MS-DOS FINGER Installation and User's Guide

January, 1991

This document provides installation and usage instructions for the MS-DOS FINGER Client program. FINGER is a utility which displays information about users on remote networked systems.

Revision/Update Information: This is a new manual.

Operating System and Version: MS-DOS V3.2 or later

Software Version: MS-DOS FINGER Client V1.0

Academic Computer Center
Saint Peter's College
Jersey City, New Jersey

January, 1991

Permission is granted to copy and redistribute this document provided the copies are made for no commercial advantage.

The information in this document is subject to change without notice and should not be construed as a commitment by Saint Peter's College. Saint Peter's College assumes no responsibility for any errors that may appear in this document.

DISCLAIMER: The software described in this document is provided "**as is**". No guarantee is made by the author or the author's employer as to the suitability, reliability, security, usefulness, or performance of this software.

The following are trademarks of Digital Equipment Corporation:

DEC	DECnet	PCSA
RSTS/E	VAX	VMS

The following are trademarks of Microsoft Corporation:

MS-DOS	Windows

The following is a trademark of Adobe Systems:

PostScript

The following are trademarks of US Robotics:

Courier	HST

Copyright ©1991 Terence M. Kennedy

All Rights Reserved.
Printed in U.S.A.

This document was prepared using VAX DOCUMENT, Version 1.2

Contents

PREFACE v

CHAPTER 1 PREPARING TO INSTALL MS-DOS FINGER 1–1

1.1 PREREQUISITE SOFTWARE 1–1

1.2 ACCESSING THE ONLINE RELEASE NOTES 1–1

1.3 INSTALLATION PROCEDURE REQUIREMENTS 1–1

CHAPTER 2 INSTALLING MS-DOS FINGER 2–1

2.1 DETERMINING A LOCATION FOR MS-DOS FINGER 2–1

2.2 COPYING THE DISTRIBUTION FILES 2–1

2.3 SETTING THE ENVIRONMENT VARIABLES 2–2

2.4 COMPLETING THE INSTALLATION 2–3

CHAPTER 3 USING MS-DOS FINGER 3–1

3.1 INVOKING MS-DOS FINGER 3–1

3.2 ERROR MESSAGES 3–2

3.3 REPORTING PROBLEMS AND OBTAINING UPDATES 3–3

Contents

APPENDIX A SAMPLE MS-DOS FINGER INSTALLATION A–1

APPENDIX B REBUILDING MS-DOS FINGER FROM SOURCE B–1

INDEX

Preface

This manual describes how to install and use the MS-DOS FINGER program. Rather than trying to explain what FINGER does, I'll let the person who designed the original FINGER program explain:

```
From LES@SAIL.Stanford.EDU Mon Feb 19 21:32:35 1990
Received: from cayuga.cs.rochester.edu by sol.cs.rochester.edu (4.0/q) id
 AA13568; Mon, 19 Feb 90 21:32:33 EST
Received: from Sail.Stanford.EDU by cayuga.cs.rochester.edu (5.59/q) id
 AA20402; Mon, 19 Feb 90 21:32:26 EST
Message-Id: <Kl$B9@SAIL.Stanford.EDU>
Date: 19 Feb 90  1831 PST
From: Les Earnest <LES@SAIL.Stanford.EDU>

Here is a response to your conjectures, mostly regurgitated from an
article that I posted on Human-nets in 1985.  Feel free to forward it to
alt.folklore.computers, or I can post it there if you prefer.  I haven't
been reading that newsgroup, but a quick look indicates that maybe I
should.

Finger was named for the act of pointing.  I recall that sometime after
it became popular I received a message from a system administrator who
thought that it should be renamed so that users would not have to use a
"dirty" word.  I gave his request all the consideration that it deserved.

I created Finger around 1971 to meet a local need at the Stanford
Artificial Intelligence Lab.  People generally worked long hours there,
often with unpredictable schedules.  When you wanted to meet with some
group, it was important to know who was there and when the others would
likely reappear.  It also was important to be able to locate potential
volleyball players when you wanted to play, Chinese food freaks when you
wanted to eat, and antisocial computer users when it appeared that
something strange was happening on the system.

The only tool then available for seeing who was running on our DEC-10
computer was a WHO program that showed IDs and terminal line numbers for
people who were logged in.  There was no information available on people
who were not logged in.  I frequently saw people running their fingers
down the WHO display saying things like "There's Don and that's Pattie
but I don't know when Tom was last seen." or "Who in hell is VVK and
where does line 63 go?"

I wrote Finger and developed the supporting database to provide this
information in traditional human terms -- real names and places.  Because
I preferred to talk face to face rather than through the computer or
telephone, I put in the feature that tells how long the terminal had been
idle, so that I could assess the likelihood that I would find them there
if I walked down the hall.

The program was an instant hit.  Some people asked for the Plan file
feature so that they could explain their absence or how they could be
reached at odd times, so I added it.  I found it interesting that this
feature evolved into a forum for social commentary and amusing
observations.

Finger was picked up by a number of other groups with DEC-10 computers
that were connected to Arpanet -- software flowed in all directions around
the net in those days.  It later migrated to Un*x, probably via U.C.
Berkeley.  Somewhere along the line the idea arose to provide a network
Finger service.  I don't remember who suggested that but it seemed like a
good idea at the time so I stuck it in.  Some other anxious people wanted
to be able to verify that their mail was delivered to specific addressees,
so the Mail feature was also added.
```

```
While I was somewhat surprised by the popularity of Finger, it has not
been as successful as an earlier program that I invented -- the spelling
checker.  It too was created to fill a personal need that many others
apparently share.  We didn't think about commercial development and
software protection in those days, but if we had we probably could have
made something out of it.  On the other hand, I enjoyed the camaraderie
of those gentler times and have no regrets.

        -Les Earnest (Les@Sail.Stanford.edu)
```

Now that personal computers can participate in DECnet networks (using DEC's PCSA product set) it has become feasible to implement the FINGER command on these systems so that PC users can determine who is logged on to the larger computers in their network, see if they have mail waiting on any system, etc.

Intended Audience

This manual is intended for end-users of IBM-compatible personal computers running the MS-DOS operating system. The user is assumed to have a functional PCSA configuration installed on their PC, or to be able to install one using the DEC documentation. PCSA installation details are beyond the scope of this manual, although some information is supplied to help configure FINGER.

Document Structure

This document consists of three chapters and two appendices.

Chapter 1	Contains pre-installation information.
Chapter 2	Describes the installation procedure.
Chapter 3	Describes using MS-DOS FINGER.
Appendix A	Contains a listing of a sample installation.
Appendix B	Describes rebuilding MS-DOS FINGER from the source files.

Related Documents

You can find additional information in the following documents:

- *MS-DOS FINGER Release Notes* contain information and updates not included in this manual. The release notes are part of the software distribution kit.
- The user documentation that accompanies the MS-DOS operating system. This material was provided by your computer supplier as part of your computer system.
- The documentation supplied with the PCSA software, particularly the *Getting Started*, *User's Guide*, and *Programmer's Reference* manuals.

1 Preparing to Install MS-DOS FINGER

1.1 Prerequisite Software

MS-DOS FINGER requires the MS-DOS operating system version 3.2 or later and PCSA version 3.0 or later to run. Earlier versions may function but have not been tested.

1.2 Accessing the Online Release Notes

MS-DOS FINGER provides online release notes, which you can display or print (once the software has been installed) with the following commands:

To display: TYPE FINGER.REL

To print: COPY FINGER.REL PRN

1.3 Installation Procedure Requirements

Before installing MS-DOS FINGER, ensure that the following requirements are met:

- Operating System Version

  MS-DOS FINGER V1.0 runs under MS-DOS version 3.2 or later.

- Layered Product Versions

  MS-DOS FINGER has been tested with PCSA V3.0. It may also function with other versions of PCSA.

- If you are installing MS-DOS FINGER on a remote boot disk, privilege to modify files on the remote boot disk.
- Approximately 10 minutes.
- Approximately 175,000 free bytes of disk space for the installation. After installation the space used can be reduced to approximately 20,000 bytes by removing the on-line documentation and source code.
- If you received your distribution in a form other than PC-readable media, suitable tools to convert it to PC format.
- If you desire to rebuild the software (not normally necessary), a copy of the Microsoft C compiler, version 5.1 and a copy of the Microsoft Macro Assembler, version 5.1.

2 Installing MS-DOS FINGER

This document assumes that the MS-DOS FINGER distribution media is located in the A: diskette drive. If that is not true on your system, please substitute the appropriate drive and path where necessary.

This document assumes that you are familiar with a text editor. The EDLIN editor provided with MS-DOS is used in the examples, however any editor capable of producing simple ASCII text files can be substituted as appropriate. If you don't know how to use any editor, refer to the EDLIN documentation supplied with your copy of MS-DOS.

2.1 Determining a Location for MS-DOS FINGER

You should first determine where to locate MS-DOS FINGER. If your PC boots from its own hard disk, you should install MS-DOS FINGER there. If your PC boots from a PCSA disk service, you may wish to install MS-DOS FINGER on the disk service, especially if other PCs boot from the same disk service and they wish to use MS-DOS FINGER as well. If you decide to install to a PCSA disk service, consult the PCSA *Server Administration with Commands* manual for more information.

If you install MS-DOS FINGER on your PC's hard disk, you should select a directory in the MS-DOS **PATH**, so that you will be able to use the FINGER command regardless of the current directory. To determine the current PATH value, issue the PATH command:

```
C:\>path ENTER
PATH=C:\DOS;C:\QEMM;C:\SYSTEM;C:\BIN;C:\UNIX;C:\DECNET
```

In this example, any of the listed directories would work. However, since MS-DOS FINGER is a DECnet application, we will install it in the C:\DECNET directory. Similarly, you should choose a list from the PATH on your system.

2.2 Copying the Distribution Files

Now, use the MS-DOS **COPY** command to place a copy of the program and related documentation in the selected directory:

```
C:\>copy a:finger*.* c:\decnet ENTER
A:FINGER.EXE
A:FINGER.REL
A:FINGER.TXT
A:FINGER.PS
A:FINGER.C
        5 file(s) copied
```

2.3 Setting the Environment Variables

MS-DOS provides a feature called the **ENVIRONMENT** to store various settings. If you desire, you can store two configuration settings for FINGER there.

The **FINGROUT** environment variable defines a ‘‘default router’’ to be used if a node is unknown to your PC. In many configurations, the DECnet database in your PC only has information on a subset of the nodes actually reachable on your network. Since FINGER can only access nodes that exist in the PC’s DECnet database, this will prevent you from accessing the other nodes on your network. However, one or more of the nodes known to your PC may be able to route requests to other nodes which are not known by your PC. If you contact your network manager or system administrator and ask them for the name of a network router known to your PC, you can store that information in the FINGROUT environment variable. Subsequent FINGER requests to nodes unknown to your PC will be routed through the network router. The following session shows the addition of the FINGROUT environment variable for a default router named SPCVXA:

```
C:\>edlin autoexec.bat [ENTER]
End of input file
*1l [ENTER]
       1:*@echo off
       2: path=c:\dos;c:\qemm;c:\system;c:\bin;c:\unix;c:\decnet
       3: prompt $p$g
*4i [ENTER]
       4:*set fingrout=SPCVXA [ENTER]
       5:*^Z
*exit [ENTER]
```

The second environment variable, **FINGUSER**, defines the username FINGER should supply when the short-hand username ‘‘.’’ is given. This is a substitute for giving the full username when you are asking for information about your account on another system. Most FINGER implementations can determine this automatically, but since the PC has no concept of usernames, one must be fabricated. For example, if your username was CS4321AB, you could preset it in the FINGUSER environment variable:

```
C:\>edlin autoexec.bat [ENTER]
End of input file
*1l [ENTER]
       1:*@echo off
       2: path=c:\dos;c:\qemm;c:\system;c:\bin;c:\unix;c:\decnet
       3: prompt $p$g
       4: set fingrout=SPCVXA
*5i [ENTER]
       5:*set finguser=CS4321AB [ENTER]
       6:*^Z
*exit [ENTER]
```

2.4 Completing the Installation

You should now view the *MS-DOS FINGER Release Notes*, found in the file FINGER.REL, for additional information not contained in this manual. Copies of this manual in both ASCII form (file FINGER.TXT) and PostScript form (file FINGER.PS) are provided for your convenience. The source code to the MS-DOS FINGER program is also provided (file FINGER.C). If you have no use for these files, you may delete them to conserve space on your disk drive.

3 Using MS-DOS FINGER

3.1 Invoking MS-DOS FINGER

All FINGER options are given on the command line. If you issue the FINGER command with the option */?*, help information will be displayed, as shown in the following example:

```
C:\>finger/? ENTER
Displays information about users on remote computers.

FINGER [username]@hostname [options]

  username  A user specification or wildcard in the remote host's
            syntax.
  hostname  The DECnet node name or address of the remote host.
  options   Command options, passed to the remote host. Use the
            command FINGER @hostname /HELP for more information.

A default routing host name may be preset in the FINGROUT envi-
ronment variable. See the documentation for more information.
```

If you don't specify a hostname on the command line, you will receive a message showing the FINGER program's version and creation date:

```
C:\>finger ENTER
MS-DOS Finger Client V1.0-02 - 22-Jan-91

You are the only user, of course.
```

If you specify a hostname, MS-DOS FINGER will establish a DECnet connection to the requested system and obtain the requested information. For example:

```
C:\>finger @spc11z ENTER
[SPC11Z.DECnet]

St. Peter's College - RSTS SIG Newsletter System
SPC11Z PDP-11/83, RSTS V10.0-L, Tuesday, 22-Jan-1991 03:52, 8 Jobs, 25 Max.
Uptime 14 02:35:29, since Tuesday, 8-Jan-1991 01:17

04-Jun-90 - The 1989 RSTS SIG tape is now online in [89,*]

Job Username       PPN   Progrm Term   Login  CPU  ST Location         TTType
 1  SYSTEM         1,2   ERRCPY Det          00:20 SR - Detached -
 2  SYSTEM         1,2   NPKDVR Det          00:25 SL - Detached -
 3  SYSTEM         1,2   OMS    Det          00:25 SL - Detached -
 4  SYSTEM         1,2   PBS... Det          00:31 SL - Detached -
 5  SYSTEM         1,2   EVTLOG Det          01:45 SL - Detached -
 6  SYSTEM         1,2   MOPSRV Det          00:03 XH - Detached -
 7  SYSTEM         1,2   MAILQ  Det          00:09 SR - Detached -
 8  SYSTEM         1,2   FINSRV Det          00:01 RN - Detached -
```

If you specify a username as well as a hostname, MS-DOS FINGER will ask the remote host about that particular user:

```
C:\>finger terry@spc11z ENTER
[SPC11Z.DECnet]

SPC11Z RSTS/E, Tuesday, 22-Jan-1991 03:52

TERRY [1,254] (Terry Kennedy) is not logged in.
```

```
Default directory: _SY0:[1,254]
Last login: Sunday, 20-Jan-1991 15:35 from KB17:
Mail is forwarded to: SPCVXA::TERRY
Plan:

Operations Manager, Academic Computer Center

Office Hours: Varied and unenviable. Late afternoons and
evenings, usually. Call first to make sure I'm in.

Phone: (201) 435-0252
```

If you specify the username as a single period and have defined the **FINGUSER** environment variable, MS-DOS FINGER will use the contents of the environment variable as the username. This provides a convenient shorthand if you have a long username.

Some Finger implementations allow the use of options to control their operation. The options allowed depend on the remote FINGER program in use. Many remote FINGERs support the **/HELP** option to return help information, including the list of supported options. If you wish to specify options on the FINGER command line, you should place them after the hostname, separated by at least one space, as shown in the following example:

```
C:\>finger terry@spc11z /noplan ENTER
```

3.2 Error Messages

MS-DOS FINGER can return several error messages. All FINGER error messages begin with the prefix **?FINGER:** and then have a detailed message text. Some of the more common messages are:

?FINGER: Must specify nodename: FINGER [username]@nodename

You specified a username without giving a nodename. Since usernames have no meaning on an MS-DOS system, the command cannot complete.

?FINGER: Insert space before options

You specified options immediately following the nodename. Re-issue the command with a space between the nodename and the options.

?FINGER: DECnet not started

The PCSA software has not been started. Network services are not available.

?FINGER: Network send: Unrecognized object

The remote node does not support the FINGER command.

?FINGER: Network send: Unrecognized node name

You have specified a nodename that DECnet on your PC does not recognize.

?FINGER: Network send: Node unreachable

There is currently no network path available between your PC and the remote system. Try again later. If the problem persists, contact your local network manager for more assistance.

?FINGER: Network read: Timed out

The network operation did not complete within 45 seconds. If your network is very busy, retry the command later. This condition can also result from hardware problems or configuration errors with the PCSA software. If this problem persists, contact your local network manager for more assistance.

?FINGER: Network send: message-text
?FINGER: Network select: message-text
?FINGER: Network read: message-text
?FINGER: Network close: message-text

Other messages indicate a problem with MS-DOS FINGER, or with the PCSA software on your machine. If the problem persists, file a problem report.

3.3 Reporting Problems and Obtaining Updates

While MS-DOS FINGER has been extensively tested, problems can arise due to unforseen configurations, new versions of related software, or oversights. If you find a problem in MS-DOS FINGER, please contact the author and report it using one of the following methods:

Network mail:

```
terry@spcvxa.spc.edu              (Internet)
terry@spcvxa.bitnet               (BITNET)
...!rutgers!njin!spcvxb!terry     (UUCP)
```

Regular mail:

```
Saint Peter's College
Academic Computer Center
2641 Kennedy Blvd.
Jersey City, NJ 07306 USA
Attn: Terry Kennedy
```

By telephone:

```
+1 201 435 0252
```

MS-DOS FINGER and other FINGER implementations are discussed on the info-finger network mailing list. Announcements of new versions, bug fixes, etc. are also made via this mailing list. To subscribe, send a network mail message to:

```
i-finreq@spcvxa.spc.edu           (Internet)
i-finreq@spcvxa.bitnet            (BITNET)
...!rutgers!njin!spcvxb!i-finreq  (UUCP)
```

New versions of MD-DOS FINGER and other FINGER implementations are available from the SPC fileserver and the RSTS/E SIG Bulletin Board. To access the SPC fileserver, send a network mail message with the text (**not** subject) HELP to:

```
vmsserv@spcvxa.spc.edu            (Internet)
vmsserv@spcvxa.bitnet             (BITNET)
...!rutgers!njin!spcvxb!vmsserv   (UUCP)
```

Using MS-DOS FINGER

To access the RSTS/E SIG Bulletin Board, dial +1 201 915 9361 with a modem capable of operating at a speed of 300, 1200, or 2400 baud. You may also call with a US Robotics Courier HST modem for 9600 baud access. When connected, you will receive further instructions.

A Sample MS-DOS FINGER Installation

The following is a sample installation of MS-DOS FINGER:

```
C:\>path [ENTER]
PATH=C:\DOS;C:\QEMM;C:\SYSTEM;C:\BIN;C:\UNIX;C:\DECNET

C:\>copy a:finger*.* c:\decnet [ENTER]
A:FINGER.EXE
A:FINGER.REL
A:FINGER.TXT
A:FINGER.PS
A:FINGER.C
        5 file(s) copied

C:\>edlin autoexec.bat [ENTER]
End of input file
*1l [ENTER]
       1:*@echo off
       2: path=c:\dos;c:\qemm;c:\system;c:\bin;c:\unix;c:\decnet
       3: prompt $p$g
*4i [ENTER]
       4:*set fingrout=SPCVXA [ENTER]
       5:*^Z
*exit [ENTER]

C:\>edlin autoexec.bat [ENTER]
End of input file
*1l [ENTER]
       1:*@echo off
       2: path=c:\dos;c:\qemm;c:\system;c:\bin;c:\unix;c:\decnet
       3: prompt $p$g
       4: set fingrout=SPCVXA
*5i [ENTER]
       5:*set finguser=CS4321AB [ENTER]
       6:*^Z
*exit [ENTER]
```

B Rebuilding MS-DOS FINGER From Source

If you desire to make changes to MS-DOS FINGER, or if you wish to rebuild the program for other reasons, you may rebuild it as shown below. You will need the PCSA function library, which is an optional component of the PCSA product. Consult the PCSA *Installing the DECnet PCSA Client* manual for details on installing the function library.

The PCSA function library must be compiled and built before it can be used. As the PCSA documentation is essentially useless for explaining this process, it will be described here. It is assumed that the programming library file (and **only** those files) are in a subdirectory called NETSRC.

Edit the file REDIRCT.ASM to correct a syntax error, as shown here:

Before editing:

```
        jmp     end             ;Return
error:  mov     ax,0            ;An error occured
end:    RETURN
```

After editing:

```
        jmp     fnd             ;Return
error:  mov     ax,0            ;An error occured
fnd:    RETURN
```

Edit the file PERROR.C to correct a conflict in the declaration of an argument to the perror( ) routine:

Before editing:

```
void perror(cp)
char    *cp;
```

After editing:

```
void perror(cp)
const char    *cp;
```

Configure the assembler modules for the small memory model by issuing the following command:

```
C:\NETSRC>copy ssize.h size.h ENTER
```

Configure the assembler modules for operation without Microsoft Windows by editing the file BEGIN.H:

Before editing:

```
; Size independent symbols
;
WINDOWS EQU 1       ; Windows support
;WINDOWS EQU 0        ; No Windows support
```

After editing:

```
; Size independent symbols
;
;WINDOWS EQU 1       ; Windows support
WINDOWS EQU 0        ; No Windows support
```

Create the file BUILDLIB.RES, containing the following lines:

```
netlibsn
y
+BCMP&
+BCOPY&
+DNETACCE&
+DNETADDR&
+DNETALIA&
+DNETCHKS&
+DNETCONN&
+DNETEOF&
+DNETHTOA&
+DNETINST&
+DNETMSC&
+DNETNCNT&
+DNETNTOA&
+DNETOTOA&
+DNETPASS&
+DNETPATH&
+DNETW&
+DNETWIN&
+FILEDIR&
+GETCNE&
+GETCNENI&
+GETDATE&
+GETSNENI&
+GETTIME&
+GKNONOD&
+GNODEADD&
+GNODEBAD&
+GNODEBNM&
+GNODEENT&
+GNODENAM&
+GREMNOD&
+HCLOSE&
+HCREATE&
+HDUP&
+HDUP2&
+HOPEN&
+HREAD&
+HSEEK&
+HTELL&
+HWRITE&
+BZERO&
+NERROR&
+PAUSEC&
+PERROR&
+PRGACC&
+PRGBND&
+PRGCLS&
+PRGCON&
+PRGGPN&
+PRGGSN&
+PRGGSO&
+PRGLSN&
+PRGRCV&
+PRGSEL&
+PRGSHD&
+PRGSIO&
+PRGSND&
+PRGSOC&
+PRGSSO&
```

```
+PTRCONV&
+PUTEXEC&
+SETDATE&
+SETTIME&
+CSREG&
+UPPER&
+DECNET&
+DNETSES&
+DSREG&
+FBCOPY&
+MSDOS&
+REDIRCT&
```

Set the **INCLUDE** environment variable to a search list with the current directory first. This is necessary because the PCSA function library sources have include commands for files with the same names as standard C compiler include files, and thus the local directory must have precedence over the standard directory. Assuming that the C compiler INCLUDE directory is C:\INCLUDE, and the function library directory is C:\NETSRC, the following command will accomplish this:

```
C:\NETSRC>set include=c:\netsrc;c:\include ENTER
```

Compile all the C source files with the Microsoft C compiler, version 5.1, by issuing the following command, observing the case sensitivity of the compiler command line:

```
C:\NETSRC>for %c in (*.c) do cl/c /AS /Zp /Ox %c ENTER
```

All of the C source files should compile without errors. Assemble all the assembler source files with the Microsoft assembler, version 5.1, by issuing the following command line:

```
C:\NETSRC>for %c in (*.asm) do masm %c; ENTER
```

All of the assembler source files should assemble without error. Create the programming library using the following command:

```
C:\NETSRC>lib @buildlib.res ENTER
```

You now have a small-model network interface library without Windows support in the file NETLIBSN.LIB. This library does not have to be re-created unless you install a newer version of PCSA.

To compile and link the MS-DOS FINGER program, assuming the FINGER source is located in directory C:\FINGER, be sure you have the INCLUDE environment variable set up as shown above and issue the following commands:

```
C:\NETSRC>cd \finger ENTER
C:\FINGER>cl/c /AS /Zp /Ox finger.c ENTER
C:\FINGER>link finger ENTER
(when prompted for a library, respond with NETLIBSN ENTER
```

You should not have received any errors during the above steps. At this point you have a new version of MS-DOS FINGER. You should test it for proper operation and then place it into service by replacing the version you installed from the distribution media.

Index

D

Disk Service • 2–1

E

Environment Variable
  FINGROUT • 2–2
  FINGUSER • 2–2
  INCLUDE • B–3
  PATH • 2–1
Error messages • 3–2

F

FINGROUT Environment Variable • 2–2
FINGUSER Environment Variable • 2–2

H

Hard Disk • 2–1
Hostname • 3–1

I

INCLUDE Environment Variable • B–3
Installation • 2–1, A–1

L

Location
  Of Files • 2–1

M

Mailing list • 3–3
Messages
  Error • 3–2
MS-DOS
  PATH • 2–1
  Required Version • 1–1

O

Options • 3–1, 3–2
Origins • v

P

PATH Environment Variable • 2–1
PCSA
  Disk Service • 2–1
  Required Version • 1–1
Problems
  Reporting • 3–3

R

Rebuilding • 1–1, B–1
Release Notes • vi, 1–1, 2–3
Reporting problems • 3–3
Requirements • 1–1

U

Updates • 3–3
Username • 3–1
  Shorthand for • 3–2