JRAM

USER'S MANUAL

TALL TREE SYSTEMS
1120 San Antonio Road, Palo Alto, CA 94303

TALL TREE SYSTEMS

Tall Tree Systems, the brainchild of John Henderson, is a growing company that creates and manufactures both hardware and software for use on the IBM PC, XT, AT and other compatible machines. We focus solely on computers compatible with the IBM Personal Computer and are widely respected as specialists in this field.

Tall Tree Systems welcomes you to the world of high speed computing and megamemory using RAM disks and/or expanded memory. We know that your concern is for efficient, effective computing whether you work in your home, in a small business, or in a large corporation. Every product we make builds upon the last, creating a system with unmatched speed, storage capacity, and flexibility.

HARDWARE: We feature JRAM multifunction memory boards. The JRAM boards couple all of the most popular I/O functions - serial ports, parallel ports, and a clock/calendar - with Tall Tree's added dimensions - big memory and first class software. Using both the 64K and the 256K chips, the JRAM has a memory capacity ranging from 512K to 2 Megabytes - in one slot! The JRAM is also a modular multifunction board. You can purchase only those options that you need.

The JDISKETTE controller, which is available either as a module or a stand-alone board, allows you to put 1.2 megabyte diskette drives in the IBM PC or XT for the ultimate in diskette storage and IBM AT compatibility. We also have the JRAM-AT board, a multifunction memory board specifically designed for the IBM AT.

SOFTWARE: We have created several software products designed to expand your computer's memory and keep apace of the advancing technology while allowing you to switch memory size easily between the operating system and the RAM disks. In order to meet the changing needs of our users and allow them to customize their systems we keep adding new features to our software:

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) and high density diskette drives in a PC or compatible. Put 800 K, 1.2 Mb, or more on one diskette.

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

One or more of these installable device drivers will work with your system and can be used with local area networks. We include the source code for all of our device drivers to enable you to tailor them to meet your specific needs.

This manual was written by Janet Bein, Bruce Owen, and John Swen.

JRAM-3 ERRATA SHEET

B11150

January 1986

11.2: CHANGE OF ADDRESS INFORMATION

The JRAM-3 board can be set to either the E or F address page in your computer. We are currently shipping our boards set to the F address page, because that leaves page E available as a swapping page for JPAGER.BIN or JLASER.BIN in case you want to one of these drivers as well as a RAM disk.

Our previous JRAM boards were shipped set to page E, and our software recognizes page F as the address for board 5. If you leave the board set as it is shipped to the F page, for example, MEMTEST will identify this board as board #5. There is an error in Table 11.2.1, which appears on page 11-5 of your manual. Please replace it with the following table.

11.2.1: SETTING THE SWITCHES ON THE JRAM-3

The six switches on the top right corner of the JRAM-3 board assign a unique set of bankswitching control addresses to the board for use by JETDRIVE, JSPOOL, multitasking software, and other packages using the Tall Tree Systems bankswitching standard. Every JRAM, JRAM-2, and JRAM-3 board in the computer must be identified by its own distinct addresses.

If you have more than one JRAM board, set the switches on the boards as indicated in Table 11.2.1a. If you have both JRAM-3 and earlier JRAM boards in your system, the JRAM-3 boards must come after the earlier boards. For example, you might set your JRAM-2 as card 5 and your JRAM-3 as card 6.

TABLE 11.2.1a
JRAM-3 SWITCH POSITIONS

CARD #	SW 1	SW 2	SW 3	SW 4	SW 5	SW 6	BANK SELECT REGISTER MEMORY ADDRESS
1	ON	-	-	-	ON	-	E800
2	-	ON	-	-	ON	-	E400
3	-	-	ON	-	ON	-	E200
4	-	-	-	ON	ON	-	E100
5	ON	-	-	-	-	ON	F800
6	-	ON	-	-	-	ON	F400
7	-	-	ON	-	-	ON	F200
8	-	-	-	ON	-	ON	F100

CHANGING THE DEFAULT PAGE FOR THE SOFTWARE WITH /S=

Tall Tree Systems device drivers default to page 13 (D hex) of your computer. If you have a Tall Grass hard disk or some other conflict with page 13, you may change the default by using the /S=<#>b option in your device line for JBOOT.BIN (See Section 35 Advanced Features). The device driver for the RAM disk, JDRIVE.BIN can share the same address page as JBOOT.BIN and it will move automatically with JBOOT.BIN when you use the /S= option. If you do not have a Tecmar graphics card or an EGA card and if your DOS size is no more than 640K, you can probably use page 10 (A hex). For example,

```
DEVICE=JBOOT.BIN 640K /S=10B
DEVICE=JDRIVE.BIN
```

Remember, if you plan to use both a RAM disk and JLASER.BIN or JPAGER.BIN, you must also assign a separate swapping page for that driver. JLASER.BIN or JPAGER.BIN cannot share the same address as JDRIVE.BIN.

If you have your JRAM-3 switches set to page F (1 and 6 ON), you can use page 14 (E hex) for JPAGER.BIN or JLASER.BIN. For example,

```
DEVICE=JBOOT.BIN 640K /S=10B
DEVICE=JDRIVE.BIN 256K
DEVICE=JLASER.BIN /S=14B
```

JSPOOL ON A RAM DISK
Moving JSPOOL to another Address Page

When JSPOOL is put on a RAM disk, its driver uses page 10 (A hex). If you have a Tecmar graphics card or some other conflict with page 10, you may change the address by using DEBUG to patch JSPOOL.

First, choose an address page that is not used by any hardware or software in your system. Please refer to the following memory address map.

1024 ~~960~~	15(F)	JRAM/ROM*
960 ~~896~~	14(E)	JRAM/ROM in AT*
896 ~~832~~	13(D)	Tall Tree software default JBOOT.BIN & JDRIVE.BIN JLASER.BIN
832	12(C)	hard disk ROM
768	11(B)	monitor
704	10(A)	Either 704K DOS or JSPOOL on RAM disk or JDRIVE.BIN or JLASER.BIN
640	9	available for swapping if you reduce DOS size to 576K
576	8	available for swapping if you reduce DOS size to 512K
512	7	not available for swapping with 512K motherboard
448	6	not available for swapping with 512K motherboard
384	5	not available for swapping with 512K motherboard
320	4	not available for swapping with 512K motherboard
256	3	NEVER AVAILABLE FOR SWAPPING / MOTHERBOARD
192	2	NEVER AVAILABLE FOR SWAPPING / MOTHERBOARD
128	1	NEVER AVAILABLE FOR SWAPPING / MOTHERBOARD
64	0	NEVER AVAILABLE FOR SWAPPING / MOTHERBOARD

***NOTE: For information on how JRAM boards can be set to pages also used by ROM, see Section 14 "How JRAM Boards Work".**

Each page represents a 64K block of address. Blocks 0-9 are the 640K usually available for DOS memory. Blocks 11 and 15 are never available. Block 12 is not available in XTs and ATs. Blocks 10-15 are not available with the JRAM-AT board.

All JRAM boards can be set to either block 14 or 15. If you have a PC or an XT, you can set the board to page 15, freeing page 14 as a possible swap page.

Change JSPOOL to an available block with the following patch. (Use the hexadecimal notation for numbers above 9, 10=A, 12=C, 14=E.)

Make sure that the DOS program DEBUG.COM and a copy of JSPOOL.BIN are both on the currently logged drive. Type in what is shown in italics, followed by a carriage return each time. The rest will be displayed by the computer. This example shows JSPOOL being changed to use block 8. Copy this new version of JSPOOL.BIN to the root directory of your boot disk and reboot.

```
A>DEBUG JSPOOL.BIN
-E1AA
-OA.08
-W
Writing XXXX BYTES
-Q
```

JRAM-3
TABLE OF CONTENTS

B11150

PAGE

INTRODUCTION

THE JRAM-3

PIGGYBACK MODULES (provided separately for module orders)

PAGE

JETDRIVE

SECTION 30: INTRODUCTION TO THE COMBO DISK

SECTION 31: INSTALLING JETDRIVE

SECTION 32: USING THE RAM DISKS

SECTION 33: USING JET

SECTION 34: JET OPTIONS

SECTION 35: ADVANCED FEATURES

SECTION ONE
HOW TO USE THIS MANUAL

1.1:"MODULAR" MANUALS

In keeping with our hardware's modular flexibility, we have adopted a system of "modular" manuals. Each manual is made up of numbered sections, which are independently paginated. This allows us to update manuals more frequently. Please note that we have purposely skipped numbers to allow for possible additions to the major parts or divisions of the manual.

Information regarding the board itself begins in section 10. If you have any piggyback modules, you will find the necessary information beginning in section 20. If you do not have a piggyback module, the section will not be included. Information regarding the JETDRIVE software package begins in section 30; the JSPOOL software package begins in section 40.

YOU DO NOT HAVE TO READ THIS MANUAL FROM COVER TO COVER! You can simply refer to those sections that apply to the hardware and/or program(s) you are using. Many of you will be able to manage just by reading the introductory and installation sections of the parts that apply to the program(s) that you are using.

If you are installing your own chips or if you need information on a particular program, such as JET, please refer to the Table of Contents for the appropriate section numbers and pages.

SECTION TWO
ALLOCATING MEMORY

2.1:USES OF MEMORY

The JRAM board and its accompanying software allows you to use the amount of RAM you have installed in your machine in several different ways.

You must allocate memory by typing the various installable device drivers into a CONFIG.SYS file. When the computer boots up, it will check the CONFIG.SYS file and automatically install any drivers you have listed, allocating the amount of memory you have requested for each of these drivers.

Your computer's memory can be allocated to the following:
1) DOS
2) JETDRIVE's electronic disk
3) JSPOOL's spooler buffer
4) JPAGER'S expanded memory

The **amount** of memory that you allocate depends on which software programs you are using and what you are using them for. A large spreadsheet, for example, will require more memory than a smaller spreadsheet. A suggested amount of memory to be allocated to DOS and/or JPAGER will probably appear in the manual for your software application program. The program should notify you with an error message if there is not enough memory available for your application. In that case, simply adjust your CONFIG.SYS file accordingly (assuming that you have installed a sufficient amount of RAM).

You will find detailed information on installing Tall Tree System's software packages and setting up a CONFIG.SYS file in sections 30, 31, 40, 41, 60 and 61 of this manual. We advise you to look at all these sections before you proceed. If you intend to use only one of Tall Tree System's programs, you can simply read the section that applies to your particular needs: 30 and 31 for JETDRIVE; 40 AND 41 for JSPOOL; 60 AND 61 for JPAGER. If you intend to use a combination of these programs, you will find much of the information is repetitive. You may not be able to run all programs together with the same system.

THE JRAM BOARD

chl 850965

```
22880120            7270  44554
TALL TREE SYSTEMS                       JRAM-3 SERIAL
JRAM 3 SERIAL W/ 0K UP TO 2MB, JPAGER, JSPOOL
 -), JETDRIVE         IBM PC            DISKETTE
MODULE                MEMORY
```

SECTION TEN
INTRODUCTION TO THE JRAM-3

10.1: WHAT IS JRAM-3 ?

JRAM-3 is the fourth generation of Tall Tree Systems memory boards for IBM personal computers and compatible machines. JRAM-3 incorporates the traditional Tall Tree features: enormous memory and first class software for RAM disks and spoolers. The JRAM-3 was developed to fit the Lotus/Intel/Microsoft specification for expanded memory. It allows you to access up to eight megabytes of memory for *expanded memory specification (EMS)* software programs.

10.2: COMPATIBILITY

JRAM-3 works with the IBM PC, XT, and many compatible machines. Generally, if the machine is running PC DOS 2.0 or above, or MS DOS 2.0 or above, it can use JRAM-3 with the JRAM Combo software package.

10.3: MEGAMEMORY

The IBM PC, XT, and compatibles used to be limited to 640K of user memory. Tall Tree's JRAM boards broke through this electronic memory barrier by utilizing "**bank-switched memory**", which lets you expand your workspace to up to 704K of DOS memory, plus a super-fast RAM disk of up to **sixteen megabytes**. The JRAM-3 has added yet another memory multiplying device, "**page switching**", to give you up to **eight megabytes** of **expanded memory**. This expanded memory can be utilized by *EMS* software application programs.

A single JRAM-3 board, fully populated with 256K chips, holds **two megabytes** of RAM. For real memory power, you can put up to eight JRAM boards in one machine for a total of **sixteen megabytes**.

10.4: MODULAR MULTIFUNCTIONS

Like our other multifunction boards, JRAM-3 features I/O modular capabilities. Although these modules are not included in the initial release of JRAM-3s, the boards are built with the add-on capability. The following modules are available:

1) Clock/calendar module
2) Serial port, parallel port, clock/calendar module
3) Serial port, serial port module
4) Serial port, serial port, clock/calendar module
5) Diskette controller

NOTE: Although the port connectors are on the JRAM board, the ports cannot be enabled without the appropriate JRAM piggyback module. These modules are smaller boards that fit into the perpendicular sockets on the JRAM board.

10.5: POWERFUL SYSTEMS SOFTWARE

JRAM-3 comes with two bank-switching device drivers. JBOOT.BIN initializes the board and handles JRAM bank-switching in 64K blocks. JPAGER.BIN manages the 16K "page-switching" operation for the expanded memory specification. JRAM-3 also includes JETDRIVE and JSPOOL, the best RAM disk and printspooling software you can buy. JETDRIVE creates the only RAM disk that lets you reboot (<ctrl-alt-del>) without losing the disk's contents. JETDRIVE also includes JET, a high-speed file transfer utility that is two to fifteen times faster than the DOS COPY command, and five times faster than BACKUP and RESTORE.

JSPOOL is a print spooler that lets you continue working with your computer while it prints out files. JSPOOL lets you select from seven different printers, place its buffer on RAM disk to free DOS memory, and control twenty-two printer and spooler features to maximize your computer's efficiency.

10.6: TECHNICAL SPECIFICATIONS

Maximum supply currents	+5 volts	+12 volts	-12 volts
JRAM-3, 512K or 2Mb, no I/O	600 ma	-	-
Serial-Parallel-Clock Module	330 ma	13 ma	20 ma
Serial-Serial-Clock Module	300 ma	24 ma	42 ma
Clock Module	82 ma	-	-
Maximum total current possible	930 ma	24 ma	42 ma

Maximum access time	275 nanoseconds
Board length	13.3 inches
Board height w/o edge terminals	3.9 inches
Board height with edge terminals	4.2 inches

RAM required: 64K or 256K chips, not slower than 250 nanoseconds maximum access time.

SECTION ELEVEN
INSTALLING THE JRAM-3 BOARD

11.1: BASIC INSTALLATION

INSTALLATION CHECK-LIST

I) Prepare computer.
A) Check motherboard RAM switches.
B) Insert plastic card guide.

II) Check JRAM-3 switches.
Switch one and switch six should be ON
if this is your first board.

III)Check shunts P2, P3, P4.
Should be set to zero
if this is your first board.

IV) Set shunts on piggyback module
if you have one.

V) Insert JRAM-3.

VI) Close up and plug in.

VII)Run MEMTEST diagnostic.

VIII) Install JRAM software.

I) Prepare computer.
Turn off the computer, remove the cover, and remove the metal piece covering the opening in the back of the computer opposite any unfilled expansion slot. Save the screw that held the expansion slot cover in place.

A) Make sure that the motherboard RAM switches are set to show the amount of non-JRAM memory on your computer's motherboard and on any other memory boards in your system. (Please refer to your computer's manual for details on the correct setting).

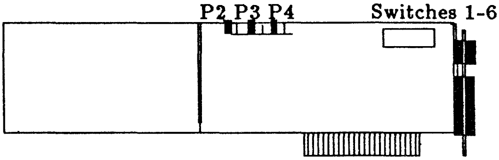
NOTE: The motherboard RAM sockets must be filled unless you have a COMPAQ portable.

***If you are simply adding a JRAM-3 to the boards already in your machine, you do not need to change the motherboard memory switches.**

*If you are removing an ordinary memory board, you must reset the motherboard memory switches to show the reduced amount of non-JRAM memory installed.

B) Snap the supplied plastic card guide into the two small holes in the front of the PC chassis opposite the desired expansion slot.

The JRAM-3



II) Check JRAM-3 switches. SW1 and SW6 should be ON. The six switches on the top right corner of the JRAM-3 board assign a unique set of bankswitching control addresses to the board for use by JETDRIVE, JSPOOL, multitasking software, and other packages using the Tall Tree Systems bankswitching standard. Every JRAM, JRAM-2, and JRAM-3 board in the computer must be identified by its own distinct addresses.

The JRAM-3 board can be set to either the E or F address page in your computer. We are currently shipping our boards set to the F address page, because that leaves page E available as a swapping page in case there is a conflict between the default page for our software (page 13, D hex) and some other peripheral that you have added to your system.

Our previous JRAM boards were shipped set to page E, and our software recognizes page F as the address for board 5. If you leave the board set as it is shipped to the F page, for example, MEMTEST will identify this board as board #5. This will not interfere with the board's functioning in any way. If you would like to change the board select, please refer to table 11.2.1 for alternative switch positions.

III) Check Shunts P2, P3, and P4 set to zero. The expanded memory specification requires that each board must have a unique set of control addresses different from those used by the Tall Tree Systems' standard. These addresses are assigned by the three black shunts on the top edge of the JRAM-3 board. If this is not your first board, please refer to section 11.2.2 for the correct shunt settings.

IV) Set the shunts on the piggyback module if you have one. See section 20 and the section for your particular module.

V) Insert JRAM-3. Slide the JRAM-3 board into the desired expansion slot. Make sure that you plug the edge connector (the gold-plated section that projects from the bottom of the board) firmly into the expansion slot connector, and that the board is well seated in the plastic card guide on the front of the chassis.

Attach the JRAM-3's bracket to the back of the PC with the screw that held the expansion slot cover in place.

VI) Close up and plug in. Replace the cover of the PC. Plug in your computer.

VII) Run the MEMTEST memory diagnostic to check your installation, particularly if you have installed your own RAM chips.

Please refer to Section 12 for detailed instructions.

VIII)Install JRAM software.
Boot up your machine and install the JRAM software as described in the software sections of this manual.

* You must install JBOOT.BIN to use JRAM-3 memory.

**If you intend to use expanded memory, you must install both JBOOT.BIN and JPAGER.BIN. See Section 61 for information on how to install JBOOT.BIN and JPAGER.BIN.

11.2: INSTALLING MORE THAN ONE JRAM-3 BOARD

11.2.1: SETTING THE SWITCHES ON THE JRAM-3

I) Prepare computer.
Turn off the computer, remove the cover, and remove the metal piece covering the opening in the back of the computer opposite any unfilled expansion slot. Save the screw that held the expansion slot cover in place.

A)***If you are simply adding a JRAM-3 to the boards already in your machine, you do not need to change the motherboard memory switches.**

B) Snap the supplied plastic card guide into the two small holes in the front of the PC chassis opposite the desired expansion slot.

II) Set JRAM-3 switches.
The JRAM-3 board can be set to either the E or F address page in your computer. We are currently shipping our boards set to the F address page, because that leaves page E available as a swapping page in case there is a conflict between the default page for our software (page 13, D hex) and some other peripheral that you have added to your system.

If you leave the board set as it is shipped to the F page, for example, MEMTEST will identify this board as board #5. This will not interfere with the board's functioning in any way. If you would like to change the board select, please refer to table 11.2.1 for alternative switch positions. We suggest that you leave your first board set as card 5, and set your second board as card 6, etc.

If you have both JRAM-3 and the earlier JRAM boards in your system, the JRAM-3 boards must have higher numbers than the earlier boards.

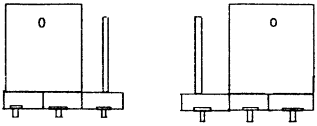
TABLE 11.2.1
JRAM-3 SWITCH POSITIONS

CARD #	SW 1	SW 2	SW 3	SW 4	SW 5	SW 6	BANK SELECT REGISTER MEMORY ADDRESS
1	ON	-	-	-	ON	~~ON~~	E~~F~~800
2	-	ON	-	-	ON	~~ON~~	E~~F~~400
3	-	-	ON	-	ON	~~ON~~	E~~F~~200
4	-	-	-	ON	ON	~~ON~~	E~~F~~100
5	ON	-	-	-	~~ON~~	ON	F~~E~~800
6	-	ON	-	-	~~ON~~	ON	F~~E~~400
7	-	-	ON	-	~~ON~~	ON	F~~E~~200
8	-	-	-	ON	~~ON~~	ON	F~~E~~100

IV) Set Shunts P2, P3, and P4.
The *EMS* standard also requires each board to have a unique set of control addresses, different from those used by the Tall Tree Systems standard. These addresses are assigned by the three black shunts on the top edge of the JRAM-3 board.

The shunts are small black plastic hoods that contain metal contacts. Each shunt connects two of three metal pins. The shunt slips onto either the left pair or the right pair of either set of pins, selecting one option or another. To move the shunt, stick the end of a bent paper clip in the small hole and slide it off one pair of pins, then slide it onto another. The shunts are labelled, from left to right, P2, P3, and P4, but the labels are hidden until you slip the black hoods off. When the shunt connects the left-hand pair of pins, it is in position 0; when it connects the right-hand pair of pins, it is in position 1. These positions are marked on the board.

Figure 11.2.1: HOW TO SET SHUNTS



Position "0"   Position "1"

Set the shunts on your JRAM-3 boards as indicated in table 11.2.2. *Note that the shunt settings must correspond to the switch settings.* For example, if you set the switches to indicate card 1 or 5, then you must set the shunts to indicate card 1 or 5. The expanded memory specification allows only four boards, so you can have only one board with the shunts set in the 1 or 5 position, only one in the 2 or 6 position, and so on. You can disable the *EMS* by removing the shunt from P2 altogether. (We suggest slipping it on a single pin for possible future use.)

If your system also includes JRAM or JRAM-2 boards, you can use the full four JRAM-3 boards allowed by the *EMS* by setting the JRAM-3 boards as cards 5 through 8. Skipping card numbers is permissible, but JRAM-3 boards must have higher card numbers than JRAM-2 boards.

TABLE 11.2.2

JRAM-3 SHUNT POSITIONS

CARD #	P2	P3	P4	Write	Read and Enable Write
1 or 5	0	0	0	220-23F	220-227
2 or 6	0	0	1	220-23F	228-22F
3 or 7	0	1	0	220-23F	230-237
4 or 8	0	0	1	220-23F	238-23F

In the unlikely event that these I/O addresses conflict with other hardware in your system, three other sets of I/O addresses are available as shown below in table 11.2.3. To use these alternate I/O control addresses, you must specify in the JPAGER.BIN line in CONFIG.SYS which set you have selected. See section 61.1 for details.

TABLE 11.2.3

ALTERNATE JRAM-3 SHUNT POSITIONS

SET 2

CARD #	JMP1	JMP2	P2	P3	P4	Write	Read and Enable Write
1 or 5	-	-	1	0	0	280-29F	280-287
2 or 6	-	-	1	0	1	280-29F	288-28F
3 or 7	-	-	1	1	0	280-29F	290-297
4 or 8	-	-	1	1	1	280-29F	298-29F

SET 3

1 or 5	Wire	-	-	0	0	240-25F	240-247
2 or 6	Wire	-	-	0	1	240-25F	248-24F
3 or 7	Wire	-	-	1	0	240-25F	250-257
4 or 8	Wire	-	-	1	1	240-25F	258-25F

SET 4

1 or 5	-	Wire	-	0	0	2A0-2BF	2A0-2A7
2 or 6	-	Wire	-	0	1	2A0-2BF	2A8-2AF
3 or 7	-	Wire	-	1	0	2A0-2BF	2B0-2B7
4 or 8	-	Wire	-	1	1	2A0-2BF	2B8-2BF

Wire = wire installed - = no wire or shunt installed

IV) Set the shunts on your piggyback module if you have one.

V) Insert the JRAM-3.

VI) Close up and plug in the computer.

VII)Run MEMTEST from a standard system diskette.

VIII) Install JRAM software if you have not already done so.

11.3:INSTALLING OR ADDING YOUR OWN RAM CHIPS

You may order the board unpopulated or only partially filled, and add RAM chips later, as your needs grow. You can order or install any number of banks of 64K or 256K chips, depending on your needs and budget. You can even start with 64K chips and replace them, bank by bank, with 256K chips as your needs grow and the 256K chips get cheaper. But be aware of this: when you buy a populated JRAM board, all the RAM is thoroughly tested before the board is shipped out. We have found that many memory related problems occur due to improper insertion of RAM chips or a bad batch of RAM.

If you do buy your own RAM, we recommend the plastic chips, which are cheaper than ceramic chips and more compatible with our boards. We also have found best results when the board is populated with newest possible chips of the same date code. The date code is printed on each chip by year and week. For example, 8525XXX signifies that the chip was manufactured in the 25th week of 1985. If you wish, you may contact our Customer Service Department for some chip brands that we are currently using in our boards.

If you have found an improperly inserted chip, you should just pull out the offending chip, straighten any bent pins, and re-insert it properly. Be careful, especially if you have never inserted chips before, and you can do it right the first time.

11.3.1: FILLING ONLY PART OF THE BOARD

Each vertical column of nine sockets on the JRAM board holds one bank of chips. A bank of nine 64K **bit** chips holds 64K **bytes** of data. A bank of 256K **bit** chips holds 256K **bytes** of data. (Each byte of memory is made up of one bit on each of the nine chips.) You do not have to fill the whole board, but you do need to fill whole banks.

YOU SHOULD FILL A WHOLE BANK AT A TIME, BEGINNING WITH THE VERTICAL COLUMN FURTHEST TO THE RIGHT,i.e., closest to the logic section of the board. You can mix banks of 64K and 256K RAM chips on a single board or on several boards. You can **not** mix chips **within** a single bank. For best results, you should group chips with the same brand and nanosecond rate. Stuff your 256K chips first, followed by the fastest of your 64K chips (i.e., stuff in order of sensitivity). We do not recommend using more than two different chip speeds, although it will probably work. The fastest chip you can use on the current rev board is 120 nanoseconds, and the slowest is 275 nanoseconds.

11.3.2: INSERTING RAM CHIPS

Each RAM chip MUST be inserted so that the small semi-circular notch on one end of the chip is over the small semi-circular groove on one end of the socket. This groove is marked in white on the JRAM board itself, to remove all possible doubt.

Insert the chips gently, by lightly resting the row of pins farther away from you in the socket and carefully pressing the chip towards them until the row of pins closer to you can set down onto the contacts. Then push the chip straight down and seat it firmly in the socket.

After inserting your RAM chips, go back and inspect each one for pins bent under the chip or outside the socket. If you find a bent pin, pull the chip out by gently working a small screwdriver under it, straighten the pin, and carefully reinsert it correctly.

RAM chips are MOS devices, and like all MOS devices, they are subject to damage by static electrical discharges. You should always assume that your body is carrying a static charge, and carefully discharge yourself by firmly touching a grounded metal surface such as the chassis of the PC before handling your RAM chips. Static is more of a problem in lower humidity. Walking on a carpet or wearing wool or some synthetic fabrics can build up a dangerous static charge on your body.

11.4: USING JRAM MEMORY WITH A NON-DOS OPERATING SYSTEM

JRAM memory is not the same as ordinary memory. It must be initialized by JBOOT.BIN (part of the JETDRIVE software package) or a similar program before it can be used. JBOOT.BIN runs only under PC-DOS and MS-DOS, but you can still use it to initialize JRAM memory for use by a different operating system, such as UNIX, CP/M, P-system, and so on.

To use a different operating system, first boot up with a DOS boot disk as described in the JRAM Software Manual in the section on JETDRIVE installation. The CONFIG.SYS file on this boot disk should allocate as much memory to DOS as possible, up to 704K. A typical CONFIG.SYS file for this purpose looks like this:

```
A>COPY CON CONFIG.SYS
DEVICE=JBOOT.BIN 704K
<F6>
```

After booting up with JBOOT.BIN and a CONFIG.SYS file like the one above, put your alternate operating system boot disk in drive A: and warm reboot by pressing <Ctrl><Alt><Del>. When the new operating system comes up, all the memory you allocated to DOS, including JRAM memory, will be available for it to use exactly like ordinary memory. The JRAM memory will only be turned off if you turn the computer off or write data to address pages E or F (depending on the page select setting on the JRAM board). A warm reboot will have no unusual effect.

To use JRAM memory beyond 704K with a different operating system, you will have to write a driver for that particular operating system.

SECTION TWELVE
TESTING INSTALLATION AND RAM

12:1 MEMTEST

All JRAM boards are tested and "burned in" for at least eight hours. If you have purchased a populated board from Tall Tree, it is very unlikely that you would have any memory problems. However, if you have installed your own RAM, or if you ever suspect a memory problem, you should test your JRAM memory by running MEMTEST. (MEMTEST replaces JRCK, a previous diagnostic program). MEMTEST tests both 64K and 256K chips on the same board at the same time, and takes approximately a half hour per half megabyte of RAM.

First, save everything from your RAM disks onto a physical diskette, because MEMTEST destroys everything in memory. Copy the MEMTEST.COM file from your Tall Tree Systems distribution diskette onto a system diskette if you have not already done so. Put your distribution diskette safely away. You should boot up the computer with a regular DOS diskette, that is, without any extra drivers installed in CONFIG.SYS.

Put the copy of your system diskette (with MEMTEST.COM) into drive A: and type:

```
A>MEMTEST<ENTER>
```

The program will instruct you to enter the necessary information as follows:

Enter # of Banks of Non-JRAM Memory (1-9):

This refers to the memory that you have on your motherboard plus the memory you might have on any other non-JRAM expansion card. A bank is 64K bytes. If you have 640K or more of non-JRAM memory, use the hex equivalents of the numbers. For example,

Total Amount Non-JRAM Memory	<ENTER>
256	4
512	8
640	A

Enter swapping page 9=640K A=704K D=896K

The PC's memory is divided into "pages " or "segments" of 64K each. There are 16 pages, beginning with page 0. Pages 10-15 are usually represented by the hex notation A - F. Memtest can use the swapping pages mentioned above. The swapping page **addresses** are given in K figures: 640K refers to the first address on page 9; 704K refers to the first address on page A (the tenth page); and 896K refers to the first address on page D (the thirteenth page).

Most users should select swapping page <D>.

Of the sixteen pages of memory, the first ten, 0-9, are generally used for DOS. Page B is dedicated to the IBM video display. Pages E and F are both reserved for ROM and are shared by the JRAM bank select registers. If you wish, you may consult the source code on your disk to determine a different swapping page to suit your particular needs.

Enter type 0=JRAM, 1=JRAM-2A, 2=JRAM-2B
or JRAM-3, 3=JRAM-AT

This refers to the particular model of JRAM board that you are using. You can identify which model you have as follows. Hold the JRAM board so that the gold fingered connector is on the right hand side bottom edge.

Directly above the gold fingered connector(s) is the logic section of the board. To the left of the logic section, you will see the RAM banks.

If your board has three large horizontal rows, you have the original JRAM board, so you should enter <0>.

The JRAM-2 and following models look totally different from the original JRAM. They have eight vertical columns of RAM. The way to distinguish the JRAM-2A from the JRAM-2B is by the position of the page and board select shunts.

If the page and board select shunts are on the upper right hand edge of your board, you have a JRAM-2B board, so you should enter <2>.

If your board has all the other attributes of the JRAM-2 except the page and board select shunts are located within the logic section of the board, you have a JRAM-2A, so you should enter <1>.

After you have entered your answers to these three prompts, you will receive an initial report.

The screen will display the following message:
Testing for JRAM banks
The following banks were found ...

followed by a listing with card number(s), segment number(s), bank number(s), type of chip and a series of address numbers.

NOTE: MEMTEST numbers its banks 0 - 7 starting with the bank to the furthermost left of the board (i.e., the bank furthest from the logic section).

If the initial report indicates an accurate number of banks (i.e., the same number of banks that you have populated on the board) followed by a series of 0's, this means that no bad banks were found on the preliminary run-through. Press the carriage return key to start the diagnostic for a thorough check of the memory.

You will have to wait between 10 and 20 minutes (depending on how much JRAM memory you have installed) while the diagnostic program runs through a complete cycle. If the memory checks out okay, you will see a single asterisk appear on your screen. Each time it runs through the cycle, another asterisk will appear as long as no bad memory is found. You may also see occasional periods. If any bad memory is found, an IC number will appear at the far right of the line.

It is usually sufficient to run the diagnostic for a half hour per half megabyte of RAM. But if you want to be sure to have a thorough and complete test of the memory, let it run longer (even overnight). **The diagnostic will continue to run indefinitely until YOU STOP IT BY PRESSING <Ctrl C>.**

If the initial report does not indicate an accurate number of banks, chances are that you have a bad chip or a badly stuffed chip in one or more of your banks. Press <1> to add a bank. DO NOT PRESS THE CARRIAGE RETURN KEY AFTER THE ONE! At this stage of MEMTEST, the carriage return has its own specific function, which is to start testing the RAM. If you do press the carriage return inadvertently, press <Ctrl C> to abort the program and start MEMTEST again.

After you have pressed the carriage return key, you will receive some additional prompts, which are relatively self-explanatory. When the program asks for which segment to use, enter any free segment from the choice available, i.e., one that was not on the list that appeared in the initial report. For example, you could enter D. The program will then ask you to enter <1> if you have 256K chips. If you have 64K chips, just press carriage return.

The screen will display a list of banks similar to the initial report except that it will have added the additional bank that you requested. Now press the carriage return key to start the diagnostic part of the program.

The screen will then begin to scroll repetitively. Stop the screen by pressing Ctrl S, and look at the far right of the line for the IC number. The IC numbers are also marked on the boards at the top of each bank. Replace the chip or restuff it if it is obviously loaded incorrectly. If you cannot fix the chip, remove the entire bank and press the carriage return key to start the diagnostic again.

NOTE: If you have more than one JRAM board in your computer, you can use MEMTEST to test them all at one time. But if the program indicates a bad chip, you must test the board that you suspect is faulty by itself. Set the board select shunt on the board to be tested to one. AFTER YOU HAVE FINISHED MEMTEST, BE SURE TO REBOOT YOUR COMPUTER.

12.2: THE PARITY SHUFFLE

Some users who have stuffed their own chips find that the RAM chips pass the MEMTEST diagnostic, but still cause a parity error. In most cases, this problem is due not to a "bad" chip, but rather a "noise sensitive" one. No two RAM chips will have the same level of sensitivity and some will be more prone to disruption by noise generated by the system.

The pattern in which the noise travels down the bus line strongly influences the way that the RAM chips will respond. Therefore, often the solution to a parity error is to change the bus pattern in a random fashion until the parity error stops. One possible solution is to change the positions of the boards in your machine. Please see your computer's reference manual for information on how the slots are numbered. Generally, slot one is the first slot on your left, with you facing the computer (as in the typing position). **The best arrangement for your boards is usually**

to place the JRAM board in slot one, followed by any additional memory boards or blank slots, and then, for example, the modem card, video card, disk/diskette controller card.

Another solution that often works is to shuffle around the chips on your board, which will change the resounding noise pattern. Finally, you may try pinpointing the source of the problem by depopulating your board one bank at a time until your parity error ceases. Or you can depopulate the entire board, and re-populate it one bank at a time. Keep going until the parity error re-occurs.

If you can narrow the problem down to one or two banks, you can swap in one chip at a time. All this is much easier and less time consuming if you have a common operation which causes the parity error to occur. If you are experiencing a parity error in a seemingly random fashion, be prepared to be patient.

ADDED

MEMTEST

12.3: INTRODUCTION TO MEMTEST

If you are experiencing "bad JRAM" messages or other memory errors, you may run MEMTEST to help pinpoint a bad chip. However, it is not necessary to run MEMTEST as part of the normal routine.

JBOOT.BIN INCLUDES ITS OWN MEMORY DIAGNOSTIC

When JBOOT.BIN is installed, it checks the memory each time you boot up. If it finds a memory problem, it will report a bad bank.

IN AN AT, THE IBM MEMORY DIAGNOSTIC TESTS ALL THE MEMORY INSTALLED

If you have an AT, and you have followed the standard installation procedure for installing a JRAM-AT or JRAM-AT3, the IBM memory diagnostic will test the memory on the JRAM when you boot up.

If you do wish to use MEMTEST, you must configure the JRAM-AT or JRAM-AT3 board for Mode Two.

MEMTEST is designed to identify a troublesome chip rather than a whole bank. However, the 16-bit data bus causes an echo effect. When MEMTEST is used with the JRAM-AT or JRAM-AT3 it indicates two chip numbers for a single bad chip. The pattern of the echo effect coincides with the pattern used for populating the board. It is probably easiest to replace the two chips that are indicated. If you prefer, you can replace only one of the pair. Note the IC number of the chip that you replace, and run MEMTEST again. If MEMTEST continues to report a bad chip, simply replace the other chip.

12.3.1: STARTING MEMTEST

1) If you have been using RAM disks, first save everything from your RAM disks onto a physical diskette, because MEMTEST will corrupt the contents of your RAM disk.

2) Copy the MEMTEST.COM file from your Tall Tree Systems distribution diskette onto a system diskette if you have not already done so. It is generally advisable to run MEMTEST with IBM PC DOS. You will be less likely to run into an address conflict if you MEMTEST with a very simple CONFIG.SYS--just DEVICE=JBOOT.BIN, rather than a complete file with the drivers for a RAM disk and EMS. Also, it is better to run the diagnostic at the standard clock speed and remove any other memory expansion cards.

3) Put your copy of MEMTEST.COM into drive A: and type:

```
        A>MEMTEST<ENTER>
```

The program will instruct you to enter the necessary information.

In order to receive a preliminary report, you must answer three prompts. These answers indicate which banks to use for testing and identify which JRAM board you have installed. If you only have JBOOT.BIN in your CONFIG.SYS file, you can answer the first two prompts as follows:

```
        D<ENTER>
        D<ENTER>
```

You will notice the following messages:

```
*Enter First Memory Bank to Test (1-9, A or D):
```

```
*Enter swapping page 9=640K  A=704K  D=896K
```

The answer to the third prompt depends on which JRAM board you have installed.

```
*Enter type 0=JRAM, 1=JRAM-2A, 2=JRAM-2B
or JRAM-3, 3=JRAM-3C 4=JRAM-2D or JRAM-3D
5=JRAM-AT, 6=JRAM-AT3
```

MEMTEST lists various models of our JRAM memory boards. The rev numbers for the JRAM-2 and JRAM-3 boards are printed on the back side of the memory board, along the upper edge. Type in the appropriate number for your particular board.

NOTE: The IC numbers are marked on the boards at the top of each bank. For example, column U1-9 includes IC numbers 01-09.

12.3.2: MEMTEST PRELIMINARY REPORT

1) After you have entered your answers to these three prompts, you should receive a preliminary report.

The screen should display the following message:

> **Testing for JRAM banks**
> **The following banks were found ...**

NOTE: If MEMTEST hangs at this point, try running without it without any CONFIG.SYS.

This message should be followed by a complete listing of your JRAM boards with card number(s), segment number(s), bank number(s), type of chip and a series of address numbers.

If the Preliminary Report indicates an accurate number of banks (i.e., the same number of banks that you have populated on the board) followed by a series of 0's, this means that no bad banks were found on the first run-through.

*Press the carriage return to start the diagnostic for a thorough check of the memory.

You will have to wait between 10 and 20 minutes (depending on how much JRAM memory you have installed) while the diagnostic runs through a complete cycle.

*If the memory checks out okay, you will see a single asterisk appear on your screen. Each time it runs through the cycle, another asterisk will appear as long as no bad memory is found. You may also see occasional periods. Four asterisks should be a sufficient test. You may want to run overnight if you have recently replaced some RAM.

*If any bad memory is found, an IC number will appear at the far right of the line. Because of the echo effect in the AT, you will receive two IC numbers for any single bad chip.

*It is usually sufficient to run the diagnostic for an hour per megabyte of RAM. But if you want to be sure to have a thorough and complete test of the memory, let it run longer (even overnight).

The diagnostic will continue to run indefinitely until YOU STOP IT BY PRESSING <Ctrl BREAK>.

2) This listing is a preliminary report. If you do not receive any list of banks found, please call our Technical Support Department (415)964-1397. For example,

2a) call Technical Support if you see

Testing for JRAM banks

without any list.

2b) or if you received no listing before the message

JRAM Memory Diagnostic Running.....

This message should only appear *after* the preliminary report listing the number of banks found.

3)If the initial report does not indicate all the populated banks (numbered 0-7), chances are that you have a bad chip or a badly stuffed chip in one or more of your banks.

> 3a)Press <1> to add a bank. DO NOT PRESS THE CARRIAGE RETURN AFTER THE ONE. If you do press the carriage return at this point, MEMTEST will ignore the missing bank and start testing the

RAM on the other banks. If you press the carriage return inadvertently before you have added the missing bank(s), press <Ctrl C> to abort the program and start MEMTEST again.

3b)After you have added the number of the missing bank, you should press the carriage return. Then you will receive some additional prompts.

3c)When you add a bank, MEMTEST asks which "segment" to use. This refers to a 65 byte block of memory to be used for the test. Refer to the Preliminary Report and select a segment that was not on the initial list. For example, you could enter D.

3d)The program will then ask you to enter <l> if you have 256K chips. If you have 64K chips, just press carriage return.

3e)The screen will display a list of banks similar to the initial report except that it will have added the additional bank that you requested. Now press the carriage return to start the diagnostic.

12.3.3: READING THE PRELIMINARY REPORT/ERROR MESSAGES

1)"bank"

1a)In the Preliminary Report, **MEMTEST refers to the physical banks of chips on your JRAM board.**

1b) MEMTEST numbers the eight JRAM banks 0 - 7 starting with the bank closest to the left edge of the board, that is, the bank furthest from the logic section). This follows the order in which the RAM chips are numbered, with the lowest IC numbers in the column furthest to the left.

This is just the opposite of the way that JBOOT numbers the banks! JBOOT numbers the banks starting with the bank to the furthermost right, that is, with the bank closest to the logic *section of the board. JBOOT follows the order in which the eight bit boards should be populated, that is, the column next to the logic section should always be populated first.*

1c) Check to see if the number of banks listed corresponds to the number of banks you have populated. If there are less banks listed than you have populated, this indicates a memory error. Press <1> to add a bank as described above.

2)**"card"**

2a) This refers to the way the board is set by the Page Select shunt or switches. For example, if your first board is set to page F, MEMTEST will report it as card five. If your first board is set to page E, MEMTEST will report it as card one.

3)**"seg"**

3a) If you see a zero after "seg", this indicates a memory error. Check the rest of the line for further error messages.

3b) If the preliminary report does not list all the banks you have populated, you will be able to add a bank, which may help to pinpoint the problem. To add a bank, you will have to select a segment that has not appeared on the original list.

4)"64K" "256K" or "1 bit"

4a)Beside the column of banks, you should see another column indicating the type of chips you have used to populate the board, either "64K" or "256K". If you see "1 bit" instead, this indicates a memory error. Check the rest of the line for further error messages.

4b)If you see the "1 bit" error message, add the number of the bank in which this error message occurred. Follow the instructions for adding a bank on pages 12-5 and 12-6. When you continue, the diagnostic will usually be able to pinpoint the bad IC.

5)zeroes

Beside the column listing the type of chip, you should see a couple of columns that are typically all zeroes.

The column immediately next to the type of chip shows the pass count and error center for each bank.

The following column usually has 8 zeroes followed by a slash and another zero. These digits reflect the bits in the bank. If you see any "ones" instead of "zeroes", this indicates a memory error. Check the rest of the line for further error messages.

12.3.4: CONTINUING THE DIAGNOSTIC

After you have pressed the carriage return, the screen will begin to scroll repetitively.

Stop the screen by pressing Ctrl Break.

Look at the far right of the line for the IC number. This is the number of the bad chip. Replace the chip and run MEMTEST again.

For other possible error messages, see Section 12.3.3 above.

NOTE: If you have more than one JRAM board in your computer, you can use MEMTEST to test them all at one time. But if you find a bad chip, you must test the board that you suspect is faulty by itself, with the board select shunt set to one. AFTER YOU HAVE FINISHED MEMTEST, BE SURE TO RESET ANY SWITCHES YOU HAVE CHANGED AND REBOOT YOUR COMPUTER.

SECTION FOURTEEN
HOW JRAM-3 WORKS

14.1: INTRODUCTION: WHAT JRAM-3 DOES

The microprocessor in the IBM PC has direct access to one megabyte of memory. This means that there are roughly one million "locations" for the computer to put data in or read data from. Each location holds one byte of data, and each location has a unique address. To read or change the contents of any particular memory location, the computer simply specifies which of the one million addresses it wants to deal with, and then reads from or writes to that location.

In fact, not all of these one million addresses are really available for use. Large blocks of memory addresses are reserved for special purposes, such as holding information used by the monitor screen or holding programs used to boot up the computer when you first turn it on. When you subtract all the reserved addresses from the one million possible addresses, you are left with 640K addresses that are available for programs to use.

640K addresses sounds like enough, but it often turns out to be a real limitation. One reason that people use floppy diskette drives and hard disk drives is to make more data available to the computer. Unlike addressable memory, however, these forms of storage involve mechanical devices that are very slow by the computer's standards.

JRAM-3 utilizes two special ways to get around the addressable memory limit. Both involve *bank-switching*, which is a flexible addressing method that lets a very large number of memory locations share the limited number of addresses. With JRAM-3, you can have the storage capacity of diskette or hard disk drives, without sacrificing the speed of electronic memory.

14.2: HOW IT WORKS: AN OVERVIEW

The one million addresses that identify memory locations are commonly thought of as being divided into sixteen "pages". Each page has 64K addresses. The sixteen pages are numbered (in hexadecimal) 0 through F.

Most data stored in addressable memory actually resides in electronic devices called RAM (Random Access Memory) chips. These RAM chips are, in turn, organized into "banks" of nine chips each. With the most common type of RAM chip, each bank holds 64K memory locations.

In ordinary memory, the memory locations in each bank of RAM chips are permanently set to respond to a particular page of addresses. A physical switch near the RAM banks specifies which page of addresses will access which RAM bank. If there is one bank (64K) of RAM chips on the computer's "motherboard", then the memory locations in those chips will be set to have the first page of memory addresses. The memory locations in the second bank of RAM chips will have the second page of memory addresses. When all the pages of memory addresses have been used up, you cannot add any more RAM banks. That is all you can have.

JRAM MEMORY

Tall Tree Systems pioneered a different type of memory board, the JRAM. Unlike an ordinary memory bank, each bank of JRAM RAM chips can be made to answer to any page of addresses, or not to answer to any address at all. You can make any desired bank appear in or disappear from any address page with a single command.

This flexibility is what makes JRAM boards so powerful. You can have up to sixteen megabytes of JRAM RAM banks hanging in limbo with no addresses at all. But when you want to read from or write to any of those banks, you simply command it by its code to appear at a convenient page of addresses, and it is available to use. When you are through, you can send the bank back into limbo and call up another one. When you call the first one back, whatever you stored in it will still be there, completely unaffected by its temporary stay in the twilight zone.

EMS MEMORY

More recently Lotus/Intel/Microsoft developed yet another means of accessing memory beyond the 640K boundary of ordinary memory. They refer to the additional 8 Mb of memory that can be accessed by this method as *expanded memory*. The JRAM-3 was developed to meet the expanded memory specification, and can be used with any *EMS software*. In the *EMS* method, one address bank is divided into four contiguous 16K blocks, which are also referred to as "pages". These "pages" are swapped around in a manner much like bank-switching of the larger blocks of JRAM memory. At this time, the expanded memory specification can only access 8 Mb of memory; whereas, JRAM 64K bank-switching can access up to 16 Mb of memory.

JBOOT.BIN and JDRIVE.BIN

JRAM-3 comes with software (the JRAM Combo Disk) that sets up the JRAM banks and keeps track of what is stored in them. This package can make some of the RAM memory simulate a diskette drive. This simulated disk is called a RAM disk or electronic disk, and for most applications it is the most convenient and useful way to take advantage of the large RAM memory made possible by JRAM flexible bank addressing scheme.

There are, however, other ways to use JRAM flexible bank switching capabilities. A PC using JRAM boards could have several programs effectively running simultaneously, each in a block of JRAM memory being switched in and out of address space by a special multi-tasking program. Graphics or animation programs could have megabytes of stored images, all available instantly in any address page with a single command, and with no distracting reading-in process to show on the screen.

To use JRAM memory in ways other than as an electronic disk, you have to buy or write special software to control the bankswitching. The next sections cover the details of controlling the JRAM banks. If you are familiar with assembly language programming, you may want to try controlling your JRAM memory yourself.

JPAGER.BIN

The combo disk includes another driver called JPAGER.BIN, which handles the expanded memory. If you would like to write your own application program for expanded memory, please refer to Section 63 of this manual.

14.3: THE NITTY-GRITTY: THE BANK SELECT REGISTER

On every JRAM board there is a special RAM chip called the bank select register. The contents of the bank select register control which RAM banks on the JRAM board appear at which address pages, and which do not answer to any addresses at all.

When you installed your JRAM board, you set the board select switches. These switches established the address of the bank select register on each JRAM board. There are eight possible bank select register addresses, allowing up to eight JRAM boards to be installed in one computer. Here are the eight bank select register addresses:

CARD #	SW 1	SW 2	SW 3	SW 4	SW 5	SW 6	BANK SELECT REGISTER MEMORY ADDRESS
1	ON	-	-	-	ON	-	E800
2	-	ON	-	-	ON	-	E400
3	-	-	ON	-	ON	-	E200
4	-	-	-	ON	ON	-	E100
5	ON	-	-	-	-	ON	F800
6	-	ON	-	-	-	ON	F400
7	-	-	ON	-	-	ON	F200
8	-	-	-	ON	-	ON	F100

Each bank select register has just a single address. We will see later how the sixteen memory locations can be written to individually.

The JRAM-3 installation instructions have you set the bank select registers' addresses starting with card #5, but this is purely a matter of convenience. Any order or combination of bank select register addresses will work equally well. If you set your board to page F, you will be able to use page E as memory space.

Pages E and F are both reserved for ROM, although the PC comes with ROM installed only in page F. The reason that the bank select registers can have addresses in these pages has to do with their special function and the fact that pages E and F are reserved specifically for ROM only.

Unlike ordinary RAM chips, the bank select registers cannot be read. They do not respond to read commands when the computer calls out their addresses. There is no way to check the contents of the bank select register. The only thing that ever gets to see the contents of a bank select register is the logic circuit on the JRAM board that uses the data stored in the bank select register to determine which bank of RAM on the board will respond to which page of addresses.

Because the bank select registers cannot be read, they do not interfere when the computer reads the ROM installed at the same addresses. Likewise, because pages E and F are reserved specifically for ROM, the computer can write data to the bank select registers' addresses without affecting the data stored in ROM at those addresses. It is impossible to write new data into ROM. So, even though the computer attempts to write new data into both the bank select register and the ROM installed at the same address, only the bank select register actually gets written on.

Each bank select register has sixteen memory locations, each of which effectively holds four bits, or a single hexadecimal digit. The sixteen memory locations are numbered 1 through F (hexadecimal), and they represent the sixteen pages of the address space. The hexadecimal digit stored in each location indicates which RAM bank on the board will respond to addresses in the page represented by that location. Each bank on a JRAM board has a one-digit identifying code. So, for example, if the bank select register's memory location A contains the code 4, then the bank code-named 4 is set to respond to addresses in page A.

If several bank select register memory locations contain the same bank code, then that bank will appear at several different address pages. On the other hand, since each location can only hold a single bank code, the only way to make two banks appear at the same address page is to put valid bank codes into the same bank select register location on two different boards. Incidentally, making two banks appear in the same address page in this way, or having a JRAM bank appear at an address page already occupied by an ordinary RAM or ROM bank, will result in either meaningless scrambled data, or the slightly corrupted contents of only one of the competing banks appearing when the computer tries to read the shared addresses. Fortunately, the contents of the banks are not damaged, and they can be used again as soon as they are assigned to different address pages.

Here are the codes for the eight RAM banks on each JRAM board, from left to right:

D, C, 9, 8, 5, 4, 1, 0

14.4: WRITING TO THE BANK SELECT REGISTER

We saw in the previous section that the bank select register on each JRAM board has a single unique address. In order to write a bank code into a particular bank select register memory location, you specify which location you want in the **data** you write to the bank select register's address.

Specifically, you "write" two hexadecimal digits to the bank select register's address, even though each memory location can only hold one digit. The first digit is the desired bank code. The second is the address page at which that bank is to appear. Some tricky circuitry on the JRAM board splits off this second digit and uses it to determine into which of the sixteen memory locations in the bank select register to write the first digit. For example, if you want the first bank on the second JRAM board to appear at address page 9, you write D9 (where D is the code for the first bank, and 9 is the desired address page) to E400:0000 (the address of the bank select register on the second JRAM board).

This is the "Appear" command. To summarize, the syntax of the "Appear" command is:

```
Write:  <bank code><desired address page>
To:     <bank select register address>
```

If you are writing an assembly language program to control JRAM-2 banks, you do this with a MOV command. In BASIC, you use a POKE statement. Or, you can write directly to a bank select register by using DEBUG.

The "Appear" command is complemented by the "Disappear" command. In order to send a JRAM-2 bank back into no-address limbo, you write an invalid bank code over the code of the bank you want to send into limbo. There are eight valid bank codes (D, C, 9, 8, 5, 4, 1, 0), which leaves eight invalid bank codes (F, E, B, A, 7, 6, 3, 2). Any one of these invalid codes will do.

The syntax for the "Disappear" command is:

```
Write:  <invalid bank code><address page to be emptied>
To:     <bank select register address>
```

You may want to try switching some JRAM banks by using DEBUG to write to the bank select registers.

A few cautions are in order. Remember that you generally do not want to make two JRAM banks, or a JRAM bank and an ordinary bank, appear in the same address page. You will not be able to successfully read either bank. However, you will be able to write to both banks simultaneously, if for some strange reason you want to load multiple banks with identical data.

Also, you do not want to put a JRAM bank in page E (unless you have set your bank select register addresses to page F), because sooner or later you will inadvertently write to the bank select registers and send some or all of your JRAM banks unpredictably into limbo or other address pages.

If you put a JRAM bank in page B, it will be overwritten by the monitor control system. This will usually not affect the display, but it wipes out the contents of the JRAM bank.

Finally, if you booted up with JBOOT.BIN in your CONFIG.SYS file, remember that JBOOT initially fills any empty address space up to page 9 with JRAM banks. Before you command another bank into one of these pages, send the first one into limbo. And before you do that, realize that the programs of DOS reside in part of DOS memory. If you have only a few banks of non- JRAM memory installed, some of DOS is probably stored in your lower JRAM banks. If you command these banks into limbo, you will loose part of DOS and probably loose control of your computer.

Of course, you can always reboot, or cold boot if necessary.

14.5: INITIALIZING JRAM MEMORY

When you first turn your computer on, the bank select register (like any other RAM) comes on containing random data. To prevent JRAM banks from appearing in random address pages and possibly crashing your machine, each JRAM board comes in in the "board disabled" state. In the "board disabled" state, no banks on the board can appear in address space, regardless of the contents of the bank select register. Your initializing program (or ours, JBOOT.BIN) first writes "disappear" commands to every location in each bank select register, then enables the board. This procedure overwrites all the potentially dangerous random data in the bank select register. You can also disable any JRAM board yourself by issuing the "board disable" command.

To disable a JRAM board, write a command to the bank select register's address segment **plus one**. For example, to turn off the first JRAM board, you might write D9 to E801:0000. If you are initializing the JRAM board, it will already be disabled. The board will remain disabled as long as the commands you write to it are sent to the bank select register's address segment plus one. To enable the board, write a command to the usual address (**not** the segment plus one). Continuing our example, to enable the board, you might write 4A to E800:0000. Either an "appear" or a "disappear" command will do. All the commands written to the bank select register segment plus one address, and the command written to the usual address, take effect at once when the board is enabled.

Your initializing program, then, starts by writing an invalid bank code (3, for example) for every address page (0 through F) to every bank select register's address segment plus one. For example, if you have one JRAM-2 board set so that the bank select register's address is E800:0000, your initializing program would write 30 to E801:0000, 31 to E801:0000, 32 to E801:0000...3F to E801:0000. Then, it would write the "appear" commands appropriate for your application to E800:0000. The first command written to E800:0000 enables the board and causes the desired bank to appear in the desired page.

The "board disable" and "board enable" commands are also useful for complex bankswitching schemes that involve switching multiple banks or entire boards at once. Your program can issue the "board disable" command, reshuffle the banks in any way, and then issue the "board enable" command. All the commands sent while the board was disabled (written to the bank select register's address segment plus one), and the last command written to the usual bank select register address, take effect immediately when the "board enable" command is issued. Disabling and enabling the board the board have no effect on the contents of the JRAM banks or the bank select register.

The syntax for the "Board disable" command is:

```
Write:  <any bank code><any address page>
To:     <bank select register segment plus one>
```

The syntax for the "Board enable" command is:

```
Write:  <any bank code><any address page>
To:     <bank select register address>
```

14.5.1: SETTING PARITY

If you initialize the JRAM board by booting up with JBOOT.BIN, our program will also set and check parity for you. We recommend that you let JBOOT.BIN do this work for you and then you can take over with your own application program.

However, if you do decide to write your own initializing program, you may want to set parity by writing zeros to all bytes.

14.6: CONTROLLING 256K CHIPS

This discussion so far has assumed that the JRAM boards are populated with 64K RAM chips. They can also be populated with 256K RAM chips, so that each physical bank of RAM actually contains not one, but four 64K banks of memory.

To specify which quarter of the physical bank you want to make appear or disappear with a JRAM command, you write the command to the appropriate one of four bank select register addresses. Using 64K chips, the third most significant digit (hexadecimal) of the segment of the bank select register address is always zero. This tells the JRAM board that your command applies to the first 64K of memory in the physical bank specified by the bank code. If the board is populated with 256K chips, you write commands for the second 64K of memory in the physical bank to the same bank select register address segment **plus** 10. Write commands for the third 64K to the address segment **plus** 20, and for the fourth 64K to the address segment **plus** 30.

For example, if your first JRAM board is populated with 256K chips, to make the first **quarter** of the first bank on the first JRAM board appear in page A, you write a command exactly as you would to make the first **bank** on the first JRAM board appear if it were populated with 64K chips:

```
Write DA to E800:0000
```

To make the third quarter of the same 256K bank appear at the same address page,

```
Write DA to E820:0000
```

To make a bank disappear, you write an invalid bank code to the bank select register exactly as you did for 64K chips. It does not matter what quarter bank of memory you specify; all that counts is the address page to be emptied and the invalid bank code.

Briefly, here is how the bank select register actually works. There is only one bank select register on each JRAM board, and it has just sixteen memory locations. Section 14.3, however, implied an oversimplification about these locations. Each bank select register location actually holds six bits. Four of these bits hold the bank code. The third most significant bit of the bank code is used to make the specified bank appear or disappear (zero or one). The remaining two bits in each bank select register location are filled with the value of the third most significant hexadecimal digit in the bank select register address, which must be zero, one, two, or three, specifying which quarter of the 256K physical bank to use.

TALL TREE SYSTEMS

SERIAL-SERIAL CLOCK/CALENDAR
or
SERIAL-SERIAL MODULE

S - A 11106

SECTION TWENTY
INSTALLING PIGGYBACK MODULE OPTIONS

20.1: INTRODUCTION TO PIGGYBACK MODULES

Each JRAM board has sockets for one piggyback module. Tall Tree Systems currently offers six such modules:

1) Clock/calendar module
2) Parallel port, serial port module
3) Parallel port, serial port, clock/calendar module
4) Serial port, serial port module
5) Serial port, serial port, clock/calendar module
6) JDISKETTE controller

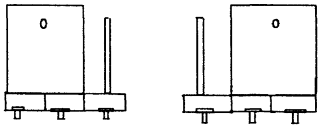
JRAM boards can be bought with the desired module already attached. piggyback module or you are adding or changing a module on a JRAM The connectors on the piggyback module are delicate. If you have to remove or insert a module, read the section on Removing and Attaching Modules.

To install any piggyback module other than the simple clock/calendar, you may have to change some of the options from the way they are set-on the module as delivered. Refer to the specific section for your module, explaining the installation details.

WARNING: When attaching a module to a JRAM board, always check the sex of the 25-pin connector on the JRAM expansion slot bracket. A serial-parallel module must go **only** on a board with a female 25-pin connector. A serial-serial module must go **only** on a board with a male 25-pin connector. Failure to observe this rule voids your warranty, and use of the port connected to the incorrect 25-pin connector may damage the port or your peripheral device. If you require an exchange for the proper sex connector, call Tall Tree Systems.

20.2: HOW TO SET SHUNTS

Figure 20.2: THE TWO SHUNT POSITIONS



Position "0" **Position "1"**

All piggyback module hardware options are selected by moving "shunts". The shunts are small black plastic hoods that contain metal contacts. Each shunt connects two of three metal pins. The shunt slips onto either the left pair or the right pair of either set of pins, selecting one option or another. To move the shunt, stick the end of a bent paper clip in the small hole and slide it off one pair of pins, then slide it onto another. The shunts are labelled, but the labels are hidden until you slip the black hoods off. When the shunt connects the left-hand pair of pins, it is in position 0; when it connects the right-hand pair of pins, it is in position 1. These positions are marked on the board.

SECTION TWENTY-ONE
REMOVING AND ATTACHING MODULES

21.1: NOTES ON REMOVING AND ATTACHING MODULES

If you have to remove or attach a JRAM piggyback module, please read the following. Properly handled, the modules can be inserted and removed hundreds of times without any problem. However, the connecting pins are fragile, and it is easy to bend and break off one of the pins. If you do bend a pin, contact Tall Tree Systems for service information. If the repair is not covered by the warranty, we will charge you a reasonable repair fee.

21.2: REMOVING SERIAL-PARALLEL-CLOCK OR SERIAL-SERIAL MODULES

To remove the serial-parallel-clock/calendar module or the serial- serial module, carefully snap the upper left corner of the module off the plastic spacer. Gently pry up on the upper right and lower left corners of the board (where the rows of terminals end). After these corners have pulled up slightly, lift the lower right corner a little. Keep alternating back and forth between the ends of the terminal rows and the corner of the terminal rows until the module pulls straight up off the JRAM board.

21.3: REMOVING THE CLOCK/CALENDAR MODULE

To remove the clock/calendar module, carefully snap the module off the plastic spacer in the center of the module. Gently pry up on the lower right corner of the module, then pry up on the lower left corner. Keep alternating between the two ends of the terminal row until the module pulls straight up off the JRAM board.

21.4: ATTACHING SERIAL-PARALLEL-CLOCK OR SERIAL-SERIAL MODULES

To attach the serial-parallel-clock/calendar module or the serial-serial modules, press the supplied plastic spacer over the hole in the upper left corner of the non-RAM section of the JRAM board. Hold the module at an angle to the JRAM board and set the single row of terminal pins lightly on the single row of sockets. Rotate the module down until the double row of terminal pins sets lightly onto the double row of sockets. Look closely at all the sockets to make sure that all the pins are sitting inside the socket cups. If any of the pins are not in the socket cups when you press the module down, you will bend them and possibly break them off.

Once you are satisfied that all the pins are properly aligned, gently press down on the upper right and lower left corners of the module. Then press down on the lower right corner, and keep alternating between the ends of the terminal rows and the corner until the module is firmly seated in the sockets. Note that some of the narrow part of the pins will still show when the module is fully inserted. Finally, snap the remaining plastic spacer into the hole from the module side.

21.5: ATTACHING THE CLOCK/CALENDAR MODULE

To attach the clock/calendar module, press the supplied plastic spacer into the lower center section of the non-RAM part of the JRAM board. Then carefully set the module terminal pins into the double row of sockets on the JRAM board.

Note: there are two plausible-looking ways to insert this module. The wrong one will burn out two chips on the PC motherboard (we know!). So, be absolutely certain that the hole in the center of the clock/calendar module lines up with the spacer attached to the JRAM board. This will position the leftmost module pins in the leftmost sockets. Also verify that all the pins are inside the socket cups, or they will bend or break when you press the module down.

When you are satisfied with the module's placement, gently press down alternately on one end of the terminal row and the other until the module is firmly seated in the sockets. Note that some of the narrow part of the pins will still show when the module is fully inserted. Finally, press the hole in the center of the module onto the plastic spacer.

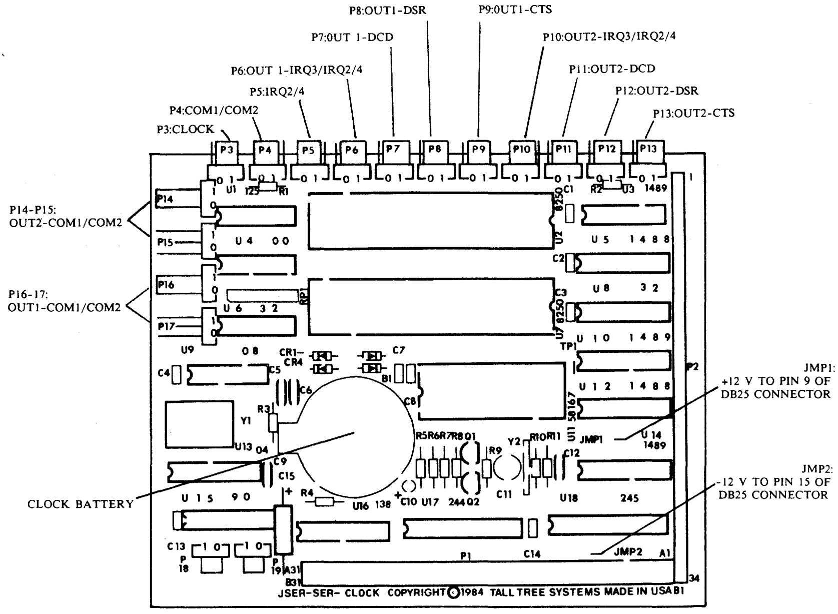
SECTION TWENTY-THREE
THE SERIAL-SERIAL MODULES

23.1: LOCATION OF SHUNTS

This section applies to both the serial-serial module and the serial- serial-clock/calendar module. The two modules are identical except that one has the clock/calendar components.

The shunts on the serial-serial modules are located along the top edge, left-hand edge, and lower left corner of the module. The shunts on the top edge are numbered 3 to 13, from left to right. The shunts on the left edge are numbered 14 through 17, from the top down. The shunts in the lower left corner are numbered 18 and 19, from left to right. The shunts are labeled on the edge of the module, but the labels are obscured when the shunts are in place.

Figure 23.1: Serial-Serial-Clock/Calendar Module



For a summary of the serial-serial module and serial-serial- clock/calendar module shunt settings, see section 23.4.

WARNING: Be sure that this module is connected to a JRAM board with a male 25-pin connector on the expansion slot bracket. Do not use a serial-serial module with a female 25-pin connector, because this would make it possible to plug a parallel device into the serial port, possibly damaging the parallel device. Using a serial-serial module with a female 25-pin connector voids the module's warranty. If you are adding a serial-serial module to an existing JRAM board with a female 25-pin connector, contact Tall Tree Systems to exchange the JRAM board for one with the correct sex connector.

23.2: THE SERIAL PORTS

23.2.2: SERIAL PORT SPECIFICATIONS

Both of the serial (asynchronous communications) ports on the serial- serial modules are RS-232C interfaces with male connectors rigidly mounted on the PC expansion slot bracket for durability and easy access. One of these is a standard DB25 connector. This connector will be referred to as Output 1. The other is a DB9 connector. This connector will be referred to as Output 2. Using this small DB9 connector lets the JRAM squeeze two serial ports onto the expansion slot bracket. Other I/O boards have a loose cable coming out the back of the machine or take up a second expansion slot for their second port. The JRAM board comes with an adaptor cable that links the standard female DB25 connector on your second serial device to the DB9 connector of Output 2.

With the proper multi-port software, the JRAM serial-serial modules let you use up to eight serial ports, of which two can be the ordinary ports you may already have.

23.2.2: SETTING THE SERIAL PORT SHUNTS

To install the serial ports, read each section below. Each of these sections includes a brief explanation of what one or more shunts does, and instructions about how you should set it. You will also find a section describing various ways to use your serial ports.

For a summary of shunt settings, see section 23.4. The shunt settings for port name assignments are shown in a table in Section 24. For an illustration of the shunt locations, see figure 23.1.

23.2.2.1: ASSIGN PORT NAMES

The IBM PC normally allows two serial ports, COM1 and COM2. The JRAM serial-serial modules allow up to eight separate serial ports. Two additional ports, COM3 and COM4, can be used through JSPOOL. To use more serial ports, you will have to purchase or write your own special software.

For information about configuring the serial-serial module to use more than two ports in your system, see Section 24.

The port names (COM1, COM2...) are assigned to the two output connectors according to the settings of shunts P14 through P17 (the ones on the left edge of the module) and the setting of shunt P4. A single shunt connector serves for both P14 and P15; another single shunt connector serves for P16 and P17. That is, if P14 is set to position 1, then there is no shunt connector at all on P15. If P17 is set to position 0, then there is no shunt connector on P16. Shunt P4 must always be set to either position 1 or position 0. Do not remove shunt P4.

* If you do not have any serial ports already, just leave the port name assignment shunts as they were shipped: P14 in postion 0, P16 in postion 0, and P4 in postion 0. These settings will cause Output 1 (the 25-pin connector) to be COM1, and Output 2 (the 9-pin connector) to be COM2.

* If you already have one serial port, you will have to change the shunts to make Output 1 (the 25-pin connector) to be COM2, and Output 2 (the 9-pin connector) to be COM3. To do this, move P14 to position 1, and move P4 to position 1. Leave P16 in position 0. COM3 will only be usable through JSPOOL or other special multiple- serial port software. For details about serial ports beyond COM2, see Section 24.

23.2.2.2: ASSIGN INTERRUPTS

DOS allows two interrupts for use by serial ports, IRQ3 and IRQ4. These interrupts are expected to correspond to COM2 and COMl, respectively.

The serial-serial module comes set so that Output 1 (the 25-pin connector) uses interrupt IRQ4, and Output 2 (the 9-pin connector) uses interrupt IRQ3.

* **If you have no other serial ports**, you can leave these settings as they are: P5 in position 1, P6 in position 1, and P10 in position 0.

* **If you already have a serial port**, you should already have set the JRAM serial-serial module ports to COM2 and COM3. In order to keep your ports as standard as possible, we suggest assigning IRQ3 to COM2, and disabling the interrupt on COM3. To do this, leave shunt P5 set to 1, move shunt P6 to position 0, and remove shunt P10. We suggest slipping the shunt on a single pin so it will not get lost.

The serial-serial module allows a variety of other serial port interrupts for specialized applications. See section 24 for details.

23.2.2.3: DISABLE READY LINES (OPTIONAL)

* If you plan to use store-bought cables designed to connect your particular peripherals to standard IBM PC serial ports, or if you plan to use the serial ports for modems, or both, verify that the following shunts are all

in position 1: P7, P8, P9, P11, P12, P13. The ready lines CTS, DSR, and DCD are enabled on both the serial ports, in accordance with the IBM serial port standard.

* If you are an experienced technican who is planning to make cables or reuse existing cables to connect serial devices to the JRAM serial ports, you may want to disable (force true) some of the RS- 232C ready lines. Read on for a brief discussion of ready lines and instructions about how to disable one or more of them.

The RS-232C interface standard includes three "ready" lines, CTS (Clear To Send), DSR (Data Set Ready), and DCD (Data Carrier Detect, or just CD, for Carrier Detect). Virtually all modems use all the RS- 232C lines in the same, simple way. A modem will plug directly into the serial port DB25 connector and work. There is only the loosest of standards, however, for other serial devices. Different devices use different ready lines, and some use lines that are not defined in RS- 232C at all. To use these devices, you must use either a special cable made to connect the particular device to the serial port used by the IBM PC, or make one of the "universal" cables described at the end of this section. The JRAM serial port allows one additional possibility: if you know exactly what you are doing, you may be able to use an existing cable by disabling (forcing true) some of the ready lines on the JRAM module itself. Check your devices's manual to see what ready line(s) it uses.

The JRAM serial-serial ports come set in the standard IBM serial port configuration, with all three ready lines enabled on both ports. To disable (force true) any of the ready lines, move the shunts as indicated in the table below. All of these shunts must be in either position 0 or position 1. Do not remove them completely.

Table 23.2.2.3: Disabling Ready Lines

Output 1 (25-pin connector):

P7 position	1: DCD used
	0: DCD not used (forced true)
P8 position	1: DSR used
	0: DSR not used (forced true)
P9 position	1: CTS used
	0: CTS not used (forced true)

Output 2 (9-pin connector):

P11 position	1: DCD used
	0: DCD not used (forced true)
P12 position	1: DSR used
	0: DSR not used (forced true)
P13 position	1: CTS used
	0: CTS not used (forced true)

23.2.2.4: POWER OPTIONS

A few peripherals, such as the Summagraphics digitizer pad and mouse, get their plus and minus 12 volts power from the computer by way of the serial port. Some other devices use the same connector pins for signals. The JRAM serial-serial modules allow you to use both types of devices. If you do not have one of the special serial devices that is powered in this way, skip this section.

To bring plus 12 volts to pin 9 of the 25-pin connector, you or a dealer must solder a jumper wire in the position marked JMP1 on the serial-serial module. To bring minus 12 volts to pin 15 of the 25-pin connector, solder a jumper wire in the position marked JMP2 on the serial-serial module. These jumper locations are illustrated in figure 23.1.

Do not plug anything other than the device for which this modification was made into the 25-pin connector. The voltage on pins 9 and 15 could damage peripherals that are not designed to use it.

23.2.3: USING THE SERIAL PORTS

There are two ways to use your serial ports. You can use the DOS MODE command, or you can use JSPOOL, the printspooler that-comes on the JRAM distribution diskette.

MODE: The IBM PC normally sends all its printer output to LPT1. With MODE, you can redirect the printer output to your serial port by typing

```
    A>MODE LPT1:=COM1:
        --or--
    A>MODE LPT1:=COM2:
```

depending on whether or not you have renamed your port. If you have special software that initializes more than two serial ports, MODE can redirect printer output to those, as well.

You then have to specify how the data is to be sent to your particular printer. See your printer manual and the DOS manual. As an example, if your COM1 printer required a baud rate of 9600, no parity, eight data bits, and one stop bit, you would type

```
    A>MODE COM1:9600,n,8,1
```

If you specify everything correctly, all printer output will be properly sent to your serial device.

You will probably want to put these MODE commands in an AUTOEXEC.BAT file. This file should be on all of your system diskettes (diskettes that you use to boot up your machine). See the DOS manual about AUTOEXEC.BAT files and the MODE command.

JSPOOL: The MODE command works, but JSPOOL is better. JSPOOL lets you control up to eight printers, four serial and four parallel, directly from the keyboard. It allows you to continue working with your computer while it is printing out files. JSPOOL also allows you to select a variety of printer features, from margins to pauses, to tailor your printing to your needs. See the JSPOOL manual and programs on your distribution diskette.

23.2.4: SERIAL PORT ADDRESSES AND PINOUTS

Port	Addresses
COM1	3F8-3FF (hex)
COM2	2F8-2FF
COM3	3E8-3EF
COM4	2E8-2EF
COM5	368-36F
COM6	268-26F
COM7	348-34F
COM8	248-24F

Most multiple port software will let you specify the addresses of each port, in the event that these do not match the default addresses.

Output 1 is the 25-pin connector on the JRAM expansion slot bracket. Output 2 is the 9-pin connector on the JRAM expansion slot bracket. The connector pins and the wires of the adapter cable are connected as shown in Table 23.2.4 below.

Table 23.2.4: Pinouts for Serial-Serial Modules
(DB9 is the 9-pin connector, and DB25 is the 25-pin connector)

Signal Name	OUT 1 DB25	OUT 2 DB9	Cable Wire	Cable DB25
TX (Transmit Data)	2	5	9	2
RX (Receive Data)	3	9	8	3
RTS (Request To Send)	4	4	7	4
CTS (Clear To Send)	5	8	6	5
DSR (Data Set Ready)	6	3	5	6
GND (Ground)	7	7	4	7
DCD (Data Carrier Detect)	8	6	2	8
DTR (Data Term Ready)	20	2	3	20
RI (Ring Indicator)	22	1	1	22
GND (Shield Ground)	1	-	-	-
+12 volts (only if JMP1 installed)	9	-	-	-
-12 volts (only if JMP 2 installed)	15	-	-	-

23.4: SUMMARY TABLE OF SHUNT SETTINGS

Shunt	Function	Delivered	Positions	Removed
P3	Clock	enabled	0 disabled 1 enabled	Same as 0
P4	COM1/COM2...	(see table in section 24.2.2.1)		Do not remove
P5	OUT1-IRQ2/4	position 1	0 IRQ2 to shunts P6 and P10 1 IRQ 4 to shunts P6 and P10	
P6	OUT1-IRQ3/4	IRQ4	0 IRQ3 1 IRQ4 or IRQ2 as selected by P5	No interrupt on IRQ2, IRQ3,or IRQ4
P7	OUT1-DCD	DCD used	0 DCD forced true 1 DCD used	Do not remove
P8	OUT1-DSR	DSR used	0 DSR forced true 1 DSR used	Do not remove
P9	OUT1-CTS	CTS used	0 CTS forced true 1 CTS used	Do not remove
P10	OUT2-IRQ3/4	IRQ3	0 IRQ3 1 IRQ4 as selected by P5	No interrupt on IRQ2,IRQ3, or IRQ4
P11	OUT2-DCD	DCD used	0 DCD forced true 1 DCD used	Do not remove
P12	OUT2-DSR	DSR used	0 DSR forced true 1 DSR used	Do not remove
P13	OUT2-CTS	CTS used	0 CTS forced true 1 CTS used	Do not remove
P14-15	COM1/COM2	OUT2-COM2	(see section 23.2.2.1)	OUT2 disabled
P16-17	COM1/COM2	OUT1-COM1	(see section 23.2.2.1)	OUT1 disabled
P18*		Removed	0 IRQ12 1 IRQ15	No interrupt on IRQ12 or IRQ15
P19*		Removed	0 IRQ10 1 IRQ11	No interrupt on IRQ10 or IRQ11
JMP1	+12 PIN 9	0 V PIN 9	Open: 0 volts to pin 9 Jumped: +12 volts to pin 9	
JMP2	-12 PIN 15	0 V PIN 15	Open: 0 volts to pin 15 Jumped: -12 volts to pin 15	

*JRAM-AT serial-serial module only

23.6: INTRODUCTION TO WIRING YOUR OWN SERIAL CABLES

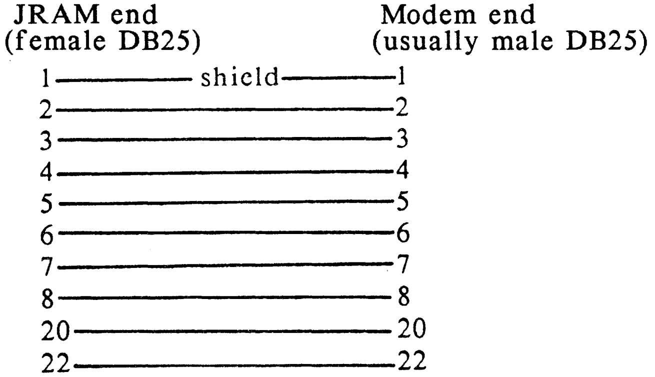
By wiring special cables for your serial devices, you can leave all the JRAM serial port shunts in their delivered positions and plug different serial devices into the same port at different times. Wiring your own cables for your serial devices is not as complicated as you might imagine. Virtually all modems use the same simple cable, and we have found that one of our two "universal" cable wirings will work with every printer that our users have tried. If you are ever in doubt about cable wiring, see your printer manual and IBM's Technical Reference Manual, or call the printer manufacturer.

All the cables described have DB25 (25-pin) connectors at each end. The JRAM end of the cable always has a female connector. The serial device end usually has a male connector, but not always. Check your printer or manual before buying any connectors. Since you are already wiring a custom cable, you may want to make the JRAM end a DB9 (9-pin) connector to plug directly into the JRAM serial port. By doing this, you can dispense with the adapter cable that came with the serial port, but you will not be able to use the special cable with any other serial port.

23.6.1: MODEM CABLES

Modem cables are simple. Every pin on the JRAM end is wired straight to the same number pin on the modem end. Not all these lines are actually used in most applications, but the unused lines do not hurt anything, and complete straight-through cables are easy to find or make.(See Figure 23.6.1)

FIGURE 23.6.1: MODEM CABLE



23.6.2: SERIAL PRINTER CABLES

All the serial printers we know of will work with one of the two "universal" serial printer cables described here. To determine which cable you need, check your printer manual. If your printer uses the RTS (Request To Send) ready line, use "universal" cable A. If your printer uses the DTR (Data Carrier Detect) ready line, use "universal" cable B. If your printer uses both, either cable will work. Many printers allow you to set which ready line is used. There is no reason to prefer one setting over the other.

FIGURE 23.6.2a: "Universal" CABLE A
(PRINTER USES RTS READY LINE)

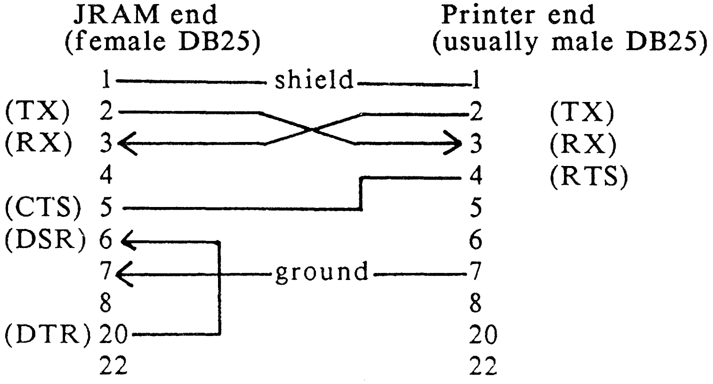
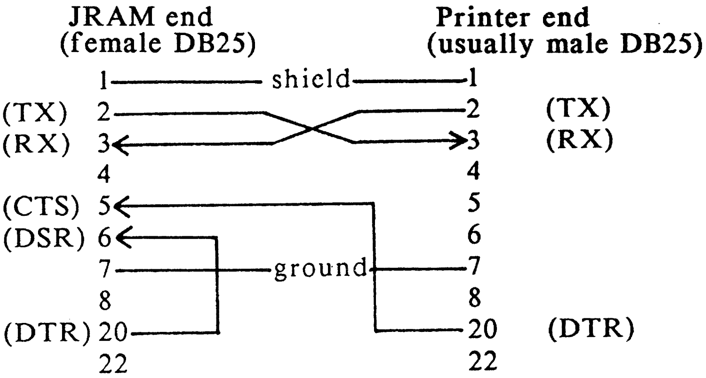


FIGURE 23.6.3b: "Universal" CABLE B (PRINTER USES DTR READY LINE)



These "universal" cables ignore some ready lines used by some printers to signal specific problems such as "out of paper" or "buffer full". These signals are different for every printer that uses them. However, no software that we know of ever checks for these signals, so there is no point in designing printer-specific cables to transmit them.

Under normal circumstances, the PC never checks pin 8 (DCD, or Data Carrier Detect) or pin 22 (RI, or Ring Indicator). However, if for some reason you want to simulate the old RS-232C interface standard better, you can connect pins 8 and 22 to pin 20 (DTR, or Data Term Ready) on the JRAM end of the cable. This should have no effect on the port's function.

SECTION TWENTY-FOUR
USING MORE THAN 2 SERIAL PORTS

24.1: INTRODUCTION TO USING MORE THAN TWO SERIAL PORTS

If you need multiple serial ports, you can install additional JRAM boards with the serial-serial modules. You can use up to eight serial ports in a single PC. Two may be ordinary serial ports; the remainder must be JRAM serial-serial module ports.

NOTE: DOS allows only two serial ports, COM1 and COM2, which use the two interrupts, IRQ4 and IRQ3. (There are additional interrupts on the AT, but DOS still allows only two serial ports.) To make use of the serial- serial modules' multiple-porting capabilities, you must use special software that sets up the additional serial ports, and you may need other software to control them, depending on your application. JBOOT.BIN (in the JETDRIVE package) sets up ports COM3 and COM4 in the table area reserved for them by DOS, and creates a similar table within itself for COM5 through COM8. JSPOOL can send data to printers, plotters, modems, and so on attached to COM1 through COM4. To receive data from these ports or to use COM5 through COM8, you need additional special-purpose software.

24.2: ASSIGNING COM NAMES ABOVE COM2

The port names (COM1, COM2...) are assigned to the two output connectors according to the settings of shunts P14 through P17 (the ones on the left edge of the module) and the setting of shunt P4. A single shunt connector serves for both P14 and P15; another single shunt connector serves for P16 and P17. That is, if P14 is set to position 1, then there is no shunt connector at all on P15. If P17 is set to position 0, then there is no shunt connector on P16. Shunt P4 must always be set to either position 1 or position 0. Do not remove shunt P4.

Table 24.2 shows how to assign the port names to the two serial port connectors. Notice that the two connectors on a single module cannot both have even names (COM2 and COM4) or both have odd names (COM1 and COM3).

Naturally, they also cannot both have the same name.

Assign the port names without skipping any COM numbers. If you have four serial ports, assign COM1 through COM4. If you do not, most multiport programs will rename the ports so that the names are sequential and start with COM1. If you already have one ordinary serial port, start your JRAM serial port names with COM2; if you already have two ordinary serial ports, start your JRAM serial port names with COM3.

TABLE 24.2
SHUNT POSITIONS for
USING MORE THAN 2 SERIAL PORTS

Output 1 (25-pin connector)

Output 2 (9 pin connector)

	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM8
COM1	X	P14-0 P16-0 P4-1	X	P14-0 P16-1 P4-1	X	P14-0 P17-0 P4-1	X	P14- P17- P4-1
COM2	P14-0 P16-0 P4-0	X	P14-0 P16-1 P4-0	X	P14-0 P17-0 P4-0	X	P14-0 P17-1 P4-0	X
COM3	X	P14-1 P16-0 P4-1	X	P14-1 P16-1 P4-1	X	P14-1 P17-0 P4-1	X	P14-1 P17-1 P4-1
COM4	P14-1 P16-0 P4-0	X	P14-1 P16-1 P4-0	X	P14-1 P17-1 P4-0	X	P14-1 P17-1 P4-0	X
COM5	X	P15-0 P16-0 P4-1	X	P15-0 P16-1 P4-1	X	P15-0 P17-0 P4-1	X	P15-0 P17-1 P4-1
COM6	P15-0 P16-0 P4-0	X	P15-0 P16-1 P4-0	X	P15-0 P17-0 P4-0	X	P15-0 P17-1 P4-0	X
COM7	X	P15-1 P16-0 P4-1	X	P15-1 P16-1 P4-1	X	P15-1 P17-0 P4-1	X	P15-1 P17-1 P4-1
COM8	P15-1 P16-0 P4-0	X	P15-1 P16-1 P4-0	X	P15-1 P17-0 P4-0	X	P15-1 P17-1 P4-0	X

TECHNICAL NOTE: Strictly speaking, these shunts do not control the names COM1 through COM8. What they control is the addresses at which the ports receive and deposit data. When the computer boots up, it checks the possible

addresses (without a special program, DOS checks only two) to see if ports are present. The first port found gets named COM1, the second COM2, and so on. If there is only one port, it will be named COM1, regardless of which of the checked addresses it has. In order to have the settings in the table above correspond to the COM names shown, you have to use the settings in COM name order. That is, if you have no other serial ports and you set the JRAM ports to be COM1 and COM2 as shown in the table, they will be COM1 and COM2 after the computer has booted up. If you set them to be COM2 and COM3, then even though JBOOT.BIN sets up the ports, they will still be named COM1 and COM2 after booting up.

Finally, if you already have a serial port that by some strange chance is set to the second address checked by DOS, and you have followed the instructions and set your JRAM ports to COM2 and COM3, you will end up with two ports assigned to COM1, and one port assigned to COM2. If this happens, reset your old serial port to COM1.

SERIAL PORT ADDRESSES

Port	Addresses
COM1	3F8-3FF (hex)
COM2	2F8-2FF
COM3	3E8-3EF
COM4	2E8-2EF
COM5	368-36F
COM6	268-26F
COM7	348-34F
COM8	248-24F

24.3: ASSIGNING INTERRUPTS TO MORE THAN TWO SERIAL PORTS

The JRAM serial-serial modules allow multiple serial ports to share one or both of the two interrupts provided by DOS. They also allow serial ports to use or share interrupt IRQ2 in PCs, XTs, and compatibles, and interrupts IRQ2, IRQ10, IRQ11, IRQ12, and IRQ15 in ATs and compatibles. The port controlling software has to take care of determining which port issued a shared interrupt. Alternatively, if your software does not make use of the serial port interrupts, they can be disabled.

The interrupts are assigned to Output 1 and Output 2 by shunts P5, P6, and P10. The optional shunts P18 and P19 allow the four additional interrupts supported by the AT and must be specifically requested when ordering. Note that not all possible combinations of interrupts are allowed on a single module.

To disable any interrupt for Output 1, remove shunt P6 and P18, if present. To disable any interrupt for Output 2, remove shunt P10 and P19, if present.

Table 24.3.1: Serial Port Interrupt Assignments

STANDARD SET

OUT 1	OUT 2	P5	P6	P10
IRQ 4	IRQ 3	1	1	0
IRQ 4	IRQ 4	1	1	1
IRQ 3	IRQ 3	1	0	0
IRQ 3	IRQ 4	1	0	1
IRQ 2	IRQ 3	0	1	0
IRQ 2	IRQ 2	0	1	1
IRQ 3	IRQ 3	0	0	0
IRQ 3	IRQ 2	0	0	1

SPECIAL SET

(Request additional shunts from factory)

OUT 1	OUT2	P5	P6	P10	P18	P19
IRQ 3	IRQ 10	0	0	-	-	0
IRQ 2	IRQ 10	0	1	-	-	0
IRQ 3	IRQ 10	1	0	-	-	0
IRQ 4	IRQ 10	1	1	-	-	0
IRQ 3	IRQ 11	0	0	-	-	1
IRQ 2	IRQ 11	0	1	-	-	1
IRQ 3	IRQ 11	1	0	-	-	1
IRQ 4	IRQ 11	1	1	-	-	1
IRQ 12	IRQ 3	0	-	0	0	-
IRQ 12	IRQ 2	0	-	1	0	-
IRQ 12	IRQ 3	1	-	0	0	-
IRQ 12	IRQ 4	1	-	1	0	-
IRQ 15	IRQ 3	0	-	0	1	-
IRQ 15	IRQ 2	0	-	1	1	-
IRQ 15	IRQ 3	1	-	0	1	-
IRQ 15	IRQ 4	1	-	1	1	-
IRQ 12	IRQ 10	1 or 0	-	-	0	0
IRQ 12	IRQ 11	1 or 0	-	-	0	1
IRQ 15	IRQ 10	1 or 0	-	-	1	0
IRQ 15	IRQ 11	1 or 0	-	-	1	1

"-" indicates shunt removed

SECTION TWENTY-FIVE
THE CLOCK/CALENDAR MODULE

25.1: CLOCK/CALENDAR SPECIFICATIONS

The JRAM clock/calendar is a twenty-four hour clock and four year calendar, backed up by a lithium battery. Used with the TREETIME program supplied on your distribution diskette and a simple AUTOEXEC.BAT file, the clock automatically sets the time and date whenever you turn on your PC.

25.2: DISABLING THE CLOCK/CALENDAR

The JRAM clock comes enabled. If you already have a clock that you prefer, or if you do not want the clock, just do not use TREETIME. If for some reason you want to electrically disable the clock, note the appropriate directions for your particular module:

1) If you have the serial-parallel clock/calendar, move shunt P7 from position 1 (enabled) to position 0 (disabled).

2) If you have the serial-serial clock/calendar, move shunt P3 from position 1 (enabled) to position 0 (disabled).

3) If you have the clock/calendar module, remove the entire module from the JRAM board. See Section 21 for details.

25.3: THE CLOCK/CALENDAR BATTERY

When the computer is turned off, the JRAM clock runs on its lithium battery. This battery should last at least a year, perhaps two years if your computer is on a lot of the time. When TREETIME displays the wrong time after you boot up, suspect that the battery has died. To replace the battery, remove the JRAM board from the computer. Lift the top clip of the battery holder very slightly with your fingernail or a small screwdriver. Press down on the rim of the battery near the base of the metal clip. The opposite edge of the battery will rise, and you will be able to slide it over the rim of the holder and out from under the clip. Replace the battery with a BR2325 lithium battery, available at computer dealerships and well-stocked drug stores. Make sure that the "plus" side of the battery is up, and that the battery and contacts are clean. Naturally, you will have to reset the clock after replacing the board and booting up.

25.4: SETTING THE TIME AND DATE

To set the time on the JRAM clock/calendar, first set the time and date as you would normally (by typing in "DATE" and "TIME", followed by the appropriate numbers. See the DOS manual for details). Then put a copy of your Tall Tree Systems distribution diskette in drive A: and type

```
A>TREETIME X
```

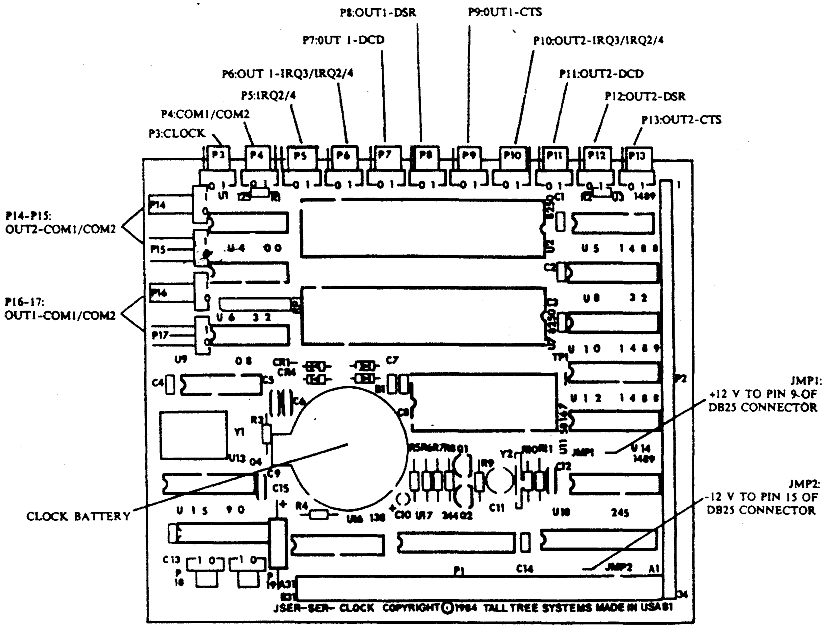
The computer will report the appropriate time and date information. Note that **anything** after the command TREETIME, including a comment in an AUTOEXEC.BAT file, will cause the clock/calendar to be reset. Do not follow the TREETIME command with anything except <enter> unless you want to reset the JRAM clock. The last thing to do is to copy the TREETIME.COM file onto your boot disk and include the TREETIME command in your AUTOEXEC.BAT file. If you do not already have an AUTOEXEC.BAT file, create one by typing

```
A>COPY CON AUTOEXEC.BAT
TREETIME
```

Each time you boot up, the clock will automatically set the time and date for you. See the DOS manual for more about AUTOEXEC.BAT files.

SERIAL-SERIAL CLOCK MODULE
ERRATA SHEET
7/31

Please note that in Figure 23.1 on page 23-1, Shunt 5 should be set to position one.



SECTION THIRTY
INTRODUCTION TO THE COMBO DISK

30.0: WHAT YOU GET:

Your Tall Tree Systems distribution diskette contains a number of files, including both the standard programs for JETDRIVE and JSPOOL as well as special programs for use with specific models of our boards. You may not wish to use all of these programs. But you must install at least JBOOT.BIN in order to access the memory on your JRAM board.

The standard JETDRIVE files are as follows:

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.

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.

Two files are <u>only</u> for the JRAM-3 boards, which support the Lotus/Intel/Microsoft specification for expanded memory:

JPAGER.BIN

This file is an installable device driver that manages
the expanded memory bank-switching operations.

JPAGER.ASM

This file is the source code for JPAGER.BIN.

The following files are part of the JSPOOL program:

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.

There are still more programs on the diskette!

MEMTEST.COM

This is the memory diagnostic program for the JRAM-boards. You will find directions on MEMTEST in section 12 of your manual. If you have populated the board yourself, you should test your JRAM memory by running this program.

TREETIME.COM

This program is only for boards that have a piggyback module with JCLOCK, Tall Tree Systems' clock/calendar.

30.0.1: EXAMPLE AND SYNTAX NOTATION

30.0.1.1: EXAMPLE NOTATION

In the installation sections, and throughout the manual, you will see commands and messages shown in a particular format. Here are the details that you will need to understand this format.

* Commands that you must type are always shown preceded by a drive letter prompt, such as A>. This prompt is supplied by the computer. Do not type a prompt before your commands.

* Certain keys are indicated by <>. Other keys are identified by the letters printed on the key cap. For example, <F6> means the key that has "F6" on it, which is located on the left side of the IBM PC keyboard.

* A few commands are issued by holding down several keys at the same time. Pressing the <Ctrl>, <Alt>, and <Del> keys at the same time, for example, causes the computer to reboot. This is written as <Ctrl><Alt><Del>. The other combinations that must be pressed simultaneously are <Ctrl><Alt><K>, <Ctrl><Alt><->, and <Ctrl><Alt><F>.

* All commands take effect when you press <enter>, and all lines are terminated by pressing <enter>. The <enter> key is not included in the examples.

* All the examples are shown in upper case, just for consistency. You can type in commands and files to the computer in upper case, lower case, or any combination of upper and lower case.

30.0.1.2: SYNTAX NOTATION

The following syntax notation is used in this manual:

* Capital letters: You must type everything shown in capitals.

* <x>: You must type in the information requested inside angle brackets. Do not type the angle brackets.

* <file name> means the full name of a file, with extension. Wild cards are allowed in file names.

* <file list> means the name of a file list file, with extension. Wild cards are not allowed in file list names.

* <source> means the drive letter of the drive from which the files are to be transferred.

* <target> means the drive letter of the drive to which the files are to be transferred.

* <subdir> means the name of a subdirectory.

* <option> means a letter specifying an option, such as P, N, or XCL.

* <#> means a decimal (not hexadecimal) number.

* All other <>'s that contain cryptic letters such as <Ctrl>, <F6>, <Alt>, and so on, refer to the single keys with those letters on them.

* [x]: Things shown in []'s are optional. Do not type the brackets.

* x...: The ellipsis (...) means that the preceding item may be repeated.

* Punctuation: All colons, slashes, backslashes, and spaces must be typed as shown.

* Upper and lower case letters are allowed, including any combinations of upper and lower case.

* To execute a command or to end a line, press the

carriage return key. For clarity, this key is not shown in the syntax descriptions.

* The syntax expression may be longer than one line. This is only due to the bulkiness of the notation. Commands will almost always be less than one line long. However, you can enter commands of up to 128 characters. The command will automatically wrap around on the screen. Do not press the carriage return key to start the second line.

For example, this syntax expression

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

represents all of these legal commands:

```
    JET A: TEXTSUB MYFILE B: BSUB

    JET A: TEXTSUB DOCSUB B:/P

    JET  textsub DOCSUB MYFILE b:/P/u/N
```

30.0.2: COPYING FILES FROM THE COMBO DISK

You may wish to copy the programs you plan to use from the distribution diskette at this point. More detailed information on installing the programs is provided in the following JETDRIVE/JSPOOL sections.

1) If you boot up from a floppy drive, make a copy of your current boot diskette. If you boot up from a hard disk, skip this step.

If you have two 360K drives, put your boot diskette in drive A: and a new (blank) diskette in drive B:. Make sure that the file DISKCOPY.COM is on your boot diskette. If it is not, copy it from your DOS master diskette.

If you only have one 360K diskette drive, you will have to create your new system diskette using only one drive. DOS will prompt you to swap diskettes as necessary to duplicate your boot diskette.

Now type in:

```
A>DISKCOPY A: B:
```

This will make an exact copy of your boot diskette. After the copy has been completed, put your original diskette away as a backup. If you need to make additional boot diskettes, you can DISKCOPY the one you have just made.

2) Use the DOS COPY command to copy the files you plan to use from your Tall Tree Systems distribution diskette to your new boot diskette, or to the root directory of your hard disk if you boot up from a hard disk. Remember, you must install JBOOT.BIN in order to access the memory on your JRAM board. To copy these files, put your Tall Tree Systems distribution diskette in drive B:, and your new system diskette in drive A:.

Now type in:

```
A>COPY B:JBOOT.BIN
```

The computer will respond:

```
One file copied
```

Now copy the other files in the same way. Again, if you have only one 360 K drive DOS will prompt you to swap diskettes as appropriate. See the DOS manual if you are not familiar with this procedure.

<u>30.0.3: CONFIG.SYS</u>

You will not be able to use your JRAM-2 or JRAM-3 board until you create a CONFIG.SYS file, which installs at least JBOOT.BIN. If you do not want to use the RAM disk, printspooler, or expanded memory, you may simply install JBOOT.BIN to fill out your DOS size with the memory on the JRAM board. If you want to use this minimal CONFIG.SYS, you should have no more than 704 K total memory in your computer (that is, motherboard memory, JRAM memory, and memory on any other expansion cards that you may have in your system).

CONFIG.SYS (for CONFIGure your SYStem) is a file that instructs DOS to load any installable device driver(s) specified in the file into memory each time you boot up. If you are already familiar with the CONFIG.SYS file and have established one on your system diskette or hard disk, you can skip the step by step explanation. If you have never set one up before, read this completely before proceeding.

1) First check to see if you already have a CONFIG.SYS file by typing:

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

The line, TYPE CONFIG.SYS, instructs the computer to display the CONFIG.SYS file on the screen if it already exists.

2) If you see the message,

```
File not found
```

you should create a CONFIG.SYS file as described below. If a file is displayed, you should append everything in your old CONFIG.SYS to the end of the new one. JBOOT.BIN must be the first driver listed in the CONFIG.SYS file.

3) Now type, for example:

```
COPY CON CONFIG.SYS
DEVICE=JBOOT.BIN 640K
<Function key F6>(Screen will display ^Z)
```

You will see the message:

```
One file 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. The CONFIG.SYS file above is for a system that has no more than 640 K total memory and has allocated all of the memory to DOS. If you have less memory in your computer, simply type the appropriate amount after DEVICE=JBOOT.BIN. The amount must be separated from JBOOT.BIN by a space, and must be followed by an upper or lower case K.

If you have more than 704 K total memory in your computer, you will have to type in a little more information so that the extra memory can be allocated or reserved.

The simplest way to do this is to list JDRIVE.BIN in your CONFIG.SYS file, which will cause your computer to allocate the surplus memory to a RAM disk.

For example, you could type:

```
COPY CON CONFIG.SYS
DEVICE=JBOOT.BIN 640K
DEVICE=JDRIVE.BIN
<Function key F6>(Screen will display ^Z)
```

Most JRAM users want to use our RAM disk program. For additional information on the many features of the

JETDRIVE program, please read section 30.1 below.

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 SystemS' 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

The JETDRIVE programs are all "installable device drivers". Installable device drivers are programs that add features to DOS without altering the way DOS functions. Therefore, the JETDRIVE programs are compatible with most 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. Use JET also for various kinds of copying, comparing, updating, and backing-up functions for diskette and hard disk drives.

SECTION THIRTY-ONE
INSTALLING JETDRIVE

31.1: INSTALLING JETDRIVE

1. If you have not already done so, make a copy of the diskette you normally boot up with, and put it away as a backup. If you have a hard disk, ignore this step.

2. If you have not already done so, copy JBOOT.BIN, JDRIVE.BIN, MAKEDOS.COM, and JET.COM from your Tall Tree Systems distribution diskette onto your new boot diskette or your hard disk.

3. Check to see if you already have a CONFIG.SYS file on your boot disk or hard disk by typing:

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

If you see the "File not found" message, you should create a CONFIG.SYS file as described below. If a file is displayed, you should append everything in your old CONFIG.SYS to the end of the new one.

4. Now you have to create a CONFIG.SYS file that lists the JBOOT.BIN and JDRIVE.BIN drivers. JBOOT.BIN must be the first driver listed in CONFIG.SYS. When the computer boots up, it will check the CONFIG.SYS file and automatically install any programs listed in it. In the CONFIG.SYS file, you must specify how much memory you want to allocate to DOS, as opposed to the RAM disks.

Memory can be used for the RAM disks, DOS, or *expanded memory. (Expanded memory is only possible with JRAM-3 boards and software that supports the Lotus/Intel/Microsoft specification.)* DOS memory is the memory used to hold programs and data that are currently being manipulated by the computer. Before you install Tall Tree Systems' enhancements, DOS memory is the only kind of memory you have.

You should probably allocate at least 128K to DOS. For most applications, we recommend allocating 192K. Some programs can take advantage of more DOS memory. With JRAM-2 and/or JRAM-3 boards, you 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.

Note: Do not use the DOS command (ASSIGN) to rename your drives. 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. For more information on CHKDSK, see your DOS manual.

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.

For example, you may use an AUTOEXEC.BAT file to instruct DOS to read the COMMAND.COM file directly from the RAM disk. COMMAND.COM holds your basic DOS commands such as COPY, REN, and DEL. Many applications programs overwrite COMMAND.COM while they are running. When you have finished your application program, DOS will automatically search the drive you booted up from for COMMAND.COM and read it back into memory.

If your boot disk is not in the drive, you will receive an error telling you to insert a system diskette. To avoid having to keep a bootable diskette in drive A: and to speed the process of quitting an application program, you can make DOS read COMMAND.COM from the RAM disk.

If your RAM disk is C:, you would add these lines to the end of your AUTOEXEC.BAT file:

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

The first line uses JET to transfer COMMAND.COM to the RAM disk, and the next two force DOS to check C: for COMMAND.COM.

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.

* Since COMMAND.COM is rarely needed on a RAM disk, JET does not transfer it unless /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 3.20 (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 you to 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
<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
<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 3.20 (c)1982,83,84,85 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.

/B Boot

This option causes JET to copy the boot sector from the source diskette onto sector zero of the target diskette. The source diskette must have a boot sector for this to work. Because /B does not transfer the hidden DOS files IBMBIO.COM and IBMDOS.COM or the file COMMAND.COM, to make a bootable diskette, you must also specify the /O option, which does transfer these files. For example, to copy a bootable system diskette, type:

```
   JET A: B:/B/O
```

The boot sector of the source diskette must be compatible with the format of the target diskette! This is the one case in which JET requires the source and target diskettes to have the same format.

Also, note that the target diskette must be formatted, but blank. Because the boot sector tells DOS where on the diskette to look for IBMBIO.COM and IBMDOS.COM, they must be placed in the same location on the diskette every time. If there are already files on the diskette, JET would have to put IBMBIO.COM and IBMDOS.COM after the old files, and the computer will not be able to find IBMBIO.COM and IBMDOS.COM when it boots up. If you want to reuse a diskette that already has files on it, you can make JET erase the old files before transferring the new ones by typing:

```
   A>JET A: B:/B/O/E
```

Do not use /B when JETting files to the electronic disk. If you do, you will overwrite data used to restore the electronic disk after a warm boot, and the contents of the electronic disk will be lost next time you warm boot. Since you cannot boot up from the electronic disk anyway, there is no reason to use the /B option in this way.

/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.

/O Operating System

This option causes JET to transfer the hidden DOS files IBMBIO.COM and IBMDOS.COM, and COMMAND.COM. Used in conjunction with the /B option, which transfers the boot sector, this option lets you copy bootable system diskettes.

/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=<#>B][X=<#>B][<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=12B.

* R=<#>B 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=<#>B 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=10B. 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-3 to page F), use -X, like -X=13B.

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. First, check to see if you already have a CONFIG.SYS file by typing:

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

If you see the "file not found" message, then you do not already have a CONFIG.SYS file and you will have to create one. Type:

```
COPY CON CONFIG.SYS<ENTER>
DEVICE=JSPOOL.BIN<ENTER>
<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. The buffer will take up one quarter of the available DOS memory. To specify a larger or smaller buffer, simply specify the 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 see a file displayed, then you already have a CONFIG.SYS file. To install JSPOOL, you will have to add one more line to CONFIG.SYS.

A typical CONFIG.SYS file looks like:

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

This CONFIG.SYS file sets up Tall Tree Systems' JETDRIVE package. If you have installed other installable device drivers or DOS options, 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 two lines like:

```
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 space then 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 a combination that is extremely unlikely to occur otherwise in text. 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 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 **@#$** commands. Features such as tabs set with command T and margins set with command M will work properly. However, JSPOOL will no longer recognize <u>any</u> further commands sent to it, because all commands are translated internally into the **@#$** 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 **@#$** commands. JSPOOL acts as a simple, "straight wire" spooler, passing data into and out of its buffer without regard to what the data is. This level of "dumbness" allows JSPOOL to handle any kind of graphics or microjustification data without errors. JSPOOL will no longer recognize <u>any</u> further commands sent to it, because all commands are translated internally into the **@#$** 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 before 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/SBDF>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.

44.3.3.3: MAKING A CONTIGUOUS FILE

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
```

SECTION SIXTY
INTRODUCTION TO JPAGER

60.1: WHAT IS JPAGER?

JPAGER is an installable device driver that manages the expanded memory on your JRAM-3 or JRAM-AT3.

The one million addresses (0-1024K) that identify memory locations can be divided into either 64K byte blocks or 16K byte blocks. The expanded memory specification allows your computer to access up to 8 Mb of memory by bank-switching four contiguous 16K blocks. This 16K block bank-switching is handled by JPAGER.BIN.

In order to take advantage of this expanded memory, you must use *EMS* software. If you are a programmer, you can write your own expanded memory application program. Please refer to Section 63 for details.

If you want to use prepared *EMS* software, you should simply add JPAGER.BIN to your CONFIG.SYS file. Your expanded memory application program will take care of the rest.

See Section 61 for examples of some typical CONFIG.SYS files.

SECTION SIXTY-ONE
INSTALLING JPAGER

61.1: INSTALLING JPAGER

To use JRAM-3 with *EMS* software, you must install both JBOOT.BIN and JPAGER.BIN in a CONFIG.SYS file.

If you have already installed JBOOT.BIN, you will have to re-copy your CONFIG.SYS file and add one more line to install JPAGER.BIN.

1. If you have not already done so, 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. If you have not already done so, copy JBOOT.BIN and JPAGER.BIN from your Tall Tree Systems distribution diskette onto your new boot diskette.

3. Check to see if you already have a CONFIG.SYS file on your boot disk or hard disk by typing:

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

This line instructs the computer to display the CONFIG.SYS file on the screen if it already exists.

If you see the message,

```
File not found
```

you can create a simple CONFIG.SYS file as in step #4.

4. Now type for example,

```
A>COPY CON CONFIG.SYS
DEVICE=JBOOT.BIN 640K
DEVICE=JPAGER.BIN
<F6> (Screen will show ^Z)
```

The "COPY CON" line creates a file called CONFIG.SYS that contains whatever is typed in from the keyboard (CONsole) until function key <F6> is pressed. The CONFIG.SYS file above allocated 640K to DOS and the rest of the memory to expanded memory.

5. If you have already installed JBOOT.BIN and JDRIVE.BIN, you will see something like:

```
DEVICE=JBOOT.BIN 640K
DEVICE=JDRIVE.BIN
```

This CONFIG.SYS file allocates 640K to DOS and sets up one RAM disk that uses all the remaining memory in the computer.

If you want to use both JDRIVE's RAM disk and expanded memory, you need to insert some additional information, including a specific amount of memory to be allocated to the RAM disk. For example, you might type or retype:

```
A>COPY CON CONFIG.SYS
DEVICE=JBOOT.BIN 640K
DEVICE=JDRIVE.BIN 512K
DEVICE=JPAGER.BIN /S=10B
<F6>(Screen will show ^Z)
```

This CONFIG.SYS file allocates 640K to DOS and 512K to a RAM disk, leaving all the remaining JRAM-3 memory to expanded memory. The amount of memory that you allocate to the RAM disk may be any multiple of 64K less than the total amount of memory already allocated to DOS. Be sure to leave sufficient memory to run your expanded memory software programs.

The /S=10B specifies the "swapping page" or "window" used by the expanded memory to be page 10 (or A in hex). This parameter is necessary because both JDRIVE.BIN and JPAGER.BIN default to page 13 (D in hex), and conflicts may arise if both try to use the same swapping window. Other swapping window conflicts are possible. You will have to specify a different swapping page if you want to use a JSPOOL buffer on the RAM disk, allocate 704K to DOS, or use the JRAM-3 together with a Tecmar Graphics Master board or some other high-performance graphics display boards. See Section 62 for alternate swapping pages.

If you have changed the JRAM-3 shunts to select one of the alternate sets of *EMS* control addresses (see Section 11), you must specify the alternate address set in the JPAGER.BIN line, according to table 61.1.1 below.

TABLE 61.1.1
JPAGER ALTERNATE CONTROL ADDRESS CODES

ALTERNATE SET	CONFIG.SYS LINE
SET 2	DEVICE=JPAGER.BIN /A=640
SET 3	DEVICE=JPAGER.BIN /A=576
SET 4	DEVICE=JPAGER.BIN /A=672

You can change both the swapping window and the *EMS* control addresses by simply listing both options in CONFIG.SYS. For example, you might use:

```
DEVICE=JPAGER.BIN /S=10B A=640
```

NOTE: If you also have JRAM or JRAM-2 boards in your computer, the total memory allocated to DOS and the RAM disks(s) must be equal to or greater than the total ordinary, JRAM, and JRAM-2 memory in your system. In other words, all memory not on a JRAM-3 board must be allocated before JPAGER.BIN in the CONFIG.SYS file.

6. Reboot the computer by pressing <Ctrl><Alt><Del>. This installs everything in CONFIG.SYS. There will be a pause as JBOOT.BIN measures how much memory is really installed in the computer. Then a message will be displayed listing the drivers you have installed and the amount of memory allocated to DOS and your RAM disk(s). Your expanded memory software will indicate how much memory there is for expanded memory. With Symphony 1.1, for example, look at the {services}s display.

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

SECTION SIXTY-TWO
BANK SWAPPING WINDOWS

62.1: ASSIGNING SWAPPING WINDOWS

Both the Tall Tree Systems bank switching standard and the Lotus/Intel/Microsoft expanded memory specification work by "swapping" blocks of memory in and out of "windows" or "pages" in the address space. These windows must be unoccupied by ordinary RAM or ROM. Moreover, conflicts may arise if more than one program attempts to swap in the same window. Tall Tree Systems products allow you to specify the address pages used by the software and hardware in order to make the most complete use of the address pages available in your computer.

To set up your system, look at table 62.1.1 below to determine what address pages are available in your computer with your peripherals. If you have an add-on hard disk or high-performance graphics board, consult its manual to see if it occupies any additional address pages. When you know what pages are free, set JDRIVE and JPAGER to use those pages as swapping windows.

JDRIVE and JPAGER can be set to use any address page for a swapping window by specifying the desired page in CONFIG.SYS. To make JPAGER swap in page 12 (C hex), for example, you would use the line:

```
  DEVICE=JPAGER.BIN /S=12B
```

This means "Swapping window equals Bank 12". The syntax is the same for JDRIVE; to make JDRIVE swap in page 9, for example, you would use the line:

```
  DEVICE=JDRIVE.BIN /S=9B
```

Both JDRIVE and JPAGER default to page 13 (D hex) if no swapping window is specified, since this is the only page that is available in virtually all systems. Note that JDRIVE and JPAGER must be set to use different swapping swapping windows if both are present in the same CONFIG.SYS file, and that both must be set to use windows that are not occupied by other hardware or software. Table 62.1.1 below summarizes the pages available in some common computers and with some common peripherals.

TABLE 62.1.1: AVAILABLE SWAPPING WINDOWS
Open spaces indicate pages available for swapping

Page	PC	XT	AT	Add-ons
15 (F hex)	ROM	ROM	ROM	JRAM cards 1-4
14 (E hex)			ROM	JRAM cards 5-8
13 (D hex)				Tallgrass hard disk
12 (C hex)		ROM	ROM	Bootable hard disk for PC
11 (B hex)	Monitor	Monitor	Monitor	
10 (A hex)				704K DOS with JBOOT JSPOOL buffer on RAM disk Tecmar Graphics Master board
9 (9 hex)				640K DOS
8 (8 hex)				576K DOS

You cannot change JSPOOL's RAM disk buffer from page 10 to a different swapping window. If you want to assign 704K to DOS or if you have a Tecmar Graphics Master board or a Tallgrass hard disk, you will have to use a JSPOOL buffer within DOS, which does not require a separate swapping window.

Some other graphics boards may also occupy page 10 (A hex); check the graphics board's manual. Any hard disk that you can boot up from occupies page 12 (C hex).

Most JRAM-2 boards shipped before August 1985 were set to use control addresses in page 14 (E hex). If you have one of these JRAM-2 boards and you want to use page 14 as a swapping window, you can make it available by setting the page select shunt on your JRAM-2 to indicate control addresses in page 15 (F hex). These control addresses are designed to occupy the same address pages as ROM in order to make more efficient use of the limited number of address pages available. Do not try to swap in page 14 without first changing the shunt on your JRAM-2 board, or unpredictable results may occur. See figure 2.1 in the JRAM-2 installation instructions for the location of the page select shunt.

Page 9 can be used for a swapping window if you have 576K or less of ordinary (non-JRAM) memory in your system and you allocate 576K or less to DOS. Page 8 can be used if you have 512K or less of ordinary memory and you allocate 512K or less to DOS.

SECTION SIXTY-THREE
EXPANDED MEMORY
PROGRAMMER'S INTERFACE

63.1 INTERRUPT 67h INTERFACE FOR PAGED MEMORY

The JRAM, JRAM-2, JRAM-AT, and JRAM-AT3 all use Tall Tree Systems established bankswitching standard to swap any 64K block of memory in and out of any 64K block of addresses. The JRAM-3 and JRAM-AT3 use not only "bank-switching" but also "page-switching", in which any 16K block of memory is swapped in and out of any of four contiguous 16K windows comprising a single 64K block of addresses which may be located at any round address segment boundary. In this section we call a 64K block of addresses a "bank", and a 16K block of addresses a "page".

Both of these memory-switching operations are accessed through assembly language utilities using the 67h interrupt.

63.1.1: EXPANDED MEMORY MANAGER (EMM)

The Expanded Memory Manager is an installable device driver that handles the memory switching operations for *EMS* software programs. Tall Tree Systems' *EMM* is JPAGER.BIN. The *EMM* is implemented as an interrupt service routine much like those available in the IBM PC ROM BIOS. The application program has to load the function number and the parameters desired in the appropriate registers and use the 67h interrupt.

The *EMM* provides one or more small "windows" of memory through which a larger memory space can be accessed. These small "windows" are the 16K pages. The *EMM* can manipulate four continguous 16K blocks to access up to 8 Mb of expanded memory space.

63.2 FRAMES OF REFERENCE AND TYPES OF MEMORY ACCESS

The computer's sixty-four 16K pages (or sixteen 64K banks) of direct address space can by considered from two frames of reference:

The PROGRAM FRAME consists of the entire sixty-four pages occupying the total address space of the computer (0-1024K). It is used by system programs such as DOS.

The APPLICATION FRAME consists of the six 64K pages located above the traditional DOS boundary, that is, above 640K.

The six pages within the APPLICATION FRAME (640K-1024K) can be used by application programs which are written to take advantage of this "expanded memory" by means of JRAM's page-switching.

The "page-switching" ability of the JRAM-3 and JRAM-AT3 is limited to 64K banks (or 4 contiguous 16K pages on 64K boundaries) within the APPLICATION FRAME. Future boards may extend "page-switching" to both frames of reference.

The APPLICATION FRAME allows **concurrent access.** It can be used concurrently by more than one application program, and handle the saving and restoring function of each mapping state.

For each frame of reference, there are two types of access: logical and physical. In logical mapping, the driver handles the mapping, turning on the requested memory page(s) at a particular location within a frame of reference. Physical mapping is performed by a calling program, which must obtain the mapping tables from the driver.

To avoid confusion, the JRAM-3 and JRAM-AT3 should be used only with the logical access method, because the memory-switching is handled by two different processes, i.e., page-switching for the APPLICATION FRAME and bank-switching for the PROGRAM FRAME. We would advise waiting to use physical mapping until the page-switching has been globalized to the entire board.

63.3 PHYSICAL ACCESS HARDWARE DEFINITION

The page hardware consists of a set of eight bit I/O ports for each page with a frame of reference. Each I/O port contains an **enable bit** (bit 7) and a **logical page selection bit** (bit 6-0). Thus, each I/O port can map up to 128 16K byte pages of two megabytes of total memory.

A group of I/O addresses consists of four I/O addresses for each page in an application frame or 16 I/O addresses for each page in a program frame.

The amount of memory in the computer can be placed on one or more memory expansion cards. Each paged memory card must contain at least one group of I/O addresses. Memory cards, which provide more than two megabytes total memory, must support I/0 addresses for more than one group.

All memory cards must be able to recognize all groups of I/O addresses--i.e., both their own I/O addresses and the addresses assigned to the other cards in the system. This is necessary so that when any output reference is made to turn on or off a particular page within a reference frame, all the non- selected cards must disable that particular page. That way the program does not have to worry about turning off pages.

EMM HANDLE and RE-ENTRANCY

An EMM handle is a value that the EMM assigns to a file or device. Previous versions of the expanded memory specification referred to an EMM handle as a Process ID.

The handle is used to test for the presence of the EMM.

The current version of the EMM is not *re-entrant*, which means that you may encounter a "busy" condition. If you do encounter the 82 return code, simply jump back and try again until it passes. Future versions of JPAGER.BIN will be "re-entrant", so that the "busy" condition will not occur.

63.4: OBTAINING ACCESS TO INTERFACE

THE "OPEN HANDLE" TECHNIQUE TO TEST FOR EMM

```
MOV      AX,3DOOH       ;DEVICE OPEN THE CALL/READ-ONLY MODE
MOV      DX,OFFSET DEVNAME
INT      21H
JC       NOACCESS
XCHG     AX,BX          ;DEVICE HANDLE FROM OPEN
MOV      AX,4407H       ;ISSUE IOCTL/STATUS REG
XOR      CX,CX          ;NO BYTES BACK
INT      21H
PUSHF
PUSH     AX
MOV      AH,3EH         ;DEVICE CLOSE CALL
INT      21H
POP      AX
POPF
JC       NOACCESS
CMP      AL, OFFH
JNE      NOACCESS
                        ;INT 67H OK HERE ON

DEVNAME DB 'EMMXXXX0',0
```

THE "GET INTERRUPT VECTOR" TECHNIQUE TO TEST FOR THE EMM

```
MOV      AX,3567H       ;GET INTERRUPT VECTOR 67H
INT      21H
MOV      DI,0AH         ;DEV NAME OFFSET
MOV      SI,OFFSET DEVNAME
MOV      CX,8
MOV      CLD
REPE     CMPSB
JNE      NOACCESS
                        ;INT 67H OK HERE ON

MOV      AH,40H         ;GET MEMORY STATUS
INT      67H
OR       AH,AH
JNZ      ERROR
```

```
MOV       AH,42H              ;GET TOTAL &
INT       67H                  UNALLOCATED PAGES
OR        AH,AH
JNZ       ERROR
MOV       [TOTPAGES],BX
MOV       [UNALOPAGES],DX

MOV       AH,43H              ;GET HANDLE
MOV       BX,[NUMPAGES]             ALLOCATE PAGES
INT       67H
OR        AH,AH
JNZ       ERROR
MOV       [HANDLE],DX

MOV       AH,45H              ;DEALLOCATE HANDLE
MOV       DX,[HANDLE]             ALLOCATED PAGES
INT       67H
OR        AH,AH
JNZ       ERROR

MOV       AH,46H              ;GET DRIVER VERSION NUMBER
INT       67H
OR        AH,AH
MOV       [VERNO],AL
```

63.5: LOGICAL ACCESS TO APPLICATION FRAME

```
MOV       AH,41H              ;GET SEGMENT ADDRESS OF
INT       67H                  APPLICATION FRAME
OR        AH,AH
JNZ       ERROR
MOV       [AFSEG],BX

MOV       AH,44H              ;MAP LOGICAL PAGE (I)
MOV       BX,[I]               AT PHYSICAL PAGE (J)
MOV       AL,[J]
MOV       DX,[HANDLE]
INT       67H
OR        AH,AH
JNZ       ERROR

MOV       AH,47H              ;SAVE MAPPING CONTEXT
MOV       DX,[HANDLE]            FOR HANDLE
INT       67H
OR        AH,AH
JNZ       ERROR

MOV       AH,48H              ;RESTORE MAPPING CONTEXT
MOV       DX,[HANDLE]            FOR HANDLE
INT       67H
OR        AH,AH
JNZ       ERROR
```

NOTE:*Physical Access is not implemented and no support is planned at this time. The following set is included for discussion with OS system programmers only. The only way supported at this time is to issue direct driver calls to JBOOT!*

63.6: PHYSICAL ACCESS TO APPLICATION FRAME

```
MOV     AH,49H          ;GET MAPPING PORT I/O PORT
LES     DI,PAGEMAP
INT     67H
OR      AH,AH
JNZ     ERROR
MOV     [BOARDS],AL

MOV     AH,4AH          ;GET LOGICAL TO PHYSICAL
LES     DI,LOGTOPHY                TABLE
MOV     DX,[HANDLE]
INT     67H
OR      AH,AH
JNZ     ERROR
```

63.7: LOGICAL ACCESS TO PROGRAM FRAME

```
MOV     AH,50H          ;GET FIRST DOS PHYSICAL
INT     67H             SWAPPING PAGE
OR      AH,AH
JNZ     ERROR
MOV     [PGMPAGES],AL
MOV     [FIRSTPAGE],BL

MOV     AH,51H          ;MAP LOGICAL PAGE (I) AT
MOV     BX,[I]          PHYSICAL PAGE (J)
MOV     AL,[J]
MOV     DX,[HANDLE]
INT     67H
OR      AH,AH
JNZ     ERROR
```

63.8: PHYSICAL ACCESS TO PROGRAM FRAME

```
MOV     AH,52H          ;GET MAPPING PORT I/O PORT
LES     DI,PAGEMAP                 ARRAY
INT     67H
OR      AH,AH
JNZ     ERROR
MOV     [BOARDS],AL

MOV     AH,53H          ;GET LOGICAL TO PHYSICAL
LES     DI,LOGTOPHY                TABLE
MOV     DX,[HANDLE]
INT     67H
OR      AH,AH
JNZ     ERROR
```

63.9: OPERATING SYSTEM SERVICE CALLS

```
MOV        AH,4BH            ;GET EMM HANDLE COUNT
INT        67H
OR         AH,AH
JNZ        ERROR
MOV        [TOTHANDLE],BX

MOV        AH,4CH            ;GET EMM HANDLE PAGES
MOV        DX,[HANDLE]
INT        67H
OR         AH,AH
JNZ        ERROR
MOV        [HANDLEPAGECNT],BX

MOV        AH,4DH            ;GET ALL EMM HANDLE PAGES
LES        DI,HANDLEPAGEMAP
INT        67H
OR         AH,AH
JNZ        ERROR
MOV        [TOTHANDLE],BX

MOV        AX,4E00H          ;GET PAGE MAPPING REGS
LES        DI,PAGEMAPREGS
INT        67H
OR         AH,AH
JNZ        ERROR

MOV        AX,4E01H          ;SET PAGE MAPPING REGS
LES        SI,PAGEMAPREGS
INT        67H
OR         AH,AH
JNZ        ERROR

MOV        AX,4E02H          ;GET AND SET PAGE MAPPING REGS
LES        DI,DESTINATIONPAGEMAPREG
LDS        SI,SOURCEPAGEMAPREGS
INT        67H
OR         AH,AH
JNZ        ERROR

MOV        AX,4E03H          ;GET SIZE OF PAGE MAP ARRAY
INT        67H
OR         AH,AH
JNZ        ERROR
MOV        [SIZEOFARRAY],AL
```

63.10: ERROR RETURN CODES FOR INTERRUPT 67H CALLS

AH = 0 Success

AH = 80H Driver software malfunction

AH = 81H Driver hardware malfunction

AH = 82h Driver is "busy" (RECOVERABLE)

AH = 83H Bad process id

AH = 84H Bad driver function code

AH = 85H No process id's are available

AH = 86H Save or Restore page mapping context error

AH = 87H Not enough pages available (RECOVERABLE)

AH = 88H Not enough pages un-allocated
(RECOVERABLE)

AH = 89H Zero pages cannot be allocated
(RECOVERABLE)

AH = 8AH Page requested out of allocated range
(RECOVERABLE)

AH = 8BH Page within frame is out of range
(RECOVERABLE)

AH = 8CH Mapping registers save area is full

AH = 8DH Mapping registers already saved
(RECOVERABLE)

AH = 8EH Mapping registers not saved
(RECOVERABLE)

AH = 8FH Subfunction Undefined

63.11: PROGRAMMING NOTE

The source code for JPAGER and JBOOT is provided on the diskette for your perusal so that you can adapt your program calls accordingly. Note: Application programs must intercept <CTRL><ALT><DEL> and call JBOOT if they are using the logical access method. When a program terminates, it must reboot so that JBOOT can free the banks of memory that were taken from the available memory by the application software.

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.

HARDWARE LIMITED WARRANTY

To the original purchaser only, Tall Tree Systems warrants its products to be free from defects in materials and workmanship for a period of one year for fully assembled JRAM boards and 30 days for bare boards from date of purchase from Tall Tree Systems or an authorized dealer. Tall Tree Systems does not warrant that RAM chips bought by you will work in the board.

In the event of malfunction, or any other indication of failure attributable directly to faulty workmanship and/or materials, call Tall Tree Systems at (415) 964-1980 for a Returned Materials Authorization (RMA) number, then mail the board to:

Tall Tree Systems
1120 San Antonio Rd.
Palo Alto, Ca. 94303
USA
Attention: Warranty Claims

SALES + CUSTOMER SERVICE
415-~~964-1980~~
493-1980
TECH SUPPORT
415-~~964-1397~~
424-8324

Tall Tree Systems will repair and/or replace defective parts and return the product in good operating condition.

This warranty will not cover any Tall Tree Systems product failure which, in Tall Tree Systems' judgement, has resulted from accident, abuse, negligence, alteration, or misapplication of the product.

This warranty is void if a serial-serial module is plugged into a JRAM board with a female 25-pin connector, or if a serial-parallel- clock/calendar module is plugged into a JRAM board with a male 25- pin connector.

This warranty is in lieu of all other warranties, expressed or implied, including warranties or merchantability and fitness for use. In no event will Tall Tree Systems be liable for incidental and consequential damages arising from or in any way connected with the use of its products.

Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation may not apply to you.

JBOOT	3.41
JPAGER	1.20
JOUTPUT	1.00
JDRIVE	3.21
JSPOOL	2.10

#080257