

# PLAN 4000™

File Server  
Apple II Pascal User's Guide

**NESTAR**

---

NESTAR SYSTEMS, INCORPORATED

PLAN 4000 (TM)

FILE SERVER

APPLE // PASCAL  
USER'S GUIDE

## DISCLAIMER OF WARRANTIES

The information in this document has been carefully reviewed and is believed to be reliable; nevertheless, Nestar Systems, Incorporated makes no warranties, either express or implied, with respect to this manual or with respect to the software described in this manual, its quality, performance, merchantability, or fitness for any particular purpose. The entire risk as to its quality and performance is with the buyer. The software herein is transferred "AS IS".

Nestar Systems, Incorporated, reserves the right to make changes to any products described herein to improve their functioning or design. Nestar programs are licensed products whose use is defined in the Nestar System Software License Agreement. Server Programs usually require a separate license for each machine on which they run.

In no event will Nestar Systems, Incorporated be liable for direct, indirect, incidental, or consequential damages at law or in equity resulting from any defect in the software, even if Nestar Systems, Incorporated has been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you.

The Regents of the University of California, Apple Computer, Inc. and International Business Machines, Inc., make no warranties, either express or implied, regarding the computer software programs described herein, their merchantability or their fitness for any particular purpose. UCSD, UCSD Pascal, and UCSD p-System, are all trademarks of the Regents of the University of California. CP/M is a registered trademark of Digital Research, Inc. Z-80 is a registered trademark of Zilog, Inc. Softcard is a trademark of Microsoft Consumer Products.

This document is copyrighted and all rights are reserved. This document may not, in whole or part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior consent, in writing, from Nestar Systems, Incorporated.

Copyright 1981, 1982, 1983 by Nestar Systems, Inc.  
2585 E. Bayshore, Palo Alto, California, 94303  
(415) 493-2223 Telex 171420 Nestar PLA

Publication number SC40-0501-0  
All rights reserved.

## How to Use This Guide

This guide provides a description of those PLAN 4000 system functions that are specific to the Apple // Pascal environment.

When familiarizing yourself with the system, read first the PLAN 4000 General Information Manual.

For information pertaining to your network's file servers, which allow you to share hard disk memory with other network users, read the File Server User's Manual. A bibliography of PLAN 4000 system manuals and guides appears at the end of this guide.

The material in this document applies to Version 2.0 of the File Server software.

We welcome criticisms and suggestions. Forms for reporting program errors and documentation errors or inadequacies are provided at the back of this manual.

CONTENTS

1. List of Figures ..... 1

2. List of Tables ..... 2

3. List of Abbreviations ..... 3

0.	Vital Statistics of Various States
1.	.....
2.	The New England States
3.	.....
4.	.....

5.	.....
6.	.....
7.	.....
8.	.....
9.	.....

10.	.....
11.	.....
12.	.....
13.	.....

# CONTENTS

## Contents

Disclaimer .....	ii
How to Use This Guide .....	iii
Contents .....	v
Chapter 1 Starting Up	
1.0 Introduction.....	1-1
1.1 Starting Up: Booting Virtual Volumes..	1-2
Chapter 2 Using the Network	
2.0 Virtual Diskettes of Variable Size...	2-1
2.1 Drives .....	2-3
2.2 The NET Program: Entering File Server Commands from the Keyboard .....	2-5
2.3 Initializing a Virtual Diskette in Pascal .....	2-12
2.4 Multiple File Servers .....	2-13
Chapter 3 Programming	
3.1 Using File Server Commands Within a Pascal Program .....	3-1
3.2 A Sample Program .....	3-7
3.3 Reading the Station Number from a Program .....	3-8
3.4 The Nestar CVAL Convention.....	3-9
Chapter 4 Additional Utilities	
4.0 Introduction .....	4-1
4.1 SETPROFILE.....	4-2
4.2 NESTAR.START Command Program.....	4-4
4.3 TREEWALK Utility Program.....	4-10

**CONTENTS**

**4.4 ZAPPASCAL Utility Program..... 4-13**

**Appendices**

**A Error and Information Messages..... A-1**

**B The Boot Program..... B-1**

**B.1 Description..... B-1**

**B.2 Operation of Boot..... B-2**

**C Quick Reference List: File Server  
Commands..... C-1**

**Index**

**Bibliography**

**Reader Comment Form**

## Chapter 1

## Starting Up

1.0 Introduction

This guide assumes that you are familiar with the Apple // microcomputer and with Apple Pascal.

The PLAN 4000 system supports Apple // user stations running with Apple Pascal Version 1.1 (based on U.C.S.D. Version II.1). In order to use Pascal on a given user station, the microcomputer must have a minimum 64K of RAM.

Three 80-column boards are supported: ALS Smarterm, M&R Sup'rterm, and Videx Videoterm. Their use is optional but recommended. A local 5.25" disk drive is optional and can be used to read and write data in standard Apple // Pascal format.

The Pascal Operating System runs on the network just as it does on a standalone microcomputer. However, there are extensions, which will be covered in this guide:

Virtual diskettes of variable size (Chapter 2)

Virtual and real disk drives (Chapter 2)

Issuing file server commands with the NET program (Chapter 2)

Issuing file server commands within a user's program. (Chapter 3)



1.1 Starting Up: Booting Virtual Volumes

At power on, your station will display the Nestar logo and prompt  
VOLUME TO BOOT?

As described in the File Server User's Manual, the network allows you to share hard disk space with other users. Hard disk space is divided into "virtual volumes", which are files formatted to resemble real diskettes.

Instead of inserting a real diskette at startup, you type in the "pathname" (the hard disk file name) of a virtual volume that is formatted to look like a Pascal diskette. The Pascal operating system will be loaded into your machine, just as happens when you boot a real diskette from a local minidisk drive.

Remember that a virtual volume has two names: its pathname (hard disk file name) and its Pascal volume name, which is used just as a real diskette's Pascal name is used.

Just as the network uses virtual volumes, it uses virtual drives. These are discussed in the File Server User's Manual and in Chapter 2 of this guide. To use a virtual volume, you must "mount" it on (that is, insert it into) a virtual drive.

Volumes are created, mounted, and otherwise handled by executing a program called NET and issuing commands from it. Just as the Pascal Filer program manipulates files within a diskette, real or virtual, the NET program manipulates virtual volumes and the drives they are mounted on. In this respect, the NET program is a Filer for the network's shared hard disks.

In response to the "VOLUME TO BOOT?" prompt, enter the pathname of a virtual Pascal volume and press Return. The Network Boot program (Appendix B) will recognize that a Pascal volume is to be booted, and will mount the volume for you on drive 4. You can't mount it yourself using the NET program, because the NET program isn't available until the boot process is complete.

Most networks use a shared library volume, which contains programs and data useful to all network users (such as NET). If such a volume exists, it will be mounted automatically at startup for READ/ONLY use on drive 5. The actual name of the shared library volume is specified in the Profile volume corresponding to the address of your workstation. Your boot volume is mounted for READ/WRITE UPDATE use. Usage, access, and protection are discussed in Chapter 3 of the File Server User's Manual.

The network SETPROFILE utility program allows you to set defaults and autoboot your volume, if desired. To use SETPROFILE, see Chapter 4.

After booting your volume, you can use both real and virtual diskettes. See Section 2.1 for a discussion of real and virtual drives.

NET WORK

8  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100

## Chapter 2

## Using the Network

2.0 Virtual Diskettes of Variable Size

The NET program is used to issue Filer-like commands to create and manipulate files (called "virtual volumes") on the network hard disks.

NET's CREATE command, for example, is used to create virtual volumes. CREATE is discussed in Chapter 5 of the File Server User's Manual. To use CREATE, X(ecute #5:NET or LIB:NET from the Pascal Command line, where #5: or LIB: is the shared network library volume. The NET program is described below in Section 2.2.

Once a volume has been created, it will look like a real blank diskette to your operating system, formatted for Pascal. Like a real blank diskette, it will have to be "Zeroed" (initialized). To Zero a newly created volume, mount it for READ/WRITE usage, using the NET program MOUNT command, quit NET, and go to the Filer. Press Z for Zero, answer "N" to the question about duplicate directories, enter the size of the volume, and name it. Remember that a volume has two names: its network pathname (or hard disk file name), and its Pascal volume name. One name is for use with the NET program; the other, for use with the Pascal Editor, Filer, etc.

Size is an optional parameter when CREATEing a Pascal virtual volume. Virtual volumes do not have to be the same size as a 5.25" or 8" diskette, for instance.

## NETWORK

If no size is specified, each Pascal virtual diskette is the size of a real minidisk, i.e., 280 blocks (each 512 bytes long). Otherwise, size can be specified with the CREATE command using the parameter:

[,SIZE=b]

where b is the number of 512 byte blocks. The maximum size is limited by the size of the disk unit on which the virtual volume will be stored and by limitations in the Pascal operating system. The number of blocks for a Pascal virtual diskette must be:

>0                    (minimum)  
<=32767              (max for system hard disk)

NOTE: The system also allows size to be specified in units of:

C            (characters--1 byte each)  
S            (sectors--256 bytes each)  
B            (blocks---512 bytes each)  
blank        (blocks---512 bytes each)  
K            (kilos---1024 bytes each)

by appending the appropriate character to the end of the size specification (e.g., SIZE=280B). Numbers may be given in \$hex or decimal.

## 2.1 Drives

Each user station operating under Pascal has 10 minidisk drives (volume numbers 3-12) that can be "virtual". Drives 9-12 can also be real. Real drives are local, private minidisks attached directly to the user station; virtual drives are volumes that reside on a system hard disk connected to a file server station. In addition, drive 6 can be set real if a locally connected printer is to be used.

NOTE: Only drives 9, 10, 11, and 12 can be real for use with minidisk drives and drive 6 can be real for use with a local printer. Drives 7 and 8 are used for the Pascal devices REMIN: and REMOUT:, respectively.

The BOOT program, part of the PLAN 4000 software, sets the drive status of each drive when a Pascal virtual diskette is booted at an Apple // user station. Whether a drive is set by BOOT to be real or virtual depends upon the presence or absence of network interface cards or standard Pascal Disk II controllers in certain slots.

The slot containing the network interface card for the network you are booting from is called the "boot slot". When you boot, your Pascal drives 4 & 5 will always be set virtual. Drive 4 will hold your boot volume and drive 5 will hold the shared library volume for your network.

If you are connected to real minidisk drives by controller cards in other slots, some drives will be set for use with them when you boot. If no such cards are present, all drives will be available for use with virtual volumes on the network you boot from (although drives 3 and 6-8 must be set virtual using the NET program or

## NETWORK

STARTCMD, as described below).

At power on, the drive status for each drive is as follows:

<u>Drive</u>	<u>Drive Status</u>
3	Real. May be set virtual by using NESTAR.START program (Chapter 4) or NET program (Chapter 2).
4 & 5	Always virtual to booted slot (typically 6 but can be 1-7).
6 - 8	Real. May be set virtual by user using NESTAR.START program or NET program.
9 & 10	Real if slot 4 contains a standard microcomputer disk controller.  Otherwise virtual to booted slot.
11 & 12	Real if slot 5 contains standard Microcomputer disk controller.  Otherwise virtual to booted slot.

The convention for setting the initial drive status (as described above) will usually provide acceptable defaults for a wide variety of user station configurations. In the normal case, where the network card is in slot 6 and slots 4 and 5 are empty, the drives 4, 5, 9, 10, 11 and 12 are all virtual to slot 6. The NET program (described in the following section) can be used at any time to display and/or change the the current drive status values.

BOOT boots from the highest numbered slot containing a network card (normally 6) and looks for second and third network cards in slots 4 and 5.

If the network is temporarily not running, entering ctrl-shift-N will permit local processing to continue at the user station. However, if the user wishes to reboot, it is necessary to press RESET.

## 2.2 The NET Program: Entering File Server Commands From the Keyboard

The NET program is an interactive Pascal program supplied by Nestar, which allows a user to enter file server commands at the keyboard. NET is generally made available as a CODE file in the default library volume that is automatically mounted on drive 5 by BOOT. If NET is not found there, the system manager will know where it is installed.

Network commands are discussed in the File Server User's Manual.

The NET program sends each command, as entered, to the file server (except for the local commands described below). The file server executes the command. If there is an error, a numeric error code and an error message will be displayed. A complete list of error messages is found in Appendix A.

NET then redisplay on the user station screen its prompt (:) for another file server command to be entered. The cycle repeats until "Q" <return> or <esc><return> is entered; then control returns to the Pascal command level.

NET obeys the Nestar CVAL convention, which allows NET to be called and executed (chained to) from another Pascal program, with control returned subsequently to that program. The Nestar CVAL convention is discussed in Chapter



## NETWORK

3.

The following printout shows a sample run or session. User responses are shown in lower case to distinguish them from information displayed by Pascal and NET. (Actually, commands can be entered in either upper or lower case, and output from Pascal on a microcomputer with upper/lower case capability is usually upper and lower case.)

```
COMMAND: E(DIT,R(UN,F(ILE,C(OMP,L(INK,X(ECUTE,..
```

```
x
```

```
EXECUTE WHAT FILE? lib:net
```

```
ENTER NETWORK COMMAND
```

```
: create lunch,t=p
```

```
109,ROOT DIR NOT SPECIFIED; NO DEFAULT
```

```
: set dir /main/pascal
```

```
[No message in response to a command indicates  
that the command was successfully executed.]
```

```
: show dir
```

```
/MAIN/PASCAL
```

```
: create lunch,t=p
```

```
: q (return)
```

```
COMMAND: E(DIT,R(UN,F(ILE,C(OMP,L(INK,X(ECUTE,..
```

In this example, a user executes the NET program, which allows him or her to send commands to the file server. The user wishes to create a virtual volume for use on the network. This volume, a portion of one of the network's shared hard disks, will be of type "P", formatted to resemble a real Apple // Pascal

diskette (and since no size is given, by default the same size as a real diskette).

The user does not enter the volume's full pathname, perhaps assuming that a part of the pathname will be added by a default, set at startup by a "SET DIR" command.

No such default "partial" pathname has been set, however, and an error message is returned. The user then sets a default partial pathname, /MAIN/PASCAL, for future convenience, and asks that it be displayed ("SHOW DIR"). The user then creates the desired volume and quits the NET program. NET commands are discussed in detail in the File Server User's Manual.

In addition to the commands that it sends to the file server, NET responds to a number of local commands that it executes itself. These commands, discussed in the following sections, are concerned with displaying and setting the drive status (real or virtual) and network interface card slot number associated with each Pascal drive.

### 2.2.1 SHOW DRIVES [NOCHECK]

This command displays the status of all the Pascal units. The information displayed consists of the following:

- the unit number.
- a "#" character if the unit is a disk. (same convention as Pascal filer's "V" command.)
- volume name.
- number of blocks on device (disks only).
- REAL or VIRTUAL indication. If VIRTUAL then the "virtual channel" (i.e., the slot of the network card and station number of the file

## NETWORK

server) is displayed.

The SHOW DRIVES display has the following format:

DRV	T	NAME	BLKS	STATUS
1		CONSOLE:		REAL
2		SYSTEM:		REAL
3	#	-----		VIRT:
4	#	ABC1:	800	VIRT: SLOT=6,STN=\$FE
5	#	LIB:	1800	VIRT: SLOT=6,STN=\$FE
6	#	-----		VIRT: SLOT=6,STN=\$FE
7	#	-----		VIRT: SLOT=6,STN=\$FE
8	#	-----		VIRT: SLOT=6,STN=\$FE
9	#	-----		VIRT: SLOT=6,STN=\$FE
10	#	-----		VIRT: SLOT=6,STN=\$FE
11	#	-----		VIRT: SLOT=6,STN=\$FE
12	#	ABC2:	800	VIRT: SLOT=6,STN=\$FE

In this example, the user has a virtual volume with the Pascal volume name ABC1 mounted on drive 4 and a virtual volume with the Pascal volume name ABC2 mounted on drive 12. Drives 3 through 12 have been set to virtual. The file server station number is \$FE. The shared library virtual volume is mounted on drive 5.

Normally SHOW DRIVES will read in all the volume names to make sure the information is current. However, the NOCHECK option can be used to suppress this check.

### 2.2.2 SHOW FS

This command displays the default "virtual channel" (consisting of a network card slot number and file server station number) to which file server commands are currently being sent.

2.2.3 SET DRIVE[S] n[,n,...] REAL|VIRTUAL  
[SLOT n] [STN \$nn]

This command sets the drive number (or numbers) given to the status specified. If REAL is specified, then the real microcomputer disks are used (if present). If VIRTUAL is specified then the drive is a virtual drive connected to a file server. The SLOT and STATION number of the file server can be specified, if the drive is to be set virtual to a file server other than the current default (seen by issuing SHOW FS).

Drive numbers in the range 3 to 12 are available. For example:

```
:set drives 9,12 real
:show drives
```

DRV	T	NAME	BLKS	STATUS
1		CONSOLE:		REAL
2		SYSTEM:		REAL
3	#	-----		VIRT:
4	#	ABC1:	800	VIRT: SLOT=6,STN=\$FE
5	#	LIB:	1800	VIRT: SLOT=6,STN=\$FE
6	#	-----		VIRT: SLOT=6,STN=\$FE
7	#	-----		VIRT: SLOT=6,STN=\$FE
8	#	-----		VIRT: SLOT=6,STN=\$FE
9	#	-----		REAL
10	#	-----		VIRT: SLOT=6,STN=\$FE
11	#	-----		VIRT: SLOT=6,STN=\$FE
12	#	-----		REAL

Here, a user station has used SET DRIVES to change drives 9 and 12 at his station to real. These drives are now available for use with real floppy disk drives. (When set real drive 3 is unusable, and drives 6-8 are used by the Pascal devices PRINTER:, REMIN:, REMOUT:, respectively.)

## NETWORK

### 2.2.4 SET FS [SLOT n] [STN \$nn]

Sets the default "virtual channel" for file server commands issued by NET or by subsequently run programs to the values specified. The initial default file server channel is the slot booted from (usually slot 6) and station \$FE.

### 2.2.5 NOECHO

Commands from a file (initiated by %filename) are usually displayed when they are executed; NOECHO suppresses this.

### 2.2.6 OFF

OFF initializes your user station: unmounts all MOUNTED files, unlocks all locks, resets any default directory and private or group passwords. OFF BRUNS the BOOT program.

In order to increase overall system efficiency, the OFF command should be given whenever a user is through using his or her Apple. This frees the memory and other resources assigned to the user station.

Giving this file server command is equivalent to turning power off and on at the workstation. Note that when the power is turned off at a user station, all file server virtual volumes that are mounted remain mounted. Locks remain held and default directory, if any, remains set. If files are mounted EXC RW, for example, they will not be available to other users. It is the OFF command (which is automatically issued by the Network BOOT program) that unmounts all volumes. Thus, in order to free memory space and leave volumes available to other stations, it is

necessary to use the OFF command, or to turn the machine off and on again to unmount volumes.

### 2.2.7 SPOOL TO/SPOOL OFF

SPOOL TO specifies the device or file to which all file server responses will be routed. The redirection will continue until SPOOL OFF is issued. If the specified file already exists or is the same as the current spool file, the user will be prompted to overwrite it. To determine the current spool file, simply enter SPOOL TO without a destination. To specify device and file:

SPOOL TO device name|file name

where device name can be, for example, #6: or PRINTER: and file name can be [#n:]filename. (where n is a unit number). The file name will be appended with the suffix ".text" if it is not already specified and if the last character is not a period.

When spooling to a file, the file will be properly closed after any of the following:

1. SPOOL OFF
2. SPOOL TO some other file or device
3. QUIT from NET.
4. Write error.

### 2.2.8 Other NET Inputs

The NET program also accepts the following inputs:

; <anystring> Ignored as a comment.

## NETWORK

PRINT <anystring>      Displays the text on the console.

command; command;...    Multiple NET commands are allowed on the same input line if they are separated by a semi-colon.

%filename                Causes commands to be read in from the Pascal text filename specified.

### 2.3 Initializing a Virtual Diskette in Pascal

Suppose that, as a new user, you want to write new Pascal programs or run existing ones on the network. What must you do to get started? In general, you (or the system manager) must:

Boot an existing Pascal virtual disk.

CREATE a Pascal virtual disk using NET.

MOUNT it on drive 3, or 6-12 for RW access. (The drive must be virtual, not real. Remember that the default for drives 3, 6, 7, and 8 is real.)

Use the Filer to ZERO its directory (see the Apple // Pascal Operating System Reference Manual).

Boot the new virtual disk.

If you want to be able to boot the volume, use the Editor to create a STARTCMD.TEXT file of startup commands on the virtual disk. The commands will be executed automatically whenever the disk is booted. Startup commands are discussed in Chapter 4.

Use the Filer to transfer files from a real disk inserted in a minidisk attached to the user station, if desired. Use the Filer to transfer files from another currently mounted virtual volume.

Use the E(ditor, C(ompiler, and L(inker to create Pascal programs, just as on a standalone microcomputer.

For example, suppose that an educator has a set of arithmetic courses which he previously ran on a standalone Apple //. Now he wants to present these same courses on the network.

Let's assume that there is a system disk called COURSES and that he wants to CREATE a virtual disk called ARITHMETIC on it. He would give the CREATE command:

```
CREATE /COURSES/ARITHMETIC,T=P,D9,RW
```

(Assume that no protection is desired and that the size of a real diskette is satisfactory. CREATE is used here to mount as well as create the virtual disk.)

Then using the Pascal Filer, he would ZERO the directory on the virtual disk, and transfer files from the real disk to his virtual one.

## 2.4 Multiple File Servers

PLAN 4000 systems may have more than one file server on the same network. In addition, user stations may be attached to more than one network.

A user station identifies file servers by the number of the slot containing the network interface card for the network that the file



## NETWORK

server is attached to, and by the station number for the file server on that network.

User stations have two kinds of communications with file servers: users may issue commands to the file server, and may issue I/O requests for virtual drives. A user station may be configured so that it has virtual disks mounted on a number of different file servers at the same time. The Pascal system can then be used to transparently access any combination of these virtual disks, just as if they were real disks, regardless of what actual file server station they are stored on.

The commands SET FS and SET DRIVES allow the user to configure his or her Pascal 'drives' to correspond to virtual disks on various file servers. These commands can set drives 3-12 to be real or virtual, to define which file server the drives are connected to, and to set a default "virtual channel" that will be used when file server commands are issued.

There are many interesting and complex system configurations that can be constructed using multiple file servers, but they are all built upon the same basic primitive operations. As an example, suppose that it is necessary to manually transfer a file within a Pascal volume from one file server to another on the same network.

Assume that, in addition to the primary file server on station \$FE, the system manager has installed another file server on station \$FD. The following procedure will transfer a file called "XREF.TEXT" from Pascal volume "/MAIN/USERS/LJS/1" on the primary file server (\$FE) to the Pascal volume "/MAIN/ARCHIVE/PVOLS" on the second file server (\$FD):

1. BOOT a Pascal volume in the usual way. You will be booting from your "default" file server, which has address \$FE.
2. From the command level of the Pascal system, X(ecute the NET program. Issue the following commands:

```
SET DRIVE 11 VIRTUAL
(set D11 virtual to $FE)
```

```
MOUNT /MAIN/USERS/LJS/1,D11
(mount the source volume)
```

```
SET FS STATION $FD
(change default file server)
```

```
MOUNT /MAIN/ARCHIVE/PVOLS,D12,UPD
(Mount the destination volume on file server
$FD)
```

```
SET DRIVE 12 VIRTUAL
(Set D12 virtual to the default file server,
which is now station $FD)
```

```
SET FS STATION $FE
(Restore the default file server for future
commands)
```

```
QUIT
(Exit from NET)
```

You can use the "SHOW DRIVES" and "SHOW FS" commands to see the status of drives and the file server command channel.

3. Use "F" to go to the Pascal filer, and T(ransfer from "#11:XREF.TEXT" to "#12:\$". The file will be copied from the \$FE file server to the \$FD file server.

## NETWORK

This example is "manual" in the sense that the commands are typed interactively. The same operations can be performed from within a program by using the CMDUNIT unit provided with the system to issue the commands, and then using Pascal I/O of any kind to transfer the data. Other interesting additions, such as using symbolic or indirect names instead of the file server station numbers, can be added at the same time.

## Chapter 3

## Programming

3.1 Using File Server Commands Within a Pascal Program

NESTAR provides Pascal "units" that allow the Pascal user to write programs that issue commands to the file server. The unit library distributed is called

## NETWORK.LIBRARY

and it contains the following units:

CMDUNIT - a Pascal regular unit that allows programs to issue file server or local commands (i.e., any command accepted by the NET utility program). In fact, this is the unit that the NET utility program uses. This unit requires the unit NETUNIT, also contained in NETWORK.LIBRARY.

NETUNIT - a Pascal regular unit that allows programs to issue commands to the file server (local commands are NOT accepted here) and optionally gets responses back for processing by the program. In addition, NETUNIT contains routines that allow direct station to station communication. These routines are not described here, but a description can be found in the related Nestar Technical Note.

NETWORK.LIBRARY may be installed in SYSTEM.LIBRARY at your local installation. If not, the Pascal program must use the \$U option to specify the location of the unit library to the compiler. During linking, the same library name must also be given.

## PROGRAMMING

A program using CMDUNIT and/or NETUNIT must have the following "USES" in the source of the program:

```
USES (*$U network-library-name*) NETUNIT,CMDUNIT;
```

Note the order: NETUNIT must come before CMDUNIT for the declarations to compile without errors.

If the program uses only NETUNIT, then "USES" in the source of the program can be simplified to:

```
USES (*$U network-library-name*) NETUNIT;
```

### 3.1.1 CMDUNIT

If the user wants to execute file server or local commands (just as the NET utility program does), then the Pascal procedure CMD or CMDLOOP in CMDUNIT should be used.

The CMD procedure has the form:

```
CMD (command, options)
```

where:

command is a string variable whose value is the file server command or local command to be executed.

options is an integer variable which that be set to:

0 - don't echo command if error.

1 - do echo command if error.

Examples:

```
CMD(`show mounts`,0);
CMD(`show drives`,0);
CMD(`mount /MAIN/A,D11; MOUNT /MAIN/B,d12`,1);
```

For programs in which the user enters file server or local commands from the keyboard until the user types QUIT or <escape>, the CMDLOOP procedure is provided. It has the form:

```
CMDLOOP (PROMPT);
```

where:

PROMPT is the string variable whose value is the prompt to be issued to the user.

Examples:

```
CMDLOOP(`: `);
CMDLOOP(`FS command or QUIT: `);
```

The following program, COMMAND, demonstrates the use of CMDLOOP to execute file server commands. This is, in fact, the complete source listing of the NET utility program:

```
PROGRAM COMMAND;

USES (*$U LIB:NETWORK.LIBRARY *)
NETUNIT,CMDUNIT;

BEGIN
  CMDLOOP(`: `)
END.
```

## PROGRAMMING

### 3.1.2 NETUNIT

To execute a file server command and get the return code back for evaluation by the program, use the NFSCMD procedure. The procedure has the form:

```
RC := NFSCMD (NIL,COMMAND,RESPONSE);
```

where:

RC is an integer representing the "return code" sent back by the file server (see Appendix A for a list of return codes and their meanings). A return code of 0 means no errors.

NIL is a "virtual channel" pointer. Specify NIL for the default (and perhaps only) file server. See the related Nestar Technical Note for a discussion of multiple file servers and virtual channels.

COMMAND is a string variable whose value is the file server command to be executed.

RESPONSE is a string "VAR" variable whose value is set to the last response line sent back by the file server. If the command has been executed successfully, then the last response is usually "OK". Otherwise, the last response contains the error message text.

All intermediate response lines are written to the screen.

NOTE: The NFSCMD procedure may wait an arbitrarily long time for the command to be executed and the response returned. This wait may be inappropriate if the program needs to check for other events frequently (such as key presses). For this reason, there is also a NFSCMDO procedure. The procedure has the form:

```
RC := NFSCMDO (NIL,COMMAND,RESPONSE);
```

which is identical to NFSCMD, except that the procedure will not retry if the network is not immediately available. The calling program should retry (and check for other events). For example:

```
repeat
    RC := NFSCMDO (NIL,COMMAND,RESPONSE);
until RC <> 98; (*repeat until not network
                error (busy)*)
```

To execute a file server command and get all the responses back for evaluation, (including those usually displayed on the screen without program intervention), use the NFSCMD1 procedure. The procedure has the form (to be typed on one line):

```
RC := NFSCMD1
(NIL,COMMAND,RESPONSE,NEWCMDFLAG,DONEFLAG);
```

where:

RC is an integer representing the "return code" sent back by the file server.

NIL is a "virtual channel" pointer. Specify NIL for the default (and perhaps only) file server. See the related Nestar



## PROGRAMMING

Technical Note for a discussion of multiple file servers and virtual channels.

**COMMAND** is a string variable whose value is the file server command to be executed when NEWCMDFLAG is set to TRUE. When NEWCMDFLAG is set to false it is ignored.

**RESPONSE** is a string "VAR" variable whose value is set to the next (or first) response line sent back by the file server.

**NEWCMDFLAG** is a boolean variable whose value is TRUE when a command is initially sent to the file server. It is FALSE on successive calls which get the next response lines.

**DONEFLAG** is a boolean "VAR" variable. Its initial value is ignored and it is set to TRUE if this is the last response line and FALSE if there are more response lines.

For example, a program fragment might contain:

```
COMMAND := `file server command`;
RC := NFSCMD1 (NIL, COMMAND, RSP, TRUE, DONE);
REPEAT
process RSP (note: RC not valid until DONE=TRUE)
  RC := NFSCMD1(NIL, COMMAND, RSP, FALSE, DONE);
UNTIL DONE;
process final RSP and RC
```

### 3.2 A Sample Program

The following program, MININET, demonstrates the use of NFSCMD to send commands to the file server with a Pascal program. The MININET program will get an input line from the user and send it to the file server for execution.

MININET differs from NET in the following ways:

1. Both the code number and reply returned by the file server are displayed on the console.
2. The commands referring to the user's own drives (SHOW DRIVES, SHOW SLOT, SET Sn, and SET Dd, drive status) are not available in MININET. They are executed by the NET program, using CMDUNIT, not by the file server.

```

PROGRAM MININET;

USES (*$U LIB:NETWORK.LIBRARY *) NETUNIT;

VAR
  RC: INTEGER;
  CMD,REPLY: STRING;

BEGIN
  WRITELN ('Mini Command Program');
  REPEAT
    WRITE ('FS COMMAND? '); READLN (CMD);
    IF LENGTH (CMD) <> 0 THEN BEGIN
      RC := NFSCMD (NIL,CMD,REPLY);
      (* Send command to default FS *)
      WRITELN (RC,' ',REPLY)
    END;
  (* All done if nothing typed *)
  WRITELN ('Ciao.')
  UNTIL LENGTH (CMD) = 0;
END.

```

## PROGRAMMING

A sample execution of the MININET program :

```
Mini Command Program
NFS COMMAND? show date
DATE IS 03/10/81
FS COMMAND? foo
1,ILLEGAL COMMAND
FS COMMAND?
Ciao.
```

### 3.3 Reading the Station Number From a Program

It may be useful to read the station number (1 to 255) from within a program. To obtain the station number, issue to the file server the command "SHOW STATION" and read the corresponding response, parsing it to obtain the station number.

Using Pascal, follow the examples given in the previous sections, using one of the procedures in the NETUNIT unit to issue the command "SHOW STATION". The response string will have the form

```
STATION IS $nn
```

By parsing this string, you can obtain the station number (in hexadecimal form).

The station number can be obtained in decimal form from any Pascal program that uses the unit NETUNIT. When such a program is executed, part of the initialization process for the NETUNIT unit is the loading of the station address of each network interface card into an array called MYSTN (MYSTN: ARRAY [0..7] OF INTEGER). Thus, if the NIC in a user's station is in slot 6, the address of the NIC can be found in MYSTN [6].

A value of 0 in the MYSTN array for a particular slot indicates that the slot does not contain a network interface card.

### 3.4 The Nestar CVAL Convention

Pascal Release 1.1 allows one program to chain to another, and provides for a global string called CVAL to be passed as a parameter.

See Pages 2 and 3 of the "Addendum to the Apple Pascal Language Reference Manual" (Apple Part #031-0101-00) for a discussion of chaining programs in the Apple Pascal environment. To use these procedures in the FS environment with the Nestar shared library, the line

```
USES CHAINSTUFF;
```

that immediately follows the Program heading must be replaced by

```
USES (*$U LIB:NESTAR.LIB*) CHAINSTUFF;
```

The Nestar definition of the CVAL variable allows it to be used both to specify a series of program to be executed, and to supply parameters to those programs. Programs which obey this convention can thus be called subroutines or be linked in a chain of consecutive program executions.

The general format of the CVAL string as a program begins execution is:

```
parm//program2, parm2//program3 ...
```

where "parm" is a parameter that is being passed to the current program. When it terminates,

## PROGRAMMING

program1 is the next program to be executed, and it is passed the string starting with parm2.

For example, if the CVAL string has the following value when program ABC is executed

```
verbose//DEF,nosend
```

then program ABC will be executed with "verbose" as a parameter, after which program DEF will be executed with "nosend" as a parameter.

Complex CHAIN sequences may be implemented by programs which add to as well as remove names from the CVAL string. Two procedures that manipulate the CVAL string in accordance with this convention are shown here:

GETPARAM (parm-string);

This procedure sets "parm-string" to the parm provided to this program. In addition, if a CHAIN program name was given then a SETCHAIN is done (for this reason, GETPARAM should be called even if the invoking program doesn't expect/accept a parm). This procedure should only be called once per program since multiple calls will unstack multiple parm/program list entries.

SETPARM (parm-string, program-name-string);

This procedure sets the parm for the next program which will be executed. In addition, the program name of the program to be executed after the next program has completed can be specified (e.g, the program name could be the original program name to cause a "return" to the original program when the next program has completed). SETPARAM always returns and multiple SETPARAM calls are legal (to set up a list of programs to execute).

```
PROCEDURE GETPARG (VAR PARG:STRING):
```

```
VAR
```

```
  P: INTEGER;
  CVAL,PARG:STRING;
```

```
BEGIN
```

```
  GETCVAL (CVAL);
```

```
  P := POS ('//',CVAL); (*find the end of our parm*)
```

```
  IF P = 0 THEN BEGIN
```

```
    PARG := CVAL;
```

```
    CVAL := ''
```

```
  END
```

```
  ELSE BEGIN
```

```
    PARG := COPY (CVAL,1,P-1);
```

```
    DELETE (CVAL,1,P+1);
```

```
    P := POS (';',',',CVAL);
```

```
    IF P = 0 THEN BEGIN
```

```
      PARG := CVAL;
```

```
      CVAL := ''
```

```
    END
```

```
    ELSE BEGIN
```

```
      PARG := COPY (CVAL,1,P-1);
```

```
      DELETE (CVAL,1,P)
```

```
    END;
```

```
    IF PARG <> '' THEN SETCHAIN (PARG);
```

```
    (* XCTL pgm name *)
```

```
  END;
```

```
  SETCVAL (CVAL)
```

```
END;
```

PROGRAMMING

PROCEDURE SETPARM ( PARM,PGM:STRING);

VAR

    CVAL: STRING;

BEGIN

    GETCVAL (CVAL);

    SETCVAL (CONCAT ( PARM,'//',PGM,' ',CVAL))

END;

## Chapter 4

## Additional Utilities

4.0 Introduction

In addition to the BOOT program (Appendix B) and the NET program (Section 2.2), the PLAN 4000 system provides a number of utility programs for use at the individual network stations. Before attempting to execute them, ask your System Manager for the names of the volumes on which they reside at your installation, (usually a volume called /MAIN/LIB/APPLE2/UCSD/PASCAL, with the Pascal name LIB:).

SETPROFILE allows configuration of a profile record for a particular station address so that defaults can be preset for the BOOT process (available on /MAIN/SYSTEM/SETPROFILE).

NESTAR.START executes a file of commands on the user's boot volume at Pascal initialization time.

TREEWALK.CODE allows the Pascal user to use NET commands to manipulate entire subtrees of the shared hard disk, rather than just single volumes.

ZAPPASCAL is a program that changes SYSTEM.PASCAL so that certain operating system files can be used from the shared library volume on drive 5, rather than from the boot volume. In this way, space is freed up on the boot volume for other uses.



## SETPROFILE

### 4.1 SETPROFILE

The BOOT program supports station dependent profiles (virtual volume containing boot defaults for your station). Each station can have its own autoboot defaults, which are invoked when the station is powered on. If a profile record is not present for a particular station, a systemwide default profile is used. The system manager sets these defaults using SETPROFILE.

The profile records are file server binary (T=B) files, each 2 blocks long (size=2.B). The record /MAIN/SYSTEM/PROFILE/\$nn is used when station \$nn is powered on. If a station's profile is not present in /MAIN/SYSTEM/PROFILE, the profile for a default (/MAIN/SYSTEM/PROFILE/\$00) is used by BOOT. Therefore, /MAIN/SYSTEM/PROFILE/\$00 must be present. The default (\$00) profile distributed with the system can also be changed using SETPROFILE.

#### 4.1.1 How To Use SETPROFILE

To use SETPROFILE, boot the virtual volume /MAIN/SYSTEM/SETPROFILE. The program will be executed automatically.

SETPROFILE will display the number of the station being used and will ask for the number of the station for which a profile is to be set.

When modifying a profile, a password may be specified for the profile by entering a colon (: ) and the password after the \$nn station number.

The program then prompts for each allowed default:

## SETPROFILE

```
Boot volume: /MAIN/?
Boot automatically: NO?
Pascal access rights: RW,UPD?
DOS access rights: RW,UPD?
Library volume: /MAIN/LIB/APPLE2/UCSD/PASCAL
Station name: Virginia?
```

If no profile has already been set for the indicated station, the prompts display the defaults set in the system \$00 profile (shown in the example above), which is used if no individual station profile exists. Otherwise, the prompts list those defaults currently set in the station.

When configuring a station's profile for a Pascal boot volume, ignore the lines labeled:

```
DOS access rights
Pascal access rights
Library volume
```

Supplying a station name is optional. Such information is very useful if the station for which the profile record is being configured is a server station (i.e., file server or print server).

When the defaults have been entered, the program displays a summary and asks

```
OK to update profile (Y/N)?
```

Enter Y to accept the defaults, or N to re-enter the defaults.

When you choose to exit the SETPROFILE utility program, the network logo will appear on the screen. Power your machine off and back on to get the "Volume to Boot?" prompt.

## NESTAR.START

### 4.2 NESTAR.START Command Program

NESTAR.START is the program that automatically runs in the user station when the boot volume is mounted. NESTAR.START first checks the booted disk for a file named SYSTEM.STARTUP. If present, it is chained to. Otherwise, NESTAR.START reads a file from the booted disk called STARTCMD.TEXT and interprets each line as a command.

Note that in the PLAN 4000 system, users do not need to have SYSTEM.PASCAL or SYSTEM.APPLE on their boot volumes to run in the Pascal environment. Instead, all stations can use the Nestar provided equivalent code files on a shared library disk. See Appendix B (BOOT Program) and the ZAPPASCAL Utility Program below for more details.

The NESTAR.START program reads a file from the booted disk called STARTCMD.TEXT and interprets each line as a command. The format of each command is:

<command> ::= <verb><blanks><optional-operand>

The legal <verb> <operand> sequences are as follows:

- |                  |                                  |
|------------------|----------------------------------|
| ; <anystring>    | Ignored as a comment.            |
| PRINT<anystring> | Prints the string on the screen. |
| CLEAR            | Clears the screen.               |

NET<fileserver-command>Executes file server or local command (e.g.,MOUNT or SET DRIVE).

BOXCURSOR Turns on the M&R 80-column video Sup<sup>r</sup>R<sup>r</sup>Terminal card's boxcursor.

LOWERCASE Sets lower case mode on the M&R card.

NOECHO <command> Executes <command> but doesn't print it.

ASK <command> Asks whether to execute <command>.

ASK=Y <command> Same as ASK, except that a carriage-return (only) response means yes.

ASK=N <command> Same as ASK, except that a carriage-return (only) response means no.

ASK ... PROMPT text The PROMPT option on ASK allows you to use your own prompt text. The prompt is NOT quoted and consists of whatever follows the word "PROMPT".

NESTAR.START

CONTINUE

Asks if it is ok to continue. A reply of no means QUIT.

CONTINUE=Y

Same as CONTINUE, except that a carriage-return (only) response means yes. The prompt on screen will be

CONTINUE (Y/N)?

and any typed response will replace the default Y.

CONTINUE=N

Same as CONTINUE, except that a carriage-return (only) response means no.

SET USER id

Sets a file server lock with the id specified. The convention is that "id" are your initials and the lock name generated is USER.id. This allows a user to issue the FS command "SHOW ALL LOCKS" to determine the initials of the users currently using the network.

SHOW FORMDATE

Displays the date and time on the screen. The format is:

Day dd-Mon-Year hh:mm:ss

NEWS

Displays general system information (contents of LIB:NEWS.TEXT).

CHAIN<program-name><,parms>Passes control to the program named. Note that control is never returned to the startup program so commands following this command are not processed.

SET DATE

Sets the Pascal system date to the current date (from the file server), as if the user had used the D(ate command in the Pascal Filer.

QUIT

Exits the NESTAR.START program.

Example

```
; this is abcl:startcmd.text
PRINT
PRINT  --> THIS IS /MAIN/USERS/ABC/1  <--
PRINT
continue=y
set date
noecho boxcursor
noecho lowercase
noecho set user abc
noecho net set drive 3,6,7,8,9 virtual
net set dir /main/users, prvpw=xxx
net mount abc/temp,d9,rw,shr
PRINT
ask=n news prompt display current news
```

This list of program commands begins with a comment ignored by the startup program ("this is startcmd.text"). When the volume is booted, "THIS IS /MAIN/USERS/ABC/1" is printed on the user station screen. The user is then asked whether the startup program should continue or not. Typing "n" <return> quits the program and the Pascal command line comes up. Typing <return> causes the startup to continue.

The program then sets the Pascal system date to the current file server date. The screen's boxcursor and the lowercase options are turned on if an MnR Superterm card is being used. No indication of these two actions is given, because "noecho" is specified.

A user id is then set in the system and drives 3, 6, 7, 8 and 9 are set to virtual (Chapter 2 discusses real and virtual drives in the Pascal environment). A default partial pathname is set, as well as a private password that may be required to access directories or virtual volumes

The virtual volume /MAIN/USERS/ABC/TEMP is mounted on drive 9 for shared, read/write use.

Finally, the user is prompted to display the system's news. Pressing <return>, however, is equivalent here to "no".



## TREEWALK

### 4.3 TREEWALK Utility Program

TREEWALK allows you to apply NET program network commands to more than one virtual volume and/or directory at a time.

When you specify a command and directory, TREEWALK will apply that command to the directory, to all volumes and directories listed in that directory, and to all volumes and subdirectories beneath that directory.

To use TREEWALK, X(ecute from the Pascal command line LIB:TREEWALK.

#### Command Syntax:

command [pathname] [options]

Legal commands are:

LIST	PROTECT
DELETE	Q(uit)
FS	RENAME
HELP	SHOW PROTECTION
?	

TREEWALK commands, for the most part, are extensions of file server commands simply applied to the entire subtree of the specified path, instead of the single path. Type an <esc> to abort a TREEWALK command.

For a more detailed description of a command, type: "HELP command".

## LIST [pathname]

This command gives a nested listing of the specified pathname. The pathname is a legal file server pathname.

## SHOW PROTECTION [pathname]

Issues a FS SHOW PROTECTION command for each directory or file found by traversing the specified path. If you only wish to traverse a specific subtree, answer "Y" to the "Show [path name of subtree]" and answer "N" to the subtrees you don't wish to see.

## PROTECT [pathname] [,protection list]

Protects each file found by traversing the specified path, with the protection list. If you wish to traverse a specific subtree answer "Y" to the "Protect [path name of subtree]" prompt and answer "N" to the subtrees you don't wish to see. You may optionally be prompted before protecting each directory or file by responding "Y" to the "Prompt for each directory or file" prompt.

## DELETE [pathname]

This command issues a FS DELETE command for each directory or file found by traversing the specified path. If you wish to traverse a specified subtree, respond "Y" to the prompt "Delete [subtree]?", or "N" if you don't wish to traverse that subtree. You may optionally be prompted before deleting each directory or file by responding "Y" to the "Prompt for each directory or file" prompt.

## TREEWALK

RENAME [pathname]

This command issues a FS RENAME command for each directory or file found by traversing the specified path. If you wish to traverse a specified subtree, respond "Y" to the prompt "Rename [subtree]?", or "N" if you don't wish to traverse that subtree. If you wish to rename a directory or file, type in the new name after the prompt:

-->RENAME SUBTREE,

If you don't want to rename that path then just enter a <cr>.

FS

This command allows you to enter file server commands from TREEWALK.

Entering

FS command

causes one file server command to be issued.

Entering

FS <return>

causes the NET prompt to be displayed. File server commands can be entered one at a time until "Q" <return> is entered.

#### 4.4 ZAPPASCAL Utility Program

ZAPPASCAL is an online system utility program that runs on a network user station. It can be used by the system manager to modify the SYSTEM.PASCAL on any virtual volume so that certain system files which normally must be on the booted volume can come from a shared disk on #5: instead. Any user station which boots using the modified SYSTEM.PASCAL will see these changes.

In particular, the following three files can be affected:

<u>Original name</u>	<u>New name</u>	<u>Function</u>
*SYSTEM.STARTUP	#5:NESTAR.START	Program executed when booting
*SYSTEM.LIBRARY	#5:NESTAR.LIB	Library used for intrinsic units
*SYSTEM.MISCINFO	#5:NESTAR.MISC	Configuration information

Remember, if there is no SYSTEM.PASCAL on the booted disk, the BOOT program will then use the SYSTEM.PASCAL on the library volume. The recommended configuration, then, is for the shared version of SYSTEM.PASCAL which is on #5: to be modified with ZAPPASCAL to use shared versions of the three files, and for individual user's boot volumes not to contain SYSTEM.PASCAL.

NOTE: As shipped, the SYSTEM.PASCAL on the Apple // Pascal shared library volume

ZAPPASCAL

/MAIN/LIB/APPLE2/UCSD/PASCAL

has been "zapped" so that it looks for #5:NESTAR.START rather than \*SYSTEM.STARTUP, #5:NESTAR.LIB rather than \*SYSTEM.LIBRARY and #5:NESTAR.MISC.

If you prefer to use a copy of SYSTEM.PASCAL that has only been partially zapped, so that SYSTEM.PASCAL references the new file name rather than the original in one but not all of the three instances cited above, you should read the following instructions.

The changes must be made to an unmodified SYSTEM.PASCAL as received from Apple. If any local changes are to be made, such as the use of BINDER to change to GOTOXY module, those changes must be made after executing ZAPPASCAL.

To run ZAPPASCAL, mount the disk that contains the SYSTEM.PASCAL with the RW option, and X(ecute ZAPPASCAL. When it asks

Name of disk?

respond with the Pascal volume name (e.g. "LIB" or "#5") of the volume which contains the SYSTEM.PASCAL. It will then ask separately for each file:

Change SYSTEM.STARTUP (Y/N)?

Change SYSTEM.LIBRARY (Y/N)?

Change SYSTEM.MISCINFO (Y/N)?

You can choose to do any subset of the modifications. If the change is successful, it will say "SUCCESSFUL". If the change had been previously made, it will say "ALREADY CHANGED".

If the proper place to change cannot be found, it will say "NOT FOUND".

These 'changes' do not change the names of the files, or the files themselves. The 'changes' are modifications to SYSTEM.PASCAL that cause it search for LIB:NESTAR.MISC instead of SYSTEM.MISCINFO. Therefore, after executing ZAPPASCAL, be sure to create the NESTAR. files which the modified SYSTEM.PASCAL will be looking for. In most environments, the NESTAR.LIB will simply be a copy of SYSTEM.LIBRARY, the NESTAR.START will be the startup program by that name distributed by Nestar, and the NESTAR.MISC will be a copy of the SYSTEM.MISCINFO used by most of the stations in the installation.

If a particular user station wants to use a different version of any of the three files, it will need to have a private copy of SYSTEM.PASCAL on its booted disk. ZAPPASCAL can be run against that private version so that any subset of three files can come from #5, and the others will come as private versions with the original names from the booted disk.

NOTE: If you wish SYSTEM.COMPILER and SYSTEM.LINKER to look for #5:NESTAR.LIB instead of SYSTEM.LIBRARY on your boot volume, use the Zappascal utility after moving SYSTEM.COMPILER and SYSTEM.LINKER to the shared library volume (//LIB/APPLE2/UCSD/PASCAL).



## Appendix A

## Error and Information Messages

When the file server receives a file server command from a user station, it executes the command, if possible, and, in any case, returns two variables:

- return code - an integer between 0 and 200
- return message, such as OK, ILLEGAL COMMAND, or TYPE=APPLE // PASCAL

Messages are of several types:

- information, such as OK, IN USE, TYPE=BINARY
- user errors such as syntax errors or attempting to access a virtual volume without the necessary access rights
- system errors such as memory full or disk full; these should be reported to the system manager.

In this list, messages are classified as:

general syntax

pertaining to a specific command

system errors

file system errors.

Some Pascal I/O error messages may be returned as a result of network operations:



## ERRORS

- 16 Write-protect error: the specified diskette is write-protected.
- 64 Device error: failed to complete a read or write correctly.

Error 16 occurs when you attempt to write into a virtual volume that has not been mounted with READ/WRITE access.

Error 64 occurs when, for example, you have created a volume and Zeroed it to a larger size than you created; that is, when the system attempted to read or write past the end of the volume. This might also occur if a block number is given that is outside the range of the file.

Error 64 may also occur if the network hard disk has been damaged.

General Syntax (0-19)

0,OK

(Not an error.) The command was syntactically correct and was executed without error.

1,ILLEGAL COMMAND

The command verb cannot be recognized, or cannot be executed from this user station operating system environment. The command verb is the first word of the command, and must be separated from the rest of the command with one or more blanks.

2,NAME PARAMETER REQUIRED

The first parameter of the command must be a non-null pathname.

3,UNRECOGNIZED PARAMETER

A keyword parameter was not recognized. Check the spelling carefully, and make sure that it is properly separated from the previous and following items with commas.

4,ILLEGAL DRIVE NUMBER

A drive number must be specified with "D" followed by an integer in the range 3 to 12. Example: "D9".

5,ILLEGAL STATION NUMBER

A station number must be specified as a two-digit hexadecimal number preceded with \$, in the range \$1 to \$FE. Example: "\$2D".

## ERRORS

### 7,PARAMETER APPEARS TWICE

A keyword parameter appears more than once in a single command. There can only be one occurrence of a keyword parameter, even if multiple occurrences have the same value.

### 8,ILLEGAL TYPE PARAMETER

The "T=" value is not one of the legal volume types. It must be one of the following:

P (Apple // Pascal)	B (binary)
D (Apple // DOS),	S (system)
C (Apple // CP/M)	Y (directory)
3 (Apple /// SOS)	T (text)
U (IBM PC p-SYSTEM PASCAL)	I (IBM PC DOS)

Example: T=P

### 9,COMMAND TOO LONG

A file server command must be no longer than 80 characters. In unusual cases where this is a problem, long commands can often be shortened by using a default directory, or by assigning access rights in a subsequent PROTECT command.

### 10,PARAMETER TOO LONG

The new name given as the second parameter of the RENAME command must be no more than 15 characters long. Remember that RENAME changes only the last name in the sequence of names which constitute a pathname. If you wish to change the name of a directory, give the pathname of the directory itself as the first parameter of RENAME.

11, ILLEGAL PROTECTION PARAMETER

The value of the PUBACC, GRPACC, or PRVACC protection item is incorrect. The value must be some combination of the letters RWECD (read, write, erase, create, and delete), or may be null. For example, "GRPACC=RW" or "PUBACC=".

12, NO STATION CMD; USE \$NN PREFIX

The "station" command has been removed. Use "\$nn <command>" where nn is the station number on whose behalf you wish to issue the command (file server console only).

13, ILLEGAL NUMBER (NEGATIVE OR TOO BIG)

The number was less than 0 or bigger than 32767. Number may be specified in decimal, or in hexadecimal preceded by \$. Examples: "42", "\$1A3".

19, HELP FILE NOT FOUND

There is no HELP file for the word used as an operand of the HELP command. Use the HELP command with any parameters to see what the possible HELP parameters are. (The system manager has the option of removing all HELP files to save space.)

Create (20-25)

20, TYPE PARAMETER REQUIRED (T=)

The type parameter was missing. Example: "T=D".

## ERRORS

### 21,SIZE OUT OF RANGE (<0B OR >32752B)

The value of the SIZE parameter was negative or greater than 32752 blocks. Each block is 512 bytes. Other units that can be specified are C (characters), S (256-byte sectors), and K (1024 bytes). If the units are omitted, B (512-byte blocks) is assumed. Examples: "SIZE=280", "SIZE=2000S".

### 23,SIZE PARAMETER REQUIRED WITH T=B, =T

The SIZE parameter must be specified to create a binary volume. The size may be an overestimate of the size of the data to be BSAVED into the volume, but it must not be less.

### 24,SIZE OUT OF RANGE (<1C OR >48K)

The limits on the size of a binary (T=B) file are between 1C (one byte) and 48K (49152 bytes) regardless of the units used in the specification.

## Delete (25-29)

### 29,CURRENTLY MOUNTED OR DEFAULT DIR

You cannot delete a volume which is currently mounted by any station, or which is the default directory (see the SET DIR command) of any station.

## Mount (30-34)

### 30,DRIVE REQUIRED

You must specify what drive the volume is to be mounted on. Example: "D4".

31,IN USE

The volume you asked to mount is in use by another station. You are therefore denied exclusive use of that volume.

32,IN EXC USE

The volume you asked to mount is in exclusive use by another station. You are therefore denied any use of the volume.

34,RW NOT ALLOWED ON DIRECTORIES

You are not allowed to mount a directory (T=Y) volume for write access.

Unmount (35-39)

35,DRIVE OR "ALL" REQUIRED

You must specify a drive number or "ALL" on the UNMOUNT command. Examples: "UNMOUNT D1", "UNMOUNT ALL".

36,NOT CURRENTLY MOUNTED

The drive number specified in the UNMOUNT command does not have any virtual volume mounted on it by the file server.

Lock (40-44)

40,ILLEGAL LOCK NAME

A lock name must be 1 to 15 characters long.

## ERRORS

### 41, IN USE

The lock name specified is in use by another station. You are therefore denied exclusive use of the lock.

### 42, IN EXC USE

The lock name specified is in exclusive use by another station. You are therefore denied any use of the lock.

### Unlock (45-49)

### 45, NOT HELD

The lock name specified is not currently held by your station.

### BSAVE, BRUN, BLOAD (50-55)

### 50, CMD NOT ALLOWED FROM THIS STATION

The BSAVE/BLOAD/BRUN commands may only be executed from an Apple // user station.

### 51, NOT BINARY

The pathname identifies a volume which is not binary (T=B). Only binary volumes can be used for BSAVE, BLOAD, or BRUN commands.

### 52, NOT PREVIOUSLY SAVED

The binary volume was created but has never had any data written into it using the BSAVE command. It can not not be used with BLOAD or BRUN until it has been BSAVED into.

## 53, FROM PARAMETER REQUIRED

For BSAVE, the FROM parameter, which specifies the memory area from which data is to be taken, must be provided. Example: "FROM=\$1000".

## 54, SIZE PARAMETER REQUIRED

For BSAVE, you must provide the SIZE parameter on the BSAVE command to indicate how much data is to be saved. This value must be less than or equal to the size of the volume specified on the original CREATE command. Example: "SIZE=256.C".

## 55, FILE WAS THERE; REP NOT SPECIFIED

Data has already been saved in the binary volume. To replace that data, you must specify "REPLACE" as a parameter in the BSAVE command.

Set/List (56-58)

## 57, FILE IS NOT A DIRECTORY

The pathname given in a SET DIR command identifies a volume which is not a directory (T=Y).

## 58, USE: SET TIME [YYMMDD][,HH[MM]]

If the date parameter was supplied, it was incomplete or contained invalid fields. If the time of day parameter was supplied, it contained invalid fields.



## ERRORS

### Show (59-69)

#### 59,TYPE=SYSTEM

(Not an error.) The volume mounted on the drive for which you did a "SHOW TYPE" is a system volume (T=S).

#### 60,SHOW OPTION NOT RECOGNIZED

The operand of a "SHOW" command was not recognized. Check the spelling and make sure there are no extraneous items in the command.

#### 61,DRIVE REQUIRED

For the "SHOW TYPE" command you must specify the drive number on which the volume is mounted. Example: "SHOW TYPE D12".

#### 62,NOT CURRENTLY MOUNTED

The drive number specified has no volume mounted on it by the file server.

#### 63,TYPE=APPLE // CP/M

(not an error.) The volume mounted on the drive for which you did a "SHOW TYPE" is an Apple // CP/M volume. (T=C).

#### 64,NO DEFAULT DIRECTORY SET

(Not necessarily an error.) This response to the "SHOW DIR" command indicates that there is no default directory established for this station by the file server.

65,TYPE=APPLE // PASCAL

(Not an error.) The volume mounted on the drive for which you did a "SHOW TYPE" is an Apple // Pascal volume (T=P).

66,TYPE=APPLE // DOS

(Not an error.) The volume mounted on the drive for which you did a "SHOW TYPE" is an Apple // DOS volume (T=D).

67,TYPE=BINARY

(Not an error.) The volume mounted on the drive for which you did a "SHOW TYPE" is a binary volume (T=B).

68,TYPE=DIRECTORY

(Not an error.) The volume mounted on the drive for which you did a "SHOW TYPE" is a directory (T=Y).

69,UNKNOWN TYPE

The volume mounted on the drive for which you did a "SHOW TYPE" is not a standard volume type.

70,yyymmddhhmmssw

(Not an error.) Returns date and time information in encoded form. For example, the FS command `TIMESTAMP` returns the current year, month, day, hour, minute, second, and day of the week in the form 70,8104301859381 (that is, 6:59 P.M., Sunday, April 30, 1981).

## ERRORS

71,--t--tt--t-t----

(Not an error.) Returns types of virtual disks currently mounted on drives 1-16 (Y,P,D,B,C,3,U,I,T). The system responds to the FS command SHOW TYPES using this format.

72,TODAY IS DD-MON-YYYY HH:MM:SS

(Not an error.) This is the response to the SHOW TIME command.

73,TYPE=TEXT

(Not an error.) The volume mounted on the drive for which you did the SHOW TYPE is a text volume (T=T).

74,TYPE=Apple /// SOS

(Not an error.) The volume mounted on the drive for which you did the SHOW TYPE is an Apple /// SOS volume (T=3).

75,TYPE=IBM PC DOS

(Not an error.) The volume mounted on the drive for which you did the SHOW TYPE is an IBM PC DOS volume (T=I).

76,TYPE=IBM PC P-SYSTEM PASCAL

(Not an error.) The volume mounted on the drive for which you did a "SHOW TYPE" is an IBM PC

p-SYSTEM Pascal volume (T=U).

Local Commands (90-92)

90,DRIVE(S) REQUIRED (LOCAL COMMAND)

A drive number is required for this command.  
 Example: "SET DRIVE 3 VIRTUAL"

91,DRIVE NUMBER CAN NOT BE USED

There are 12 drives (also called "volumes" in Apple Pascal, but only some of them can be made virtual with the SET DRIVE local command.

92,NO MORE CHANNEL SPACE (MAX=3)

In the Apple Pascal environment, a "channel" is established for each different file server you have simultaneous access to. There is space for three channels, so you may have drives virtual to no more than three file servers at the same time. There also must be a channel established for the default file server to which commands are sent. This error indicates that you have asked for more than three file server channels to be established.

System Errors (94-99)

94,NOT CONNECTED

This is a network error that indicates that the connection to another station has been terminated prematurely.

## ERRORS

### 95, UNRECOGNIZABLE NETWORK RESPONSE

The last response message from file server did not contain a numeric error code as expected. This may indicate incorrect logic in the use of NETUNIT procedures from within Pascal programs.

### 96, NO NETWORK CARD IN SLOT

The slot number specified does not have a Nestar network interface card installed, or the card is defective.

### 97, FILE SERVER MEMORY FULL

The file server has no space left for tables needed to complete your request. This does NOT refer to memory space in the user station.

### 98, NETWORK ERROR

The network routines were unable to complete the transaction. The error may be transient; see the description of NETUNIT for more information. This error does not occur from most user level programs, such as NET.

### 99, USER ABORT

The network abort key <CTRL><SHIFT>-N was pressed while a network transaction was queued up or in progress. The transaction was aborted.

File System Errors (100-140)

## 101,END OF FILE OR RECORD OUT OF RANGE

An attempt was made to read a block or sector of a volume which is outside the legal range for that volume.

## 102,FILE NOT FOUND

The volume specified by the pathname was not found, or one of the directories in the pathname was not found. Check the spelling of each filename in the pathname. If the name does not begin with a slash, check that the current default directory is the correct one.

## 103,BAD DELIMITER IN PATHNAME

A delimiter other than "/" or ":" was found in a pathname. Make sure that you have not omitted a comma separating the pathname from other operands in the command.

## 104,FILE NAME OR PASSWORD TOO BIG

A single filename (the part between slashes in a pathname), or password (the part after a colon in a pathname) is longer than the maximum of 15 characters.

## 105,NO ACCESS FOR READ TO DIRECTORY

You have been denied read access to a directory which is part of the specified pathname.

## ERRORS

### 106, NON DIRECTORY FOUND IN PATHNAME

One of the names in the pathname (other than the last name) identifies a volume which is not a directory (T=Y). Only the last thing in a pathname can be other than a directory.

### 107, END OF PATHNAME IS A DIRECTORY

The pathname specifies a directory (T=Y) in a context where a non-directory volume is required.

### 109, ROOT DIR NOT SPECIFIED NO DEFAULT

The specified pathname does not begin with a slash, and there is no default directory recorded for this station by the file server. If you wish to completely specify the pathname, begin with a slash and the name or number of the disk unit. If you wish to use the current default directory, do not begin the pathname with a slash. You may use the "SHOW DIR" command to find out the current default directory, and the "SET DIR" command to establish one.

### 110, NO ACCESS FOR READ

You have been denied read access to the volume specified by the pathname, or to a directory along the path.

111,NO ACCESS FOR WRITE

You have been denied write access to the volume specified by the pathname.

112,NO ACCESS FOR APPEND (not yet implemented)

You have been denied append access to the volume specified by the pathname.

113,NO ACCESS FOR ERASE

You have been denied erase access to the volume specified by the pathname. Erase access is necessary to delete the file.

114,NO ACCESS FOR CREATE

You have been denied access to create or rename an entry in one of the directories specified in the pathname.

115,NO ACCESS FOR DELETE

You have been denied access to delete or rename an entry in one of the directories specified in the pathname.

116,CANT DELETE NON-EMPTY DIRECTORY

The pathname specified in a DELETE command identifies a directory, and that directory is not empty, that is, it still points to other volumes. Only empty directories can be deleted by a single file server DELETE command. To delete non-empty directories (i.e., subtrees)



## ERRORS

see the description of the TREEWALK utility in Chapter 4 of this manual.

### 117,FILE SERVER MEMORY FULL

The file server has no space left for tables needed to complete your request. This does NOT refer to memory space in the user station.

### 118,DISK FULL

There is not enough contiguous space left on the disk unit to create the volume. The LIST command when used to display the root directory will give information about the space available on a disk unit. Example: "LIST /MAIN".

### 119,DIRECTORY FULL

There is not enough space left to create more entries in the directory, and the directory cannot be expanded. Directories are automatically expanded as necessary to accomodate new entries.

### 120,INTERNAL ERROR

An internal error has been detected by the file server. Additional information is written on the console of the file server. That information and the circumstances surrounding the error should be transmitted to your support organization for diagnosis. The file server should be restarted as soon as practical.

## 121, UNINITIALIZED DISK

The disk unit has not been initialized. All disks must be formatted and initialized using the file server Format Disk utility; see the File Server Installation and Operation Manual.

## 122, WRONG SOFTWARE VERSION

The disk format is not compatible with version of the file server currently running. (This error cannot occur with any file servers so far released.)

## 123, FILE ALREADY EXISTS

The file you have asked to create, or the new name used in a RENAME command, already exists in the directory.

## 124, DISK I/O ERROR, SUBCODES x,y

A hardware I/O error was detected. The details of the error are described in the subcodes; see the table of I/O errors for more information.

## 125, VERIFY FAILED (BAD MEMORY)

All disk write operations are verified by reading back the recorded data and comparing it to the data stored in memory. This error indicates that the disk data did not compare correctly. It sometimes indicates a memory error in the file server, and not a disk error.

## ERRORS

### 128,BAD UNIT NUMBER IN PATHNAME

The first item after the initial slash in a pathname is a number, but it is not in the legal range for unit numbers (1 to 4).

### 130,NEED PRIVATE PASSWORD FOR PROTECT

In order to execute the PROTECT command for any volume, the private password must be specified in the pathname or as a default private password. The password must be specified whenever the volume has a non-null private password, regardless of the access rights currently assigned to the volume.

### 131,UNIT NAME NOT FOUND

The name after the initial slash of a pathname is not the name of a disk unit currently recognized by the file server. You can also use the unit number in place of a name, if you wish.

### 133,DISK NOT READY

The disk unit specified is not ready. The one minute warmup period after initial power up may not have elapsed. It may also indicate a controller or disk drive failure.

## Appendix B

## The BOOT Program

B.1 Description

The BOOT program is a standard part of the file server, and provides a means for users to load Apple Pascal 1.1. It is automatically loaded and run when an Apple with the autostart ROM and a Nestar network interface card is powered up. If an Apple disk controller is in a higher slot than the network card, however, the Apple will boot from the local disk.

The BOOT program is a machine language program that executes in the memory area \$800 to \$3FFF. It has been provided as a BSAVED virtual Network file named /MAIN/BOOTS/APPLE2/BOOT. This section describes the action of BOOT as distributed by Nestar. You should consult your System Manager to see if any local modifications have been made to the version on your file server.

BOOT is brought into a user station's memory and executed whenever the Apple Autostart ROM (present on Apple // Plus, Apple //e, and Apple // with Language Card or Integer machines that have upgraded by installing the ROM) recognizes a power on condition. On machines which do not have the Autostart ROM, you must explicitly enter

CnØØG

(where n is the slot in which a network interface card is installed), in response to the Apple // Monitor prompt ("\*"). Entering PR#6 or 6<ctrl>P will NOT cause the BOOT program to be run. The Network interface card differs from

## BOOT

Apple Disk // controllers in this respect; an Apple disk controller will boot whenever any character is printed to the slot they are plugged into.

When a user station is powered up, BOOT will automatically unmount all disks in use by that station, free any locks held, and reset any default directory or passwords. This also happens when the OFF file server command is issued. It is strongly recommended that this be done by every user who is finished using an Apple on the system. This gives a clear indication to others that the station is not in use, and also frees up resources other stations may need.

### B.2 Operation of BOOT

BOOT will display on the Apple screen the Nestar logo, followed by:

```
PLAN 4000 BOOT X.X
STATION ADDRESS $NN
SLOT N    SERVER $TT
```

VOLUME TO BOOT: /MAIN/

where NN is your station address (in hex, from 00 to FE), and N and \$TT are the SLOT and Server numbers that communications are directed to. You should place a sticker with the station number on each user station, and you should observe that the number shown on the screen agrees. If not, please consult your System Manager.

NOTE: The boot program recognizes a large collection of Apple-compatible serial or communications cards installed in slot 3, and

directs its I/O to them automatically. Whenever I/O is being directed to the slot 3 device, the message

(I/O BEING DIRECTED TO SLOT 3 DEVICE)

is displayed on the standard Apple video output as an indication to the user.

A user may disable the automatic recognition of a slot 3 terminal-like card by pressing the space bar down between the time that the BOOT program is downloading and begins running. If this is done, the standard Apple video output and keyboard input will be used regardless of a card in slot 3.

The user may now enter the volume name of a virtual Pascal volume he wishes to load his system from. This name may be up to 40 characters long, including the prefix. A default prefix is displayed on the screen (shown above as /MAIN). The user may backspace and edit this prefix exactly as if he had typed it himself, or type <CTRL>-X to erase the whole line.

You should note that use of the file server does not in any way modify the minimum features necessary to load and execute Pascal. Any errors you would get from these standard systems will still be reflected in the system. For example, you cannot boot a Pascal system into a machine which does not have 64 RAM.

After you have entered the name of the volume you wish to boot, the BOOT program attempts to locate that volume, and issue a MOUNT command for drive 4. If successful, the type of the disk is checked. The volume is mounted read/write, update unless the individual station profile or system default profile files have been changed

## BOOT

for different access and usage (discussed below in Section B.2.3).

### B.2.1 Booting Pascal Virtual Volumes

In the case of booting a Pascal disk, a check is made to see that the 64 K is available before attempting to load Pascal. If a profile record exists for the station, the pascal library volume is located, and mounted RO and SHR on drive 5 of the users station. The volume the user requested to boot is remounted on drive 4, for RW and UPD usage or for the usage specified in the station's profile in /MAIN/SYSTEM/PROFILE.

The file SYSTEM.APPLE must be present (/MAIN/LIB/APPLE2/UCSD/PASCAL) mounted on drive 5, the shared library volume, or on the user's booted volume. If SYSTEM.APPLE is not found on either, an error message is issued.

Once SYSTEM.APPLE is loaded, the Pascal system is started. In the case of booting Apple Pascal Release 1.1, SYSTEM.PASCAL need not be present on the booted disk, and can be loaded from the library disk instead. BOOT checks the version numbers of SYSTEM.APPLE and SYSTEM.PASCAL, and will only accept Apple Pascal 1.1 systems. It also verifies that the versions of SYSTEM.APPLE and SYSTEM.PASCAL are at the same level.

In the typical Pascal system, Pascal looks for the three necessary files, SYSTEM.STARTUP, SYSTEM.LIBRARY, and SYSTEM.MISCINFO, on the user's booted volume. However, as shipped, the SYSTEM.PASCAL file has been modified to search for #5:NESTAR.START, #5:NESTAR.LIB, and #5:NESTAR.MISC rather than these SYSTEM files. These NESTAR files may reside on the drive 5

library volume, so that it is no longer necessary for all users to have the files on their boot volumes. See the description of the utility program ZAPPASCAL in Chapter 4 of this manual for more details.

Pascal will now come up and follow its normal procedures. It will recognize a serial or communications card in slot 3, and direct its subsequent input/output to that card if present.

### B.2.3 Autobooting at the Individual Station

The BOOT program can be tailored to the needs of individual stations. Using the utility SETPROFILE, the System Manager can install a profile for any station. The profile will specify which volume is to be booted for the station, whether the boot is to be automatic, the Pascal access rights, and the name of the default library volume automatically mounted on drive 5. Individual users can install station profiles themselves using SETPROFILE if they have the access rights necessary to do so.

If no profile has been set for a station, BOOT will use the defaults set by the System Manager in the system \$00 profile.

If <esc> is pressed on the Apple keyboard while the BOOT program is downloading, the station dependent profile is ignored and the default profile is used. This is useful if a station is setup to autoboot a particular volume but you wish to interrupt the autoboot process and boot a different volume.





## Appendix C

## Quick Reference List - File Server Commands

CREATE pathname,type [,size][,sectors]  
                              [,protection][,drive][,usage]

- creates and optionally mounts a new virtual disk

DELETE pathname

- deletes a virtual disk from system

HELP [command name] or ? [command name]

- displays list of commands or syntax of specified command

LIST [pathname] [,VERBOSE] [,NESTED]

- lists entries in default or specified directory

LOCK lockname [,usage]

- locks "lockname" for exclusive (EXC) or shared (SHR) use
- Default: EXC

MOUNT pathname, drive [,usage]

- establishes correspondence between a virtual disk and drive number

OFF

- initializes user station and downloads BOOT software

## COMMAND LIST

PROTECT [pathname] [protection]

- modifies passwords or access rights for a virtual disk or directory

RENAME oldpathname, newname

- renames specified FS file

SET DIR pathname

- sets default directory

SET [GRPPW=password] [PRVPW=password]

- sets default group or private password

SHOW DATE

- displays date as MM/DD/YY

SHOW INFO

- displays information about a virtual volume

SHOW DIR

- displays name of current default directory

SHOW LOCK [lockname]

- lists all stations using specified lockname

SHOW [ALL] LOCKS

- displays all locks set at own station or held in file server

SHOW [ALL] MOUNTS [pathname]

- gives information on virtual disks mounted

on the file server

SHOW PROTECTION [pathname]

- displays group, private and public access rights for virtual disk or directory addressed by pathname

SHOW STATION

- displays own station number

SHOW TYPE Dd

- displays type of virtual disk mounted on drive d

SHOW TYPES

- displays types of virtual disks mounted on drives 3-12

SHOW VOL\$

- displays names of system disk units currently operational

TIMESTAMP

- returns encoded form of the current date and time

UNLOCK lockname or UNLOCK ALL

- releases locked usage on a lockname or on all locknames currently held

UNMOUNT Dd or UNMOUNT ALL

- cancels previous mount(s)

## COMMAND LIST

### Parameters

access rights: set of letters from RWECD (read, write, erase, create, and delete)

drive: Dd, where d=number between 3 and 12

filepath: name[:password] [/filepath]

lockname: name of lock (same restrictions as name)

name: name of disk unit, directory, or virtual disk. (1-15 characters; no commas, control characters, slashes, colons, returns or unprintable characters; imbedded blanks ok; first character any ok character)

number: if decimal: between -32768 and 32767, inclusive if hex: between \$0 and \$FFFF, inclusive

password: a secret word used to gain access to protected volumes. Same restrictions as name. except control characters allowed.

pathname: /unitname/filepath or filepath (80 characters max. An initial slash indicates first field is a unit name; no initial slash indicates current default directory is to be prefixed to pathname.

protection: protect item [,protection]

protect item:

,GRPPW=password

,PRVPW=password

,PUBACC=access rights (Default: R)

,GRPACC=access rights (Default: PUBACC)

,PRVACC=access rights (Default: varies)

size: SIZE=number [.size-unit]

size-unit:

C: character----(1 byte)  
 S: sector----(256 bytes)  
 B: block----(512 bytes)  
 blank: block (512 bytes)  
 K: -----(1024 bytes)

type:

T=3 (Apple /// SOS)                   T=Y (directory)  
 T=C (Apple // CP/M)                   T=B (binary)  
 T=P (Apple // Pascal)                 T=S (system)  
 T=D (Apple // DOS)                    T=I (IBM PC DOS)  
 T=U (IBM PC p-System Pascal) T=T (text)

unitname: 1, 2, 3 or 4 [:password] or name  
 [:password]

usage: EXC(exclusive), SHR(shared), or  
 UPD(update) RO(read only) or RW(read/write)

#### Local NET Commands

SHOW DRIVES [NOCHECK]  
 SHOW FS  
 SET DRIVE[S] n[,n,...] REAL|VIRTUAL [SLOT n]  
 [STN \$nn]  
 SET FS [SLOT n] [STN \$nn]  
 SPOOL TO device name/file name  
 SPOOL OFF

#### Other NET Inputs

drive: 3-12  
 drive-status: R, REAL, or V, VIRTUAL  
 slot: default virtual slot



## A

access 1-3, 2-10, 4-3  
ASK 4-5  
automatic startup 4-2, 4-4

## B

blocks 2-7  
BOOT 1-2, 2-5, 2-10, 4-2, B-1  
boot slot 2-3  
BOXCURSOR 4-5

## C

CHAIN 4-7  
chaining 3-9  
CHAINSTUFF 3-9  
CLEAR 4-4  
CMDUNIT 3-1, 3-2  
COMMAND 3-4, 3-5  
command;command 2-12  
Command list C-1  
CONSOLE 2-8  
CONTINUE 4-6  
CREATE 2-1  
ctrl-shift-N 2-5  
CVAL 2-5, 3-9

## D



## INDEX

deleting volumes 4-10  
disclaimer ii  
disk controllers, Pascal 2-3  
DONEFLAG 3-6  
drive status 2-4, 2-8  
drives 2-3

## E

80 column cards 1-1  
error messages A-1  
<esc><return> 2-5  
extensions to operating system 1-1

## F

file server commands 2-5

## G

GETPARM 3-10

## I

initializing virtual diskettes 2-12

## K

keyboard network commands 2-5

## L

library volume 1-3, 4-14  
locks 2-10  
LOWERCASE 4-5

## M

MININET 3-7  
multiple file server commands 2-13  
multiple file servers 2-13

## N

NESTAR.START 2-4, 4-3, 4-4  
NET 1-2, 2-1, 4-5  
NET inputs 2-11  
network utilities 4-1  
NETUNIT 3-1, 3-4  
NETWORK.LIBRARY 3-1  
NEWCMDFLAG 3-6  
NEWS 4-7  
NIL 3-4, 3-5  
NOCHECK 2-8  
NOECHO 2-10, 4-5

## O

OFF 2-10  
operating system extensions 1-1

# INDEX

## P

parm 3-9  
Pascal diskette name 1-2  
Pascal operating system 1-1  
pathname 1-2  
PRINT 2-2, 4-4  
PRINT <anystring> 2-12  
PRINTER 2-9  
programming 3-1  
protection 1-3

## Q

QUIT 2-11, 4-7

## R

RC 3-4, 3-5  
reading a station address 3-8  
REAL 2-7, 2-9  
real diskettes 1-3  
real drives 2-4  
REMIN 2-9  
REMOUT 2-9  
RESPONSE 3-4, 3-5

## S

serial card  
SET DATE 4-7  
SET DRIVES 2-9, 2-14  
SET FS 2-10, 2-14  
SETPARM 3-10

SETPROFILE 1-3, 4-2  
SETUSER 4-6  
shared library volume 1-3  
SHOW DRIVES 2-8  
SHOW FORMDATE 4-6  
SHOW FS 2-8  
SHOW STATION 3-8  
size, volume 2-1, 2-2  
SLOT 2-10  
slot 2-4  
SPOOL 2-11  
STARTCMD 2-12, 4-4  
starting up 1-1, 4-2, 4-4  
STARTUP program 4-3  
STATION 3-8  
station address 3-8  
status, drive 2-4, 2-8  
STN 2-8, 2-10  
SYSTEM.LIBRARY 4-13  
SYSTEM.MISCINFO 4-13  
SYSTEM.PASCAL 4-13  
SYSTEM.STARTUP 4-13  
SYSTEM 2-8

## T

TREEWALK utility 4-10

## U

unit number 2-7  
usage 1-3, 2-10, 4-3  
USES 3-2  
using the network 2-1  
utilities, network 4-1

# INDEX

## V

video cards 1-1  
VIRTUAL 2-7, 2-9  
virtual diskettes 1-1, 1-2, 2-1, 2-12  
virtual drives 1-1, 2-3  
volume name 2-7  
volume to boot 1-2  
volumes  
    variable size 2-1  
    virtual 1-1, 1-2

## Z

ZAPPASCAL 4-13  
ZERO 2-1

## Other

%filename 2-12  
# 2-7  
\$nn 2-10, 3-8  
; <anystring> 2-11, 4-4

## BIBLIOGRAPHY

## PLAN 4000 PUBLICATIONS

General Information Manual	GA40-0100
PLAN 4000 Physical Planning and Installation Manual	GA40-0101
File Server User's Manual	SC40-0200
File Server Apple // Pascal User's Guide	SC40-0501
File Server Apple // DOS User's Guide	SC40-0502
File Server Apple // CP/M User's Guide	SC40-0503
File Server Apple /// SOS User's Guide	SC40-0504
File Server IBM PC DOS User's Guide	SC40-0505
File Server IBM PC UCSD p-System User's Guide	SC40-0506
File Server Installation and Operation Manual	SC40-0300
PLAN 4000 System Service Manual	LA40-0401
Print Server User's Manual	SC40-0201
Print Server Installation and Operation Manual	SC40-0301
Messenger User's Manual	SH40-0204
Messenger Installation and Operation Manual	SH40-0304



Reader Comment Form

This manual is one in a series that describes the use of the PLAN 4000 system.

You are encouraged to use this form to communicate to Nestar any problems or suggestions associated with the system. We would like your comments on improving the system itself, as well as on this documentation. Possible topics for comment are: clarity, accuracy, completeness, organization, coding, retrieval and legibility.

No postage stamp is necessary if mailed within the U.S.A.

Nestar installation location:

Submitter's name :

Address :

Description of problem or suggestion:

Please mail this form to :

Nestar Systems, Incorporated  
2585 E. Bayshore Road  
Palo Alto, California 94303  
Attn: PUBLICATIONS DEPT.





No Postage  
Necessary  
If Mailed  
In The  
United States

**BUSINESS REPLY CARD**

First Class

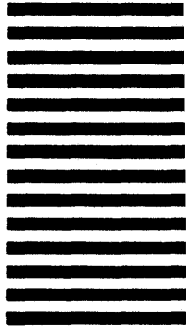
Permit No. 239

Palo Alto, CA

POSTAGE WILL BE PAID BY ADDRESSEE

**NESTAR** SYSTEMS  
INCORPORATED

2585 East Bayshore Road  
Palo Alto, California 94303



-----  
Fold here

Please do not staple

Tape

## Reader Comment Form

This manual is one in a series that describes the use of the PLAN 4000 system.

You are encouraged to use this form to communicate to Nestar any problems or suggestions associated with the system. We would like your comments on improving the system itself, as well as on this documentation. Possible topics for comment are: clarity, accuracy, completeness, organization, coding, retrieval and legibility.

No postage stamp is necessary if mailed within the U.S.A.

Nestar installation location:

Submitter's name :

Address :

Description of problem or suggestion:

Please mail this form to :

Nestar Systems, Incorporated  
2585 E. Bayshore Road  
Palo Alto, California 94303  
Attn: PUBLICATIONS DEPT.



No Postage  
Necessary  
If Mailed  
In The  
United States

**BUSINESS REPLY CARD**

First Class

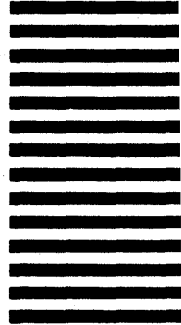
Permit No. 239

Palo Alto, CA

POSTAGE WILL BE PAID BY ADDRESSEE

**NESTAR** SYSTEMS  
INCORPORATED

2585 East Bayshore Road  
Palo Alto, California 94303



-----  
Fold here