RAMD USER'S MANUAL

PICKLES

TROUT ®

RAMD

USER'S MANUAL

Published by
Pickles & Trout
P.O. Box 1206
Goleta, California 93116
U.S.A.

10 9 8 7 6 5 4 3 2

DISCLAIMER

The publisher has made a reasonable effort to insure that the computer programs described herein are correct and operate properly and that the information presented in this publication is accurate; however they are sold and licensed without warranties either expressed or implied, including, but not limited to, the implied warranties of merchantibility and fitness for a particular purpose. The publisher is not liable for consequential damages resulting from the use of this product either individually or in concert with other computer programs. Further, the publisher reserves the right to revise this publication and the programs described herein and to make changes from time to time in the contents thereof without obligation of the publisher to notify any person or organization of such revision or changes.

TABLE OF CONTENTS

Topics Page

INTRODUCTORY NOTES

1.1 Program Requirements 1.1
1.2 Registration & Support 1.1
1.3 Manual Notation 1.1
1.4 Copying the Distribution Diskette 1.1
1.5 Basic Concepts 1.2
1.6 Configuring CP/M for Use With RAMD 1.3

EXECUTING RAMD

2.1 Introduction 2.1
2.2 RAMD Command Line Options 2.1
2.3 Example Command Lines 2.3
2.4 Command Line Options When Re-executing RAMD 2.4

USING RAMD

3.1 Introduction 3.1
3.2 Writing SUBMIT Files 3.1
3.3 The WAIT Command 3.4

TECHNICAL NOTES

4.1 Allocation Blocks and Directory Entries 4.1
4.2 RAMD and Other Programs that Reside above the CP/M 4.1

ERROR MESSAGES 5.1

1.1 Program Requirements

RAMD will work on any Radio Shack Model II/12/16 that has a Motorola 68000 microprocessor installed in it.

1.2 Registration & Support

Fill out the registration card that comes with RAMD and return it to Pickles & Trout at your earliest convenience. When this card is in our files, Pickles & Trout can support RAMD in two ways. First, our telephone support personnel may be called Monday - Friday from 9 a.m. to 5 p.m. (Pacific Coast Time). Second, if the RAMD master diskette is damaged, it may be returned to Pickles & Trout to be regenerated. The service charge for this regeneration is $18 (including handling/shipping) if you live in the USA/CANADA or $25 if you live elsewhere.

1.3 Manual Notation

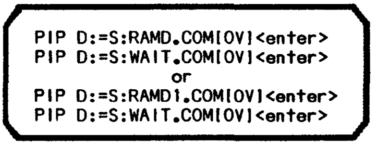
When named keys are mentioned in this documentation, the name of the key is enclosed in angle brackets (e.g., <enter>, <esc>). The "image screens" used throughout the manual attempt to depict exactly what will appear on your screen, with one addition: we have added line numbers to make it easier to describe the screens. A reference to lines 4.1-4 means that you should examine Figure 4, lines 1 through 4.

1.4 Copying the Distribution Diskette

RAMD is distributed on a P&T double-density 8" diskette. When you open the package in which this diskette is sealed, please do the following:

1. Boot up your P&T CP/M 2 operating system and mount the RAMD diskette on drive 0 (if you have a single-drive system) or drive 1 (if you have two drives). Then type "DIR S:" after the "A>" prompt, replacing the "S" with the letter of the secondary floppy disk logical drive (e.g., "B" on a typical floppy only system). As you can see, there are three files on this diskette: RAMD.COM, RAMD1.COM, and WAIT.COM. If you have a CP/M 2.2m system and are using the M16WB1 or R16WB1 warm boot module, you will want to use the RAMD1.COM file; otherwise, you should use RAMD.COM. (Note: If you are not sure which warm boot module you are using, execute MODSEL from the "A>" prompt, following the directions in Chapter 6 of your P&T CP/M 2 User's Manual. It will allow you to view the modules that are currently selected.) WAIT.COM is a useful utility to use with the SUBMIT utility.

2. When you know which version of RAMD you wish to use, copy this program to a working system diskette/hard disk logical drive by executing one of the two sets of commands depicted in Figure 1. When executing these commands, replace "D" with the letter of the destination logical drive and replace "S" with the letter of the source logical drive.



```
PIP D:=S:RAMD.COM[OV]<enter>
PIP D:=S:WAIT.COM[OV]<enter>
             or
PIP D:=S:RAMD1.COM[OV]<enter>
PIP D:=S:WAIT.COM[OV]<enter>
```

Figure 1 PIP Commands to Backup RAMD

3. Store the master P&T RAMD diskette with your other master diskettes.

4. If you cannot copy the RAMD master diskette or the program does not work properly, try performing the operation on a second computer (if possible) before telephoning Pickles & Trout. If it fails on two machines, call us for assistance. Please write down any error messages before calling.

1.5 Basic Concepts

A Model 16 (or an upgraded Model II/12) has two microprocessors, a Zilog Z80 with 64 Kbytes of RAM memory and a Motorola 68000 with 128-768 Kbytes of RAM memory. Since the average CP/M-80 program cannot access the 68000 chip and its memory, most CP/M-80 users have never been able to take full advantage of their Model 16. RAMD is designed to change this state of affairs.

RAMD is a rather unusual CP/M-80 program. Like most CP/M-80 programs, it is loaded into Z80 memory when executed and is controlled by the Z80 microprocessor during execution. Unlike most CP/M-80 programs, however, RAMD can directly access the 68000's RAM memory and continue to execute properly while another CP/M-80 program is running. RAMD's usefulness derives from these special capabilities.

While RAMD is running, any other CP/M-80 program can use the 68000's RAM memory in much the same way that it normally uses a floppy or hard disk logical drive. For all practical purposes, RAMD thus adds another logical drive to your CP/M system. This "RAM drive" has one major advantage over all other drives: since data can be transferred between Z80 and 68000 memory much faster than it can between Z80 memory and a disk drive, CP/M-80 programs that utilitize the RAM drive can commonly execute much faster than would otherwise be possible. Under the right circumstances, the savings in time and money can be dramatic.

RAMD has its greatest impact on the speed of program execution when the following criteria are fulfilled:

* The program is I/O intensive, meaning that it is constantly reading from and/or writing to a disk drive.
* A program and its data file(s) can all be copied to the RAM drive (meaning that the more 68000 memory you have the more useful RAMD can be).

RAMD's possible speed advantage is clearly indicated by comparing the times required to perform two common Vedit operations on a hard disk drive and RAMD drive (Figure 2).

	Hard Disk	RAM Drive
Load File	5.95 sec	1.25 sec
Write/Reload File (using EA)	22.00 sec	5.51 sec

Figure 2 Example Program Execution Speed Using RAMD

The RAM drive is unlike other logical drives in one key respect: it can only store data while the computer is turned on and receiving power. Thus, you must back up any data files on the RAM drive before shutting off your computer. The implications of this will be discussed further below.

1.6 Configuring CP/M For Use With RAMD

Since RAMD requires 512 bytes of Z80 memory above the operating system, your P&T CP/M 2 operating system must be lowered at least 512 bytes to make space for it. This should be done in one of two ways, depending on which version of P&T CP/M 2 you use. (Note: If you are running RAMD with another program that also resides above the operating system, see Section 4.1 below for further instructions.)

1. If you are using a copy of P&T CP/M 2.2m (with a green and white user's manual), lower the operating system by executing the LA option in the System Configuration portion of the P&T MENU (p. 4.29 in the P&T CP/M 2 User's Manual). When prompted, say that the present last address (probably FFFF hex) is not correct. Then, when asked what it should be, set the last address available to CP/M to FDFF or lower.

2. If you have a P&T CP/M 2.2e operating system (with a blue user's manual), execute RESIZER and specify, when prompted, that you wish to configure a 63.5 Kbyte (or smaller) operating system. (Please remember: when you generate a diskette with RESIZER, you may have to use SETMISC and IOFREEZE to set, for instance, the correct printer parameters.)

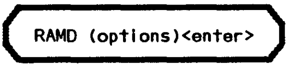
You must RESET your computer from the altered diskette before executing RAMD for the first time.

Note: Since the performance of most CP/M programs should not be significantly affected by lowering the operating system 512 bytes, they can be executed with versions of P&T CP/M 2 configured to work with RAMD. A few programs, however, may be adversely affected by this change. If you have such a program, you should consider altering your procedures slightly so that it is only executed with a copy of P&T CP/M 2 that has not been configured to work with RAMD.

NOTES

2.1 Introduction

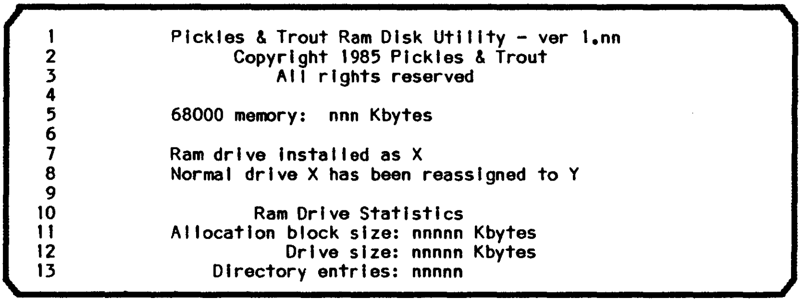
RAMD is executed by typing its name, followed by a space and the RAMD options (discussed below) that you wish to utilize. The options may be separated by one or more non-alphanumeric characters (e.g., comma, semicolon, space), but separators are not required. Each of these options has a default value (which may vary with the amount of 68000 memory available) so they do not have to be specified in many instances. The form of the command line for executing RAMD is illustrated in Figure 3.



```
RAMD (options)<enter>
```

Figure 3 Prototype RAMD Command Line

When a command line is executed, RAMD checks to ensure that your 68000 memory will support the options you have specified. Then, if everything checks out correctly, it displays a summary (Figure 4) that indicates the amount of 68000 memory (Line 4.5), the assignment of logical drive letters (Lines 4.7-8), allocation block size (Line 4.11), the amount of memory available to the RAM drive (Line 4.12), and the number of directory entries assigned to the RAM drive (Line 4.13). (Note: Line 4.8 will only appear if you have used option S to re-assign drive letters.)



```
1            Pickles & Trout Ram Disk Utility - ver 1.nn
2                   Copyright 1985 Pickles & Trout
3                        All rights reserved
4
5            68000 memory:  nnn Kbytes
6
7            Ram drive installed as X
8            Normal drive X has been reassigned to Y
9
10                     Ram Drive Statistics
11           Allocation block size: nnnnn Kbytes
12                      Drive size: nnnnn Kbytes
13               Directory entries: nnnnn
```

Figure 4 Initial RAMD Summary of Conditions

2.2 RAMD Command Line Options

Figure 5 provides a summary of the options that may be used with RAMD. Each is described in more detail below.

Ennn	Set the number of directory entries to be used on the RAM drive
Hnnn	Reserve 68000 memory above the RAM drive
Lnnn	Reserve 68000 memory below the RAM drive
T	Test 68000 memory
Dx	Set logical drive letter assigned to RAM drive
Sx	Change the logical drive letter assigned to a normal logical drive
R	Preserve the files on the RAM drive
X	Erase the RAM drive

Figure 5 Summary of RAMD Command Line Options

Please note: do not leave a space between an option letter and the letter or digits that follow(s) it.

Ennn This option allows you to set the number of directory entries available to the RAM drive to nnn, where nnn may not be less than 32 nor more than 512. The number of directory entries defaults to 64, 128, 256, or 512, depending on the amount of 68000 memory available to the RAM drive (see Figure 15 below). Please note:

* Unless you have a large amount of 68000 memory or you intend to process a very large number of small files, the default number of directory entries is normally adequate.
* If you have 128-256 Kbytes of RAM memory, specifying 32 directory entries gives you 1 Kbyte more memory for the RAM drive than the default of 64.

Hnnn The Hnnn option causes RAMD to reserve nnn blocks of memory at the high end of the 68000 memory. Each block consists of 16 Kbytes of memory. Please remember: if you are using RAMD1, you do not have to use this option to reserve memory for the M16WB1 and R16WB1 warm boot modules. However, if you wish to reserve more than 16 Kbytes when using RAMD1, you may use this option to do so.

Lnnn The Lnnn option causes RAMD to reserve nnn (16 Kbyte) blocks of memory at the low end of 68000 memory. Thus, if you are using Trisoft's CP/M 68K operating system and you have configured it to use 256 Kbytes of memory, you should reserve 16 blocks of memory by including 'L16' on the RAMD command line.

T The T option performs a non-destructive test of the 68000 memory that will be used by the RAM drive and reports the presence of any bad memory cells. If bad cells are detected, the RAM drive is not loaded and an error is reported. Note that the T option does not test areas of 68000 memory reserved by options H and L. Thus, if you wish to test all of your 68000 memory, use the command "RAMD T<enter>" before executing a command line that preserves blocks of memory.

The next two options (D and S) are related, in that they are both concerned with the assignment of logical drive letters. Read the two descriptions together.

Dx The D option assigns logical drive x to the RAM drive (where x is a letter in the range A - P). If this logical drive letter is already assigned to a physical floppy or hard disk logical drive, the latter will become inaccessable unless you use option S (described below) to assign a new letter to it. If option D does not appear on the command line, the RAM drive is made logical drive M.

Sx When the RAM drive is installed, it replaces any existing logical drive which has the same letter as that assigned to the RAM drive. You can use the S option to assign another letter to the existing drive so that you do not lose access to it. For instance, if you use option D to assign logical drive A to the RAM drive, the drive normally called drive A can be re-named drive D (for instance) by typing SD on the RAMD command line. (Note: if there already is a drive D, it will become inaccessible.) Similarly, if drive M (the default) is already assigned to a hard disk, this option allows you to assign a different letter to that hard disk logical drive.

The last two options (R and X) are related, in that they are both concerned with the state of the RAM drive after you have executed RAMD. The two descriptions should be read together.

R The R option insures that any files which have been moved to the RAM drive are preserved when you re-execute RAMD. You only have to use this option in one situation: when you are forced to RESET your computer (thereby destroying the copy of RAMD in Z80 memory) and you wish to preserve the files that are stored on the RAM drive. It is not necessary to use this option when you are merely re-executing RAMD, because RAMD will not erase the RAM drive if it finds a copy of RAMD in Z80 memory.

X The X option insures that the files on the RAM drive are erased when you execute RAMD. The files will be erased even when a copy of RAMD is still in memory (i.e., even when RAMD would normally not erase them). This is useful when you wish to the erase files created by one task before starting a new one.

Please note: There is no reason to specify both R and X on the same command line. If you do so, however, the contents of the RAM drive will be preserved if RAMD is not installed in memory and erased if it is installed.

2.3 Example Command Lines

RAMD

Execute RAMD and use the following default values: reserve space for at least 64 directory entries (depending on the amount of 68000 memory), reserve no memory above or below the RAM drive, erase the RAM drive unless RAMD is already in memory, and assign logical drive M to the RAM drive.

RAMD1

Same as RAMD, except that this version of RAMD automatically reserves 1 (16 Kbyte) block above the RAM drive for the M16WB1 or R16WB1 warm boot module.

RAMD T DC SD

Test all of the 68000 memory and, if no bad cells are found, load the RAMD program. Assign C to the RAM drive. Assign D to the disk drive that had previously been designated drive C.

RAMD X E128 H2 T

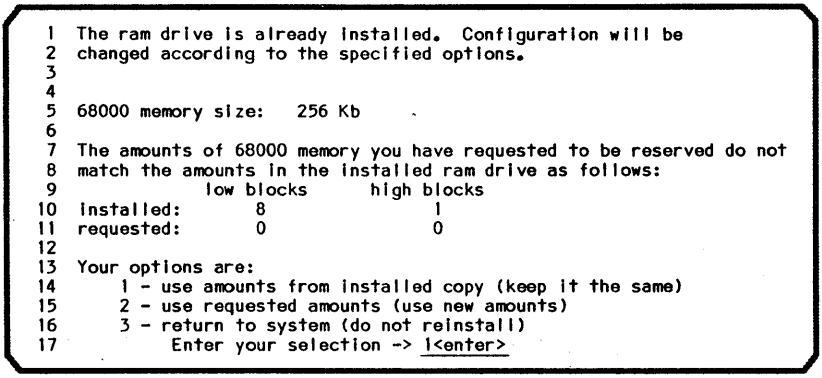
Load RAMD, erase your 68000 memory, assign 128 directory entries to the RAM drive, reserve the top 32 (2 X 16) Kbytes of 68000 memory, and test all but this top 32 Kbytes of memory.

RAMD1 L8 R

Load a version of RAMD that automatically reserves 1 (16 Kbyte) block of memory above the RAM drive. Reserve 8 (16 Kbyte) blocks of memory below the RAM drive (e.g., for a 128 Kbyte Trisoft CP/M 68K system) and ensure that the RAM drive is not erased.

2.4 Command Line Options When Re-executing RAMD

If you execute RAMD with a copy already in memory, it will display the messages shown on lines 6.1-5. As this suggests, RAMD's command options may be altered each time you re-execute the program, but some care must be taken. There are essentially two possibilities. If you are starting a new task and using the X option to erase the existing files, all of the options may be changed. On the other hand, if you are re-executing RAMD to work with files that are already stored on the RAM drive, you must not use option X nor alter options E, H, or L. To insure that you do not make a serious mistake, RAMD will check the copy in memory to determine if you have reserved the same number of blocks above and below the RAM drive. If there has been a change, RAMD will warn you of this and allow you to (1) use the parameters previously specified, (2) use the newly-requested parameters, or (3) return to the system to re-enter the parameters (lines 6.7-16). Merely type in the number of your selection, followed by <enter> (line 6.17).



```
 1  The ram drive is already installed.  Configuration will be
 2  changed according to the specified options.
 3
 4
 5  68000 memory size:   256 Kb
 6
 7  The amounts of 68000 memory you have requested to be reserved do not
 8  match the amounts in the installed ram drive as follows:
 9               low blocks       high blocks
10  installed:       8                1
11  requested:       0                0
12
13  Your options are:
14      1 - use amounts from installed copy (keep it the same)
15      2 - use requested amounts (use new amounts)
16      3 - return to system (do not reinstall)
17           Enter your selection -> 1<enter>
```

Figure 6 Warning Message When Re-executing RAMD

3.1 Introduction

There are typically four steps involved in using RAMD: executing RAMD, moving files to the RAM drive, executing programs from the RAM drive, and backing up the RAM drive. Chapter 2 discusses the first of these steps. This chapter discusses (a) the last three steps and (b) the integration of the process using CP/M's SUBMIT capability.

Moving files. Once you have executed RAMD, your 68000 memory is the rough equivalent of a normal CP/M floppy drive or hard disk logical drive. Consequently, before executing a program from the RAM drive, you must copy all of the files you wish to use to it.

Note: RAMD is not a buffering program, meaning that it will not read specific sections of a large data file into memory as they are needed by a program. Rather, as with normal drives, the RAM drive must be large enough to hold an entire file or the file cannot be processed from it.

Executing Programs. Once a program is on the RAM drive, you can execute it in the same way that you execute it on any other drive. Thus, you may wish to log on to the RAM drive, by typing X:<enter> (where X is the logical drive letter assigned to the RAM drive), before executing a program. If a program uses overlay files (as WordStar does, for instance), you must copy both the program and its overlay files to the RAM drive and log on to the RAM drive before executing the program.

Backing up. Although the RAM drive is equivalent to a normal drive, it is unique in one key respect: you can only store files on it while your computer is turned on and receiving power. Therefore, at the end of a work session, you must copy your data files back to a regular floppy or hard disk logical drive.

WARNING: Since a power failure will destroy any data files on the RAM drive, you should make it a habit to copy data files to your normal storage media at regular intervals.

Executing these steps can be very tedious, especially if they have to be repeated several times per day. The solution is to write a SUBMIT file for each of the operations that you intend to perform regularly with RAMD. The examples discussed in Section 3.2 indicate how such files are written.

3.2 Writing SUBMIT Files

The SUBMIT and XSUB utilities are discussed on pp. 8.137-139 and 8.151-153 in your P&T CP/M 2 User's Manual. Briefly stated, these utilities allow you to execute a sequence of several commands (which would normally be entered one at a time) by typing just one command, a SUBMIT command line (Figure 6). To have this capability, you must first create a text file that is composed of the commands you wish to execute. This file must have a "SUB" extension (i.e., it must be called something like WSTAR.SUB). It is normally stored on logical drive A, since you must be logged on to drive A when you execute SUBMIT. The following discussion describes how to write such files.

The following instructions are written on the assumption that you are using VEDIT, the full-screen text editor that comes with P&T CP/M 2.2m. You may generally use any text editor that produces standard ASCII files (e.g., WordStar in its non-document mode).

If you wish to create a "SUB" file to use with SUBMIT, follow these three basic steps:

1. Type in VEDIT FILENAME.SUB<enter> at the "A>" prompt (replacing FILENAME with your name for the SUB file).

2. When the VEDIT screen appears, type in the command lines you need (pressing <enter> at the end of each). If you were creating one of the three SUB files shown in Figures 8-10, the command lines depicted would be typed at this stage.

3. You must not have a blank line embedded in or at the bottom of a SUB file. Thus, when the cursor is immediately below the first character of the last command in the file, press <esc><esc> to exit to the VEDIT command prompt (*). Then type EX<esc><esc> to write the new file to your disk and return to the "A>" prompt.

Whenever you wish to execute the commands in this file, just type a command line like that shown in Figure 7 at the "A>" prompt. Please note that the ".SUB" extension is not part of this command line. (See the example in Figure 11 below.)

```
SUBMIT filename (optional parameters)<enter>
```

Figure 7 Prototype SUBMIT Command Line

Example 1: Copying Files to the RAM Drive

The first example (Figure 8) depicts a SUBMIT file that prepares the system to edit some files using Vedit. It performs three operations: (a) executes RAMD with all of the default parameters selected (line 8.1), (b) uses XSUB and PIP to copy several specific files from logical drive A to the RAM drive (logical drive M) (lines 8.4-7), and (c) logs on to the RAM drive (line 8.9). This file then leaves it to the user to execute VEDIT and backup the RAM drive at the end of the session.

```
1         RAMD<enter>
2         XSUB<enter>
3         PIP<enter>
4         M:=PGM1.BAS[V]<enter>
5         M:=PGM2.BAS[V]<enter>
6         M:=PGM3.BAS[V]<enter>
7         M:=VEDIT.COM[OV]<enter>
8         .<enter>
9         M:<enter>
```

Figure 8 Using XSUB and PIP to Copy Files

Note: the XSUB command (line 8.2) used here can save considerable execution time because it allows you to copy several distinct files to the RAM drive after just one execution of the PIP utility. By contrast, if XSUB were not used, the commands depicted on line 8.4-7 would each have to be preceded by PIP and the operating system would have to re-load PIP into memory before each copy operation.

Example 2: RAMD and a Programming Language

The second example (Figure 9) depicts a SUBMIT file that executes RAMD (with all of its default values set) and then copies Turbo Pascal system files to the RAM drive (line 9.2). It assumes that the files to be placed on the RAM drive are on drive B and that the RAM drive is drive M (the default). It then logs on to drive M (line 9.3) and executes Turbo Pascal (line 9.4). After you have finished using Turbo, it re-logs on to drive A (line 9.5) and PIP's all of the source code files that Turbo created to drive B (line 9.6).

```
1        RAMD<enter>
2        PIP M:=B:TURBO*.*[OV]<enter>
3        M:<enter>
4        TURBO<enter>
5        A:<enter>
6        PIP B:=M:*.PAS[OV]<enter>
```

Figure 9 A Programming Language SUBMIT File

Example 3: RAMD and WordStar

The third example (Figures 10a & 10b) sets up two SUBMIT files for a typical word processing application. The second of these files (Figure 10b) is somewhat more complicated - but also considerably more flexible - because it includes two symbols ($1, $2) which are replaced by "parameters" that you must specify each time you execute a SUBMIT command line to process it.

```
1        RAMD<enter>
2        PIP M:=A:WS*.*[OV]<enter>
```

Figure 10a WordStar SUBMIT File #1

```
1        PIP M:=$2:$1[V]<enter>
2        M:<enter>
3        WS<enter>
4        A:<enter>
5        PIP $2:=M:$2[V]<enter>
6        (optional ERA command here)
```

Figure 10b WordStar SUBMIT File #2

The first SUBMIT file is designed to be used at the beginning of a WordStar session. It performs two operations: loads RAMD (line 10a.1) and uses PIP to copy your WordStar files from drive A to the RAM drive (drive M) (line 10a.2).

The second SUBMIT file assumes that you have already executed the first SUBMIT file. It is executed each time you wish to begin a new task involving the use of different Wordstar text files. This SUBMIT file performs five operations: (a) allows you to copy one or more specified text files ($1) from a specified source drive ($2) to the RAM drive (line 10b.1); (b) makes the RAM drive the logged-on drive (line 10b.2); (c) executes WordStar (line 10b.3); (d) re-logs on to logical drive A (line 10b.4); and (e) then copies the altered file(s) back to the original source drive ($2) (line 10b.5). Note: if you wished to erase the text file(s) on the RAM drive at the end of each task, you could add the command "ERA M:$1<enter>" to the bottom of the SUB file.

```
SUBMIT WS1<enter>
SUBMIT WS FILE.TXT B<enter>
```

Figure 11 Example SUBMIT Command Line (with WordStar)

If you called these two SUBMIT file STARTWS.SUB and WS.SUB, you could process them with the command lines shown in Figure 11. The second of these commands requires a little explanation, because it includes two parameters, the name of a file and a logical drive letter, which SUBMIT uses to modify WS.SUB. When you execute this line, SUBMIT replaces $1 with FILE.TXT and $2 with B wherever these symbols appear in WS.SUB. When this modified WS.SUB is processed, therefore, it (a) copies FILE.TXT from logical drive B to the RAM drive; (b) executes WordStar from the RAM drive; and (c) copies the new FILE.TXT file back to logical drive B. (Note: The parameter associated with $1 in this example could be replaced with a wildcard name such as "*.TXT".)

Note: If you use WordStar's "R" option, it will over-write any temporary SUB file on drive A, thus preventing WS.SUB from completing its full cycle of commands.

3.3 The WAIT Command

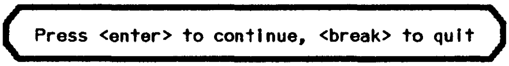
When embedded within a SUB file, the WAIT utility allows you to pause during the processing of the file or abort the processing of the file. Both capabilities can be extremely useful.

To use WAIT, simply add the line shown in Figure 12 at any place in a SUB file where you wish to pause or have the option of aborting the execution of the file.

```
WAIT<enter>
```

Figure 12 WAIT Command Line

When the processing of the file reaches a WAIT command, it displays the message shown in Figure 13.

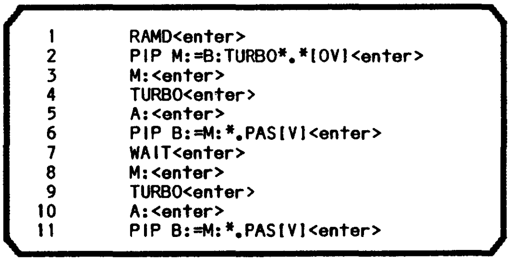


```
Press <enter> to continue, <break> to quit
```

Figure 13 WAIT Prompt

The system will then wait for input from the keyboard. If you press <enter>, the processing of the file will continue. If you press <break>, any SUBMIT file in progress will be aborted.

Figure 14 shows another version of the SUB file depicted in Figure 9. In this case, the use of WAIT (line 14.7) allows you to exit Turbo Pascal, backup the program files, and then either re-execute Turbo Pascal or abort the execution of the SUB file. Note that you may have to specify on which logical drive WAIT is to be found (e.g., A:WAIT<enter>).



```
 1        RAMD<enter>
 2        PIP M:=B:TURBO*.*[OV]<enter>
 3        M:<enter>
 4        TURBO<enter>
 5        A:<enter>
 6        PIP B:=M:*.PAS[V]<enter>
 7        WAIT<enter>
 8        M:<enter>
 9        TURBO<enter>
10        A:<enter>
11        PIP B:=M:*.PAS[V]<enter>
```

Figure 14 SUBMIT File Examplifying The Use of WAIT

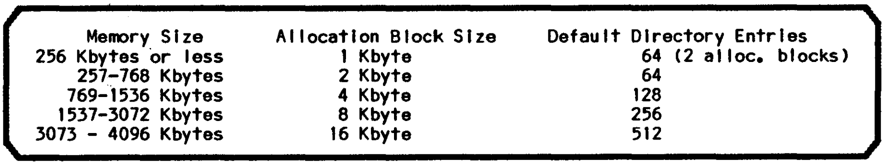
NOTES

4.1 Allocation Blocks and Directory Entries

RAMD automatically chooses an allocation block size for the RAM drive based on the amount of 68000 memory available to it. Since space is allocated to files in allocation block units (i.e., a 1 byte file will be allocated an entire allocation block even though it is much smaller than the block), RAMD chooses the smallest allocation block size possible for the RAM drive. This is controlled by the size of certain data areas inside the RAMD program.

At least one allocation block is always reserved for the disk directory. The default number of directory entries used by RAMD is typically the number that will fit in a single block. (The exception is a drive with 1 Kbyte allocation blocks. Here, two blocks are used by default.) Since 32 directory entries will fit in 1 Kbyte, a 2 Kbyte allocation block will hold 64 entries, a 4 Kbyte allocation block will hold 128 entries, etc.

Figure 15 shows the allocation block size and default number of directory entries chosen by RAMD for the various different amounts of 68000 memory available to the RAM drive.



Memory Size	Allocation Block Size	Default Directory Entries
256 Kbytes or less	1 Kbyte	64 (2 alloc. blocks)
257-768 Kbytes	2 Kbyte	64
769-1536 Kbytes	4 Kbyte	128
1537-3072 Kbytes	8 Kbyte	256
3073 - 4096 Kbytes	16 Kbyte	512

Figure 15 Allocation Blocks and Directory Entries

4.2 RAMD and Other Programs That Reside Above CP/M

If you have other programs (like CLINK from Corvus) that reside above the operating system, and you intend to run these program(s) and RAMD concurrently, note the following:

- Normally, you should execute RAMD after you have executed the other program(s) that reside above the CP/M. This is especially important when you are running a program (like CLINK) that alters the input/output operations of the BIOS. Loading RAMD first in this situation may cause the system to crash.

- You must lower the operating system far enough to accommodate both RAMD and the other program(s) that reside above the operating system. In general, if you are adding RAMD to an already functioning system diskette, you only have to move the operating system 512 bytes (i.e., 1/2 Kbyte) lower than it is at present. In some instances, however, the space required by two or more programs may be greater than the sum of the space required by each of them. Thus, if the system fails when you first execute RAMD, this may indicate that you have not yet lowered the operating system far enough to accommodate your programs. The only way to determine if this is the case is to experiment some (e.g., by lowering the operating system in 256 byte increments until the programs do work).

68000 memory error detected at nnnnnn hex, wrote nn, read 00.
Error reported by the T (memory test) option. Write down the location of the error and call a Radio Shack dealer for assistance.

Cannot have more than 512 directory entries
You have typed a number larger than 512 after the E option. This is not allowed.

Cannot locate 68000 memory. Either the 68000 board set is not installed or you have a hardware problem.
This message probably reflects a hardware problem. However, since there is a slight possibility that your RAMD program has been damaged, try it on a second computer.

Error in options
The E, H, L, D, or S option (pointed to by " ^ ") is followed by an inappropriate value. Retype the command line.

Must have a number here
Something other than a number appears after the E, H, or L option (pointed to by " ^ "). Retype the command line.

Please respond with 1, 2, or 3 only.
You have selected an incorrect option. Please select one of those specified.

Ram drive not installed due to errors detected in 68000 memory.
Error reported by the T (memory test) option.

The number of directory entries you have specified uses up all of the available memory in the ram drive. You must specify fewer directory entries (64 is usually sufficient).

There is not enough free space above the operating system to install the ram drive software. You must move the system down by at least 512 bytes (last address = FDFF) before using the ram drive.
Your CP/M operating system is not properly configured. See Section 1.6 in this manual for instructions.

This is an invalid drive letter.
You have specified a drive letter outside of the acceptable A - P range after option D or S (pointed to by " ^ "). Retype the command line.

Unexpected character in options
The command line option (pointed to by " ^ ") is not E, H, L, R, X, T, D, or S. Retype the command line.

You have attempted to reserve more memory than is available for ram drive use, leaving none for the ram drive.
Option H and/or L has reserved too many 16 Kbyte blocks. Retype the command line and reserve fewer blocks.

NOTES

PICKLES & TROUT®

P.O. BOX 1206, GOLETA, CA 93116