCOM. For example, the spooler command S5 selects the printer connected to your second parallel port. To make JSPOOL use this printer, type: **A>SPOOLCOM S5** JSPOOL would immediately switch to the desired printer. If JSPOOL was in the middle of printing a file, the file would suddenly start printing on the new printer. Because the command does not take effect until you press <Enter>, you can backspace, delete, and correct the command before you issue it. SPOOLCOM commands must be issued from a DOS prompt; SPOOLCOM does not work from within applications programs. After you issue a SPOOLCOM command, you will see a status report describing the current spooler buffer and its contents (see below for details). If you do not want to see this report, for example if you are issuing a SPOOLCOM command in a batch file, you can redirect it by using a command like:

```
A>SPOOLCOM S5>NULL
```

In addition to allowing you to enter all the spooler commands explained in section five, SPOOLCOM lets you enter a few special commands that cannot be given in any other way. Here are the special SPOOLCOM commands:

SPOOLCOM

If you enter SPOOLCOM with no command after it, JSPOOL will report the status of its buffer. A typical status report looks like:

```
JSPOOL Buffer: -Inside DOS-,
Type=0, Size=16383 bytes.
Number of chrs not yet printed= 3798 bytes.
```

The first line describes the current JSPOOL buffer. If you have put the buffer on the JETDRIVE RAM disk, information about that buffer would appear in the first line (for details about different types and sizes of buffers, see section six). The bottom line shows how many characters are in the buffer, waiting to be printed.

SPOOLCOM FLUSH

This command flushes (empties) the JSPOOL buffer. Printing stops immediately, since JSPOOL has no more characters to send to the printer. If your printer has an internal buffer, the data already in the internal buffer will be printed, and then the printer will stop.

SPOOLCOM RESET

This command flushes (empties) the JSPOOL buffer and resets many JSPOOL parameters to their default values. SPOOLCOM RESET can be useful if you have issued the D2 or D3 (Dumb spooler, see section 43.5) commands. These commands cause JSPOOL to ignore all further commands. SPOOLCOM RESET resets the Dumbness parameter to D0, the default value, and restores your control over the spooler. After issuing this command, you will have to reissue Baud rate, Select printer, Line truncate, and other settings.

SPOOLCOM EJECT

This command immediately ejects one page from the printer. As with other printer commands, if your printer has an internal buffer, the contents of the internal buffer will be printed before the command takes effect.

SPOOLCOM LINES

This command immediately ejects seven lines from the printer. If your printer has an internal buffer, the contents of the buffer at the time the command was issued will be printed before the command takes effect.

SPOOLCOM .PRINT THIS TEST LINE

If you follow SPOOLCOM by a period and some text, that text will be sent to the spooler buffer. Pressing the <Tab> key within the text will cause a carriage return and line feed at that point, and pressing <Enter> ends the text and sends it to the spooler. For example, you might type:

```
A>SPOOLCOM .THIS IS LINE ONE <Tab>
THIS IS LINE TWO
```

If everything is working properly, the printer will print:

```
.THIS IS LINE ONE
 THIS IS LINE TWO
```

This command is useful for commenting and labeling printouts, and for testing your printing setup and the current settings of various JSPOOL commands such as C (Compressed mode), M (Margin), and so on.

43.3: EMBEDDING SPOOLER COMMANDS IN THE TEXT

Another way to issue spooler commands is to embed them directly in the file to be printed. Spooler commands embedded in a file must be preceded by the special marker sequence @#$, and followed by a semicolon.

Commands embedded in a file take effect when JSPOOL "prints" them, so they offer very fine control over your printer. For example, to issue the C (Compressed mode) command from within a file, make a line of your file look like:

```
and here is what @#$ C;compressed
mode@#$ -C; print looks like.
```

When you printed the file with JSPOOL, the words "compressed mode" would be printed in compressed mode. The spooler commands would not print or cause any spaces.

The characters @#$ (pronounced "at, pound, dollar sign") were selected because they are conveniently all in a row across the top of the keyboard, yet form an extremely unlikely combination.

You can follow the @#$ sequence with any number of legal spooler commands, as long as the series of commands is terminated with a semicolon. All spaces in the series are disregarded, but extra letters and comments are not allowed.

A variation of this method of issuing commands is useful when you are printing with <Shift><PrtSc> or <Ctrl><P>. These commands print whatever is on the screen. So, you can issue spooler commands by simply typing in @#$, followed by the desired commands and a semicolon. The command will show on the screen, but will will cause the desired spooler action without printing. To make the command take effect immediately, rather than when the printer gets to it, use the prefix @#$i. With this prefix, the commands take effect as soon as they are typed in, like SPOOLCOM commands. Never embed commands with the @#$i prefix in a file, because they will take effect when they are read into the buffer, long before the printer gets to them in the file. You will have no control over what parts of the file will be affected by the command.

43.4: PUTTING SPOOLER COMMANDS IN CONFIG.SYS

If there are certain commands that you routinely use with JSPOOL, such as selecting a serial printer and setting the baud rate for it, you may want to make these commands take effect automatically when you boot up. One way to do this is to put SPOOLCOM commands in your AUTOEXEC.BAT file. Another, less visible, way is to put the commands in your CONFIG.SYS file.

To put spooler commands in your CONFIG.SYS file, follow the buffer size with /@#$i and whatever spooler commands you want. For example, you might include a line like this in your CONFIG.SYS file:

```
DEVICE=JSPOOL.BIN 45K /@#$i B2 S0
```

This line would cause JSPOOL to automatically select the 300 baud serial printer connected to your first serial port (COM1). Remember, incidentally, that you must specify the baud rate **before** selecting a serial printer.

Spooler commands issued in the CONFIG.SYS file do not have any special precedence over others. If you used the example line above, for instance, you could switch to your parallel printer at any time by typing: **A>SPOOLCOM S4.**

43.5: THE SPOOLER COMMANDS

All of the commands below may be entered in any of the ways described above: with SPOOLCOM, embedded in the text of a file, or in your CONFIG.SYS file. Any number of commands may be entered at a time. For example, you could enter the command:

```
A>SPOOLCOM B3 S1 C R M10 E
```

Commands may be in upper case, lower case, or in any combination of upper and lower case. All commands are reset to their default values when you reboot.

B Baud rate
Default: B7 (9600 baud)

This command sets the baud rate for a serial printer. To use a serial printer, you must first set the appropriate baud rate, then select the desired printer. See your serial printer manual for the baud rate codes it uses. The baud rate codes are:

Code:	B0	B1	B2	B3	B4	B5	B6	B7
Baud rate:	110	150	300	600	1200	2400	4800	9600

To set the baud rate to 300 baud, for example, type: A>SPOOLCOM B2.

C Compressed mode
Default: -C (Normal mode)

This command specifies compressed mode printing for parallel matrix printers that are capable of it. To print a file in compressed mode, use a command like: A>SPOOLCOM C. Or, you can embed @#$ C in a file to start compressed mode printing at any point in the file. The command -C cancels compressed mode.

If a parallel printer is being used, command C makes JSPOOL send the IBM/EPSON standard printer command ESC 0F (hex) for compressed mode, and command -C makes JSPOOL send the standard ESC 12 (hex) for normal mode.

This command works somewhat differently for serial printers. The printer will come on in its default printing mode. When a serial printer is being used, command C sends the C. ITOH command <Esc> E for 12 pitch, and command -C sends the C. ITOH command <Esc> Q for 17 pitch. If the printer's default is 10 pitch, the only way to get back to 10 pitch after using either command C or command -C is to use JTYPE/I to send <Esc> N directly to the printer, or to turn the printer off and on to reestablish the printer default. Some serial printers use different escape sequences to change pitch. To control these printers, use JTYPE/I to send the correct command directly to the printer. See your serial printer manual for the escape codes it uses.

-C Cancel compressed mode

This command cancels compressed mode printing for parallel matrix printers, and selects 17 pitch for C. ITOH compatible serial printers. See the discussion of the C command for details.

D Dumb spooler
Default: D0 (All data processed)

Unlike other spoolers, JSPOOL is "smart". JSPOOL scans all text sent to it to find the marker sequence @#$ that precedes spooler commands embedded in the text. In addition, control characters such as <Tab>, <Line feed>, and so on are counted and used by JSPOOL to modify the text as it is sent to the printer:

spaces are added to create the desired margins, tab characters are translated into the required number of spaces, lines are truncated at the specified length, and so on. For many applications, these are useful features.

However, there are situations in which JSPOOL's cleverness can be a problem. When many complex characters are sent to the printer, for example when you are printing graphics or using microjustification, JSPOOL may occasionally find a fortuitous @#$ in the data and execute an unintended, inappropriate spooler command. Or, JSPOOL may find <Tab>, <Carriage return>, or other characters that cause it to add spaces or give other printer commands that would be inappropriate for graphics or microjustified text applications. To prevent these accidents from occurring, you can issue various "Dumb spooler" commands to turn off part or all of JSPOOL's processing. At the maximum "Dumb spooler" setting, JSPOOL is a "straight wire" print spooler just like other spoolers on the market, stupidly passing data in and out of its buffer without any regard to what the data is.

D0: The default "Dumb spooler" setting. All spooler features are active. All text is scanned for @#$ commands, and all control characters such as <Tab>, <Line feed>, and so on are processed in accordance with the relevant spooler commands such as T (tab settings) and L (line truncate).

D1: All text is scanned for @#$ commands, but control characters such as <Tab> and <Line feed> are not processed. This means that spooler commands that depend on these control characters, such as T (tab settings) and M (margin) will not work. This level of "dumbness" is usually sufficient to permit graphics and microjustification applications to work with JSPOOL.

D2: Control characters such as <Tab> and <Line feed> are processed, but text is not scanned for @#$ commands. Features such as tabs set with command T and margins set with command M will work properly. However, JSPOOL will no longer recognize any further commands sent to it, because all commands are translated internally into the @#$ format. To regain control of JSPOOL after command D2 has been sent, use SPOOLCOM RESET or reboot.

D3: D1 plus D2. Control characters such as <Tab> and <Line feed> are not processed, and text is not scanned for @#$ commands. JSPOOL acts as a simple, "straight wire" spooler, passing data into and out of its buffer without regard to what the data is. This level of "dumbness" allows JSPOOL to handle any kind of graphics or microjustification data without errors. JSPOOL will no longer recognize any further commands sent to it, because all commands are translated internally into the @#$ format. To regain control of JSPOOL after command D3 has been sent, use SPOOLCOM RESET or reboot.

E Even page

With continuous-feed or fan-fold paper, there are two kinds of pages in a stacked printout: pages that face up, and pages that face down. JSPOOL has a feature that lets you ensure that all your printouts start with pages that face up, so you can easily see what each printout is.

At the beginning of each file you want to print, put @#$ E;. This command instructs JSPOOL to print on an "even" (as opposed to "odd") page. If the page in the printer is an even page, JSPOOL will simply start printing the file. If the page is odd, JSPOOL will make the printer eject the page, so that it can start printing the file on an even page. Provided that you have instructed JSPOOL to call the pages that face up "even", every file that starts with @#$ E; will be printed so that the first page faces up.

Before you print your first file, you have to tell JSPOOL whether the current page is even or odd. If the page currently in the printer will face up, set it "even" by typing: A>SPOOLCOM E.

If the page currently in the printer will face down, set it "odd" by typing: A>SPOOLCOM O.

If you manually advance the paper or issue an off-line form feed, use SPOOLCOM to set the current page even or odd as you did before.

H Halt the printer
Default: -H (Printer not halted)

This command stops the printing. You might issue command H from the keyboard if you saw that your printer was about to run out of paper, jam, or ingest your tie. To do this, you would type: A>SPOOLCOM H.

If you put @#$ H; at the beginning of a file, the printer will stop so that you can change to a different form or type of paper. You can also embed @#$ H; in the text of a file to stop the printer and allow you to change the type element or the ribbon. To resume printing, use SPOOLCOM R.

L Line truncate
Default: -L (Lines not truncated)

This command truncates printed lines after a specified number of characters. The excess characters are discarded. The line can be any length from 1 to 255 characters. For example, to truncate all lines at 65 characters, type: A>SPOOLCOM L65.

Note: If you are also using JSPOOL's command M (margin), the margin value must be specified <u>before</u> the line truncate value. This is because the margin value is added to the line truncate value to get the total number of characters allowed per line. Changing M after setting L will change the maximum printed line length. If M plus L exceeds 255, the line truncate command is canceled.

-L Cancel line truncate

This command cancels the line truncate command.

M Margin
Default: -M (No spaces added to margin)

This command adds spaces to the beginning of each line to create the desired left margin. For example, to move the left margin ten spaces to the right from your printer's usual left margin, you would type: A>SPOOLCOM M10.

-M Cancel margin

This command cancels the margin command.

N Null printer
Default: -N (Spooler enabled)

This command makes JSPOOL into a "bit bucket", or null device. Characters are sent to the JSPOOL buffer, but they are never stored there. This command can be useful for testing. To restore JSPOOL, use command -N.

-N Cancel null printer

Restores JSPOOL after the use of command N.

O Odd page

See Even page, above.

P Parameters for serial communications and older printers
Defaults: P8, P2, PN
(8 data bits, 2 stop bits, no parity)

This command sets various parameters for serial communications. Since most printers are oblivious to the different settings of these values, it is unlikely that you will need this command. However, some older printers and some applications using modems may require that these values be correctly set.

Command P is used to set three different parameters for serial communications: the number of data bits per byte (7 or 8); the number of stop bits (1 or 2); and the type of parity used (Even, Odd, or None). For example, to specify 8 data bits, 2 stop bits, and No parity, you would type: A>SPOOLCOM P8 P2 PN.

As with the baud setting, these values must be set <u>before</u> you select a serial port. The default values for these parameters are 9600 baud, 8 data bits, 2 stop bits, and No parity. If you regularly need to set these values, you may want to put them in your CONFIG.SYS file, or issue them as SPOOLCOM commands from a batch file.

R Resume printing
Default: R (Printer not halted)

Restarts the printer after printing has been halted by command H.

S Select a printer
Default: S4 (Parallel port LPT1)

This command tells JSPOOL which printer to use. Printers 0, 1, 2, and 3 are connected to serial ports COM1, COM2, COM3, and COM4. Printers 4, 5, and 6 are connected to parallel ports LPT1, LPT2, and LPT3. The default printer is S4, the parallel printer connected to your first parallel port. If you want to use a different parallel printer, type a command like: A>SPOOLCOM S5.

This command selects the printer connected to your second parallel port. To select a serial port, you must first specify the correct baud rate for your printer, then select the serial printer. For example, you might type: A>SPOOLCOM B6 S0.

This command sets the baud rate to 4800 baud, and then selects the printer connected to your first serial port. If you have only one parallel port, the printer connected to it will be S4. If you have only one serial port, the printer connected to it will be S0. The numbers are assigned in order, so if you have two parallel ports, they will be S4 and S5.

T Tab stops
Default: -T (DOS tabs used)

This command sets tab stops wherever you want them. Most text editors take care of this by inserting the appropriate number of spaces when you press <Tab>. However, if you are printing programs or files created directly with DOS, they may contain <Tab> characters that will mean whatever the currently set tab stops cause them to mean. DOS sets these tab stops every eight spaces (at columns 9, 17, 25, etc.). JSPOOL lets you set them in any positions, from column 1 to column 255, with up to 32 separate tab stops. To set tab stops at columns 3, 10, and 25, for example, you would type: A>SPOOLCOM T3,10,25.

Tab settings must be entered in numerical order. To clear previous tab settings, use SPOOLCOM -T.

-T Cancel tab stops

Reinstates the DOS default tab stops at every eighth column.

W Busy count

You will probably never need this command. Command W is used along with commands Y and Z to optimize the speed of the printer and computer in your particular system. See Section 47 on optimizing printer and computer speed.

Y Time slice

You will probably never need this command. Command Y is used along with commands W and Z to optimize the speed of the printer and computer in your particular system. See Section 47 on optimizing printer and computer speed.

Z Characters per tick

You will probably never need this command. Command Z is used along with commands W and Y to optimize the speed of the printer and computer in your particular system. See Section 47 on optimizing printer and computer speed.

SECTION FORTY-FOUR
ALL ABOUT SPOOLER BUFFERS

44.1: INTRODUCTION TO SPOOLER BUFFERS

This section explains the intricacies of the JSPOOL buffers so that you can choose the best size and type of buffer for each application, and so that you will understand what is happening when JSPOOL does or does not do what you want it to.

The spooler buffer is a segment of RAM memory reserved by JSPOOL to hold data to be printed. JSPOOL intercepts all data being sent to any printer, dumps it into the buffer at electronic speed, and then feeds it out of the buffer to the printer at the printer's slow, mechanical speed. Once the data is all in the buffer, it is printed as far as DOS and your application programs are concerned. You and the computer can go on to do other things, while JSPOOL takes care of sending the data little by little from the buffer to the printer.

As JSPOOL sends data from the buffer to the printer, it gradually empties the buffer. When the last character in the buffer is sent out, the buffer is completely empty again, and the printer stops printing.

If the buffer is not large enough, JSPOOL will not be able to dump all the data into it at once. The first part will go in at high speed, but as soon as the buffer becomes full, the remainder of the data will be able to go into the buffer only at the rate that JSPOOL can make space available by sending characters out of the buffer to the printer. This slows the printing process down to the speed that it used to have before you installed JSPOOL. However, as soon as the last character makes it into the buffer, the computer is free to do something else. JSPOOL will take care of printing the final buffer-full of data without holding up the computer.

To prevent this bottleneck from occurring, make the buffer big enough to hold as much data as you will ever need to put into it. If you never print more than 25K of data at a time, then a 25K buffer (or 26K, just to be safe), will be fine for your system. A larger buffer would just waste memory space, and a smaller buffer would occasionally fill up. If you do not specify the size of buffer you want in your CONFIG.SYS file, JSPOOL will automatically create a buffer that takes up one quarter of the available DOS memory.

JSPOOL gives you the choice of three different types of buffer. The default buffer type is a reserved area of DOS memory, the same part of memory that holds DOS, the program currently running, and the data that the current program is using. This type of buffer is just like the buffers used by other print spoolers, and it takes up space that would otherwise be used by your application programs. To change its size, you have to change CONFIG.SYS and reboot.

If you have installed Tall Tree Systems' JETDRIVE RAM disk package, JSPOOL allows you two other options. First, you can put the spooler buffer on your first RAM disk. This option allows you to change the size of the buffer without rebooting. If you have a JRAM board, this option also means that the buffer does not take up memory space that could otherwise be used by your application programs. You can allocate up to 640K for DOS memory, and still have as large a spooler buffer as you need, hidden away on the RAM disk.

Finally, JSPOOL allows you to define an existing file on the JETDRIVE RAM disk as the buffer. So, instead of reserving an extra area of memory to be the buffer and transferring data into it to be printed, JSPOOL reads directly from the file to the printer. This saves RAM disk space, especially if the file to be printed is large.

44.2: THE INSIDE-DOS BUFFER

Unless you tell it otherwise, JSPOOL will use part of ordinary DOS memory (as opposed to memory used for the RAM disk) for its buffer. And, unless you specify otherwise, this buffer will take up one quarter of the DOS memory available on your machine.

This type and size of buffer will work fine for most applications, but many people will find that it takes up too much of their DOS memory. Reducing available DOS memory has no effect on some application programs, but others may be slowed down, and some spreadsheet and text editing programs may become proportionally more limited in the size of spreadsheet or document they can handle.

As explained in the installation section, to specify the size of the inside-DOS buffer, simply include the desired buffer size in your CONFIG.SYS file. For example, to make a 120K buffer, the JSPOOL.BIN line in your CONFIG.SYS file would be:

```
DEVICE=JSPOOL.BIN 120K
```

44.3: BUFFERS ON THE RAM DISK

44.3.1: INTRODUCTION TO BUFFERS ON THE RAM DISK

If you have installed Tall Tree System's JETDRIVE RAM disk package, JSPOOL gives you two more options for the spooler buffer. Both of these options involve putting the buffer on your first RAM disk, rather than in DOS memory. Putting the buffer on the RAM disk allows you to change the size of the buffer at any time without having to change CONFIG.SYS and reboot. If you have a JRAM board, putting the buffer on the RAM disk also allows you to allocate up to 640K for DOS memory.

You can create two different types of buffers on your first RAM disk. A type 1 buffer works just like the inside-DOS buffer, except that it is located on the RAM disk instead of in DOS memory. A type 2 buffer is actually a file that already exists on the RAM disk. By specifying a type 2 buffer, you can make JSPOOL read directly from a file, without using up space for a separate buffer. All this manipulation of buffers is done with the command MAKEFILE, as explained in the next few sections.

Even when you make a buffer on the RAM disk, JSPOOL still requires that you have an inside-DOS buffer, even if only a very small one. *The inside-DOS buffer is not used when there is a buffer on the RAM disk. Unfortunately, putting the buffer on the RAM disk does not automatically free up the DOS memory space taken up by the inside-DOS buffer. Due to a limitation in DOS, there is no way to reclaim the inside-DOS buffer's reserved memory without changing the CONFIG.SYS file and rebooting. So, if you want to use a buffer on the RAM disk, the best thing to do is to allocate a very small buffer in your CONFIG.SYS file, like:*

```
    DEVICE=JSPOOL.BIN 129
```

44.3.2: MAKING A TYPE 1 BUFFER

Creating a type 1 buffer

You can create a type 1 buffer of any size up to the total size of your first JETDRIVE RAM disk. To create a type 1 buffer of 45K, transfer MAKEFILE.COM to the RAM disk, go to the RAM disk prompt, and type:

```
    C>MAKEFILE 45K/SB
```

This command will create a 45K spooler buffer on the RAM disk (the /SB stands for Spooler Buffer). You will see a message like:

```
Ver 1.05b Ser.#060132
Copyright(c)1983 Tall Tree Systems
JSPOOL.DAT
Creating NEW File.
FILE, Size = 46080 bytes.  First cluster = 541
CONTIGUOUS, Available Size = 46080 bytes.
First sector = 564, No. sectors (size) = 90

JSPOOL Buffer: JSPOOL  DAT
Readonly FILE, Size = 46080 bytes.
First cluster = 541
Type=1, # chrs not yet printed= 0 bytes.
```

The buffer will show in the directory as JSPOOL.DAT. To create a type 1 buffer from a prompt other than that of the first RAM disk, you must specify the RAM disk's drive letter and a name for the buffer, as described below. For your protection, you cannot change the type or size of the buffer unless it is empty. If the buffer is not empty, flush it before using MAKEFILE.

Naming the buffer

If you want to call the spooler buffer something else, or if you want to create a type 1 buffer from a prompt other than that of the first RAM disk, you can specify the RAM disk drive letter and the name of the new buffer with a command like:

```
    A>MAKEFILE C:MYBUFFER.BUF 45K/SB
```

You must specify the drive letter of the RAM disk if it is not the currently logged drive. If you specify a different drive letter, or if you specify no drive letter when the RAM disk is not the currently logged drive, the file will be created on the specified drive, even though it is a physical diskette or hard disk drive, but the file will not become the spooler buffer. The spooler buffer can only be in RAM, either RAM used for DOS memory or RAM used for the RAM disk.

The buffer is a read-only file, so you cannot delete it with DEL, COPY anything to it, or otherwise mess its contents up. Only JSPOOL and MAKEFILE can change the contents, size, or type of the spooler buffer.

Changing the buffer's size
To change the size of the buffer, just repeat the command you used before, specifying a different size buffer. For example, after setting up the named buffer above, you could change its size by typing:

```
    A>MAKEFILE C:MYBUFFER.BUF 100K/SB
```

The usual message will be interrupted by a beep and the prompt:

```
    *ERROR* File already exists, ok to REPLACE?
(Y/N)?
```

Since you want to change the size of the spooler buffer, type Y. You will see a message like:

```
File DELETED. Creating NEW File.
FILE, Size = 102400 bytes. First cluster = 239
CONTIGUOUS, Available Size = 102400 bytes.
First sector = 262, No. sectors (size) = 200
```

You can suppress the beep and the prompt by using /SBD (Spooler Buffer Don't prompt), like:

```
    A>MAKEFILE C:MYBUFFER.BUF 100K/SBD
```

Hiding the buffer
You can make the JSPOOL buffer a hidden file by adding /H to the MAKEFILE command, like:

```
    A>MAKEFILE C:BUFFER 70K/SBH
```

The buffer will not show in the RAM disk's directory, but will still function properly and will be reported correctly by SPOOLCOM.

Deleting the type 1 buffer
You can also delete the spooler buffer entirely, and then create a different one. To delete the spooler buffer, type a command like:

```
    A>MAKEFILE C:MYBUFFER.BUR 0
```

This command will make the buffer zero size, and thus delete it. Again, you will see the prompt:

```
    *ERROR* File already exists, ok to REPLACE?
(Y/N)?
```

Again, you can use /D to suppress the prompt, like:

```
    A>MAKEFILE C:MYBUFFER.BUF 0/D
```

Finally, to return to the inside-DOS spooler buffer, use the command:

```
    A>MAKEFILE/SI
```

The /SI stands for Spooler Inside DOS. This command does not automatically delete the buffer on the RAM disk. If you want to get rid of it, you must delete it as described above.

General notes
For your own protection, you cannot make any of these changes if there is anything still in the current spooler buffer waiting to be printed. If you are willing to throw away the contents of the current spooler buffer in order to change its size, name, or type, first flush the current buffer. You can do this by typing: **A>SPOOLCOM FLUSH** or, if you have installed the JETDRIVE RAM disk package, you can flush the buffer by pressing <Ctrl><Alt><F>. Most simply of all, you can use /F (for Flush) with any MAKEFILE command. For example, to automatically flush the inside-DOS buffer and then create a type 1 buffer on the RAM disk, type: **A>MAKEFILE 78K/SBF** If you do not want to see all these lengthy messages, for example if you are including MAKEFILE in

your AUTOEXEC.BAT file, you can redirect the messages to a non-existent output device by using a command like:

```
    A>MAKEFILE 39K/SBR>NULL
```

44.3.3: MAKING A TYPE 2 BUFFER

44.3.3.1: IS THE FILE CONTIGUOUS?

You can make any file on your first JETDRIVE RAM disk into a type 2 buffer, provided that the file is contiguous. A contiguous file is one that is stored in a continuous, uninterrupted area of RAM disk memory. Files are usually contiguous unless they have been heavily edited since they were transferred to the RAM disk. You can use MAKEFILE both to tell you if a file is contiguous, and to make a contiguous file if it is not.

To determine if the file you want to use as a type 2 buffer is contiguous, use a command like:

```
    C>MAKEFILE MYFILE.TXT
```

If the file is contiguous, you will see a message like:

```
Ver 1.05b Ser.#060132
Copyright(c)1983 Tall Tree Systems
FILE, Size = 46080 bytes. First cluster = 541
CONTIGUOUS, Available Size = 46080 bytes.
First sector = 564, No. sectors (size) = 90
```

If you see this message, the file can be made directly into a type 2 buffer, as explained in the next section. If the file is not contiguous, you will see a message like:

```
Ver 1.05b Ser.#060132
Copyright(c)1983 Tall Tree Systems
FILE, Size = 4992 bytes. First cluster = 322
*ERROR* NON-CONTIGUOUS,
Available Size = 4096 bytes.
First sector = 652, No. sectors (size) = 8
```

If you see this message, you will have to make a contiguous file before setting up a type 2 buffer. See section 43.3.3.3.

44.3.3.2: MAKING A TYPE 2 BUFFER FROM A CONTIGUOUS FILE

Creating a type 2 buffer
To make a contiguous file into a type 2 buffer, use MAKEFILE/S. Continuing the example above, to make MYFILE.TXT into a type 2 buffer, type:

```
    A>MAKEFILE MYFILE.TXT/S
```

You will see a message like:

```
Ver 1.05b Ser.#060132
Copyright(c)1983 Tall Tree Systems
FILE, Size = 46080 bytes. First cluster = 541
CONTIGUOUS, Available Size = 46080 bytes.
First sector = 564, No. sectors (size) = 90

JSPOOL Buffer: MYFILE TXT
Readonly FILE,  Size = 46080 bytes.
First cluster = 541
Type=2, # chrs not yet printed= 41005 bytes.
```

The file will start printing immediately if the currently selected printer is turned on.

Unlike the inside-DOS and type 1 buffers, a type 2 buffer is not "emptied" as it is printed, and it will accept no additional data. This protects the file from being damaged, but it also prevents JSPOOL from printing anything other than the contents of the file. If you try to send additional data to the printer while JSPOOL is set to use a type 2 buffer, the data will not be printed. You can flush a type 2 buffer just like any other buffer. Flushing a type 2 buffer does not harm the file.

After printing a type 2 buffer
When you are done printing the file as a type 2 buffer, you need to return JSPOOL to a type 1 or inside-DOS buffer in order to print any further material. First issue

the command: **A>SPOOLCOM -N**

This command is necessary because JSPOOL automatically issues the N command (Null device) to itself when it starts using a type 2 buffer, in order to prevent new data from being written over the contents of the type 2 buffer. The -N command cancels the Null device setting and allows JSPOOL to put new data into the current buffer. Be sure not to print anything until you have established a new buffer, or you will overwrite the file that you just used as a type 2 buffer.

Now establish a new buffer by issuing the appropriate MAKEFILE command. To return to the inside-DOS buffer, type: **A>MAKEFILE/SI**

To return to a type 1 buffer on the RAM disk, or to create a new one, type a command like:

```
    A>MAKEFILE C:NEWBUF.BUF 75K/SB
```

Or set up a new type 2 buffer as described above.

Restoring the printed file
After being used as a type 2 buffer, the printed file is left as a read- only file. This means that it cannot be edited or deleted. To return the file to its original read/write condition, use MAKEFILE/-R. First, determine the size of the file. Continuing the previous example, you would type:

```
    A>MAKEFILE MYFILE.TXT
```

You would see a message like:

```
Ver 1.05b Ser.#060132
Copyright(c)1983 Tall Tree Systems
Readonly FILE, Size = 46080 bytes.
First cluster = 541
CONTIGUOUS,  Available Size = 46080 bytes.
First sector = 564, No. sectors (size) = 90
```

Then, you would use the size reported in this message to "recreate" MYFILE.TXT as an ordinary read/write file by typing:

```
    A>MAKEFILE MYFILE.TXT 46080/-RD
```

The -R removes the Readonly characteristic and makes the file an ordinary read/write file again. The D suppresses a prompt asking if you really want to change the file.

<u>44.3.3.3: MAKING A CONTIGUOUS FILE</u>

If the file that you want to make into a type 2 buffer is not contiguous, you can use MAKEFILE to make a contiguous file, and then copy the file you want to spool to the new, contiguous file. You can then make this new, contiguous file into a type 2 buffer as explained in section 6.3.3.2. Finally, you can delete the original, non-contiguous file to free some more space on the RAM disk.

You know that your file is not contiguous if you enter a command like:

```
    A>MAKEFILE MYFILE.TXT
```

and you see a message like:

```
Ver 1.05b Ser.#060132
Copyright(c)1983 Tall Tree Systems
FILE,  Size = 4992 bytes. First cluster = 322
*ERROR* NON-CONTIGUOUS,
Available Size = 4096 bytes.
First sector = 652, No. sectors (size) = 8
```

If you see a message like this, you have to make a contiguous file to use as a type 2 buffer. Continuing the example, you would make a contiguous file of the same size with a command like:

```
    A>MAKEFILE TYPE2BUF 4992
```

You will see the usual message describing the new file, which will always be contiguous. Of course, there has to be enough contiguous memory available on the RAM disk. If there is not enough contiguous memory available, you will see the *ERROR* NON-CONTIGUOUS message again, and you will have to make room on the RAM disk by deleting files. Once you have made a new, contiguous file, use COPY to move the data to be printed into the new contiguous file. Continuing the example, you would type:

```
A>COPY MYFILE.TXT TYPE2BUF
```

Now the contents of MYFILE.TXT are also in TYPE2BUF. If you are short on RAM disk space, you can delete the original MYFILE.TXT. Before deleting the file, transfer it to a physical diskette, just to be safe.

Since TYPE2BUF is a contiguous file, you can now make it into a type 2 buffer.

SECTION FORTY-FIVE
OPTIMIZING PRINTER SPEED

45.1: INTRODUCTION TO OPTIMIZING PRINTER SPEED

Your computer can only do a finite number of operations per millisecond. JSPOOL uses a fraction of the computer's time to send characters to your printer, so JSPOOL does slow down whatever else the computer is doing while it is printing. The default settings of JSPOOL allow many printers to print at close to their maximum speed, and some simpler parallel printers to print at about 75% of their maximum speed, while slowing down the computer about 4 to 7%. If your particular printer seems to be printing slower with JSPOOL, or seems to pause between lines, you can adjust JSPOOL to give higher priority to the printer and lower priority to whatever program it is running at the same time. You may be able to achieve some improvement in printer speed without a noticeable loss in computer speed, depending on your particular system.

45.2: THE THREE PRINTING VARIABLES

The PC has an internal clock that ticks 18 times a second. JSPOOL uses a short "slice" of the computer's time immediately after each tick to try to send a few characters to the printer. JSPOOL hands control back to the program in the foreground when the printer is busy or when it uses up its time slice, whichever comes first. The time slice variable is Y. It is specified as a number between 1 and 225, where 1 represents 1/225 of the computer's time, and 225 represents all of the computer's time. The default setting is 64, or approximately 25% of the computer's time when JSPOOL uses the entire time slice.

The variable Y, like the W and Z variables described below, can be set with a SPOOLCOM command, in the CONFIG.SYS file, or with a command embedded in a file, just like any other spooler command.

The spooler variable W controls the "busy count", or how many times JSPOOL will try to send a character to the printer before giving up and handing control back to DOS before the time slice is finished. W prevents JSPOOL from wasting its entire time slice trying to send characters to a busy printer. For a given printer, there is usually a threshold value of W at which it will virtually never give up trying to send characters except when the printer is busy with a time-consuming task like printing a line. Any value much above that will simply waste computer time when the printer is busy, without improving the printing speed. Any value much below the threshold value will drastically reduce printing speed. Because of this threshold effect, it is fairly easy to find the best setting for W. W is set as a number from 1 to 100. The default setting is 11.

The spooler variable Z represents the maximum number of characters that JSPOOL will attempt to send in any given time slice. Once JSPOOL has sent Z characters, it automatically gives control back to DOS or the current application program. This prevents JSPOOL from wasting time trying to send additional characters that the printer will not be able to accept. If Z is too low, JSPOOL will quit sending characters too early, and will slow down the printer. If Z is too large, JSPOOL will consistently try to send more characters than the printer will accept, and will thus waste up to a whole busy-count's worth of computer time at every clock tick. Like W, the optimum setting for Z is easy to find because it is a threshold value. Any setting below the optimum will noticeably slow the printer, while settings above the optimum will have virtually no effect on printer speed, but will slow down the computer. Z is specified as a number from 1 to 100. The default setting of Z is 12.

45.3: OPTIMIZING Y, W, AND Z

The best settings of Y, W, and Z will depend on your particular printer and your own preferences for printer and computer speed. JSPOOL's default settings are pretty good for most systems, but by going through the procedure below, you may be able to increase your printer's speed without a noticeable loss in computer speed. In particular, if you have a parallel printer that appears to pause between lines longer when you use JSPOOL, you should be able to reduce the pause by following the procedure below.

First, temporarily set W and Z to 100 (a very high value for these variables). Do this by typing:

```
A>SPOOLCOM W100 Z100
```

Now, increase Y until the printer's speed is acceptable. The best way to judge this is to make a file at least thirty long lines and time how long it takes to print the file for different values of Y.

The default setting of Y is 64, which amounts to a slice of about 25% of the computer's time. Remember, because JSPOOL will usually give up trying to send characters to the printer before its time is up, the actual computer time used up will be substantially less. We do not recommend increasing Y above 200, because the keyboard and monitor response will become sluggish, and you may occasionally lose keystrokes if you are typing in commands while printing.

In general, the larger you make Y, the faster the printer will print, and the slower the computer will run. But because the whole time slice is only rarely used, even a Y setting of 200 will typically slow the computer down only about 10%.

Once you have selected the smallest acceptable value of Y, reduce W to just above the value at which the printing speed suddenly starts to drop. This value is typically around 7 to 12, but the threshold value may be different for your particular printer.

Finally, reduce Z to just above the value at which the printing speed suddenly starts to drop. This value is typically around 7 to 14, but again, the threshold value will depend on your particular printer.

Once you have determined the settings of Y, W, and Z that best satisfy your needs for printer and computer speed, you will probably want to set them automatically every time you boot up. You can do this by specifying the values in your CONFIG.SYS file. For example, you might use:

```
A>JSPOOL.BIN 73K /@#$i Y125 W7 Z10
```

If you have more than one printer, you might make a little batch file for each printer that would automatically set the optimal settings for the printer and then select it. For a serial printer, the batch file might look like:

```
A>COPY CON SERIAL.BAT
SPOOLCOM B5 S0 Y100 W9 Z13
<F6>
```

To use your serial printer, you would simply type:

```
A>SERIAL
```

PROGRAM LICENSE

You should read carefully the following terms and conditions before opening this diskette package. Opening this diskette package indicates your acceptance of these terms and conditions. If you do not agree with them, you should promptly return the package unopened; your money will be refunded.

You may:

A. use the JETDRIVE and JSPOOL programs on a single machine.

B. use and copy the JBOOT.BIN, JDRIVE.BIN, MAKEDOS.COM, JET.COM, DUMDISK.BIN, JBOOT.ASM, JDRIVE.ASM, JSPOOL.BIN, SPOOLCOM.COM, MAKEFILE.COM, JPAGER.BIN, JPAGER.ASM and JSPOOL.ASM programs as many times on as many diskettes as desired for use on a single machine.

You may not use, copy, modify, or transfer these programs, or any copied, modified, or merged portion, in whole or in part, except as expressly provided for in this license.

If you transfer possession of any copy, modification, or merged portion of these programs to another party, your license is automatically terminated. You agree, upon such termination, to destroy the program together with all copies, modifications, and merged portions in any form.

PROGRAM LIMITED WARRANTY

These programs are provided "as is" without warranty of any kind, either expressed or implied, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the programs is with you. Should the programs be defective, you (not Tall Tree Systems) assume the entire cost of all necessary servicing, repair, or correction. In no event will Tall Tree Systems be liable to you for any damages, including any lost profits, lost savings, or other incidental or consequential damages. Some states do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

Tall Tree Systems does not warrant that the functions contained in these programs will meet your requirements or that the operation of the programs will be uninterrupted or error free. However, Tall Tree Systems warrants the diskettes on which the program is furnished to be free from defects in materials and workmanship, and will replace defective diskettes free of charge.