

Macintosh. EtherTalk for A/UX 2.0

ď.

Alpha draft • APPLE CONFIDENTIAL

May 10, 1989 • Kristi Fredrickson • x47162

Developer Technical Publications

• Apple Computer, Inc. 1989

APPLE COMPUTER, INC.

This manual is copyrighted by Apple or by Apple's suppliers, with all rights reserved. Under the copyright laws, this manual may not be copied, in whole or in part, without the written consent of Apple Computer, Inc. This exception does not allow copies to be made for others, whether or not sold, but all of the material purchased may be sold, given, or lent to another person. Under the law, copying includes translating into another language.

The Apple logo is a registered trademark of Apple Computer, Inc. Use of the "keyboard" Apple logo (Option-Shifi-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

© Apple Computer, Inc., 1989 20525 Mariani Avenue Cupertino, CA 95014-6299 (408) 996-1010

Apple, the Apple logo, AppleShare, AppleTalk, Apple IIGS, MacApp, Macintosh, and ProDOS are registered trademarks of Apple Computer, Inc.

A/UX, APDA, AppleCD SC, APW, EtherTalk, Finder, GS/OS, HyperCard, HyperTalk, MPW, and MultiFinder are trademarks of Apple Computer, Inc.

IBM is a registered trademark of International Business Machines Corporation. Microsoft and MS-DOS are registered trademarks of Microsoft Corporation.

LightSpeedC is a trademark of Symantec Corporation.

ITC Zapf Dingbats is a registered trademark of International Typeface Corporation.

Linotronic is a registered trademark of Linotype Co.

POSTSCRIPT is a registered trademark, and Illustrator is a trademark, of Adobe Systems Incorporated.

Simultaneously published in the United States and Canada.

Draft 5/10/89 EtherTalk 2.0 Release Notes

Note to reviewers: Standard release note format to be used in final draft.

1. New Features

EtherTalk 2.0 for A/UX extends the power of AppleTalk by enabling network users to share AppleTalk LaserWriter and ImageWriter printers over an Ethernet network. Need information here regarding bridge/router requirements for printing. These 2.0 enhancements were accomplished by modifying the pre-existing protocol stack to implement the features of Extended AppleTalk for EtherTalk, while maintaining support for LocalTalk through the same stack.

The EtherTalk 2.0 includes changes to the library interface in order to provide access to protocols through library calls rather than through direct system calls. This enhancement makes user programs portable, and eliminates the need for the AppleTalk and Extended AppleTalk Link Access Protocols (ALAP and ELAP). The accompanying man pages give details regarding these and other technical changes.

2. Installing EtherTalk 2.0

Installation of EtherTalk 2.0 involves loading three floppy disks onto your system. One of these disks contains the EtherTalk software, and the other two contain the modified Toolbox. See the revised Chapter 10 of the A/UX Network System Administration, enclosed, for detailed installation instructions. You have the option of de-installing EtherTalk to restore the old version of Appletalk, recovering the original LocalTalk and Toolbox files.

Note, however, that if you are one of the test sites that installed a previous version of EtherTalk, some of the LocalTalk files are probably already lost, and it isn't possible to recover them.

EtherTalk 2.0 Release Notes

5/10/89-1

3. Updating your manual set

This section describes how to update your existing AppleTalk documentation with the EtherTalk Version 2.0 revisions. You might want to save the outdated version of the documentation if there is a possibility that you might later decide to de-install EtherTalk and return to the old LocalTalk release.

Replace Chapter 10, "Installing and Administering AppleTalk" in A/UX Network System Administration with the enclosed Chapter 10. This chapter now applies to both LocalTalk and EtherTalk, so it will still be valid even if you decide to de-install EtherTalk and return the previous 1.0 version of AppleTalk.

Update the A/UX Command Reference, the A/UX Programmers Reference, and the A/UX System Administrators Reference with the enclosed revised man pages.

Note to reviewers: In final Release Notes, will itemize which man pages go in each document.

These are the revised pages:

```
Cat_cho_prn(1)
atprint(1) (Formerly at_printer)
at_status(1)
atlookup(1) (Formerly at_nwelkup)
appletalk(1M)
newunix(1M)
atp(3N)
ddp(3N)
lap(3N)
nbp(3N)
pap(3N)
rtmp(3N)
zip(3N)
```

If there is a chance that you might decide to de-install EtherTalk and restore LocalTalk, save the old man pages so that you can restore the documentation to the LocalTalk version.

5/10/89-2

A/UX 1.1 Release Notes

Note to reviewers: The initial release of A/UX 1.1 Network Applications Programming is not yet final, so EtherTalk 2.0 updates to that document are not included here.

Chapter 10 Installing and Administering AppleTalk

Contents

1.	The AppleTalk Network System under A/UX	1
2.	Installing and setting up LocalTalk	2
	2.1 Setting up a LocalTalk network	2
	2.2 Installing AppleTalk with a LocalTalk card under	
	A/UX	2
3.	Installing and setting up AppleTalk with an EtherTalk card	
	under A/UX	3
	3.1 Installation requirements	3
	3.2 Installation procedure	3
	3.3 De-installation	5
4.	Using AppleTalk to print in A/UX	5
	4.1 Setting up the printer spooler	6
	4.2 Printing files with 1p	8
	4.3 Printing files with atprint and	
	at printer	9
		10
	4.5 Printing files created by a Macintosh	
	•	12
	••	13
5.	Summary of AppleTalk commands	13
	• ••	14

Pro Control of the Co • ***** • - **ii -**

Chapter 10

Installing and Administering AppleTalk

1. The AppleTalk Network System under A/UX This chapter describes the A/UX implementation of AppleTalk and is designed for network administrators who want to set up an AppleTalk network.

A network system is a communication environment in which network devices and software observe a common set of rules for communicating. These rules are called network protocols, and they explicitly describe each step in the process of interaction between the network devices.

The AppleTalk network system includes a set of protocols, each of which governs a different aspect of the communication process, such as how network devices are identified, and how data is formatted for transmission. AppleTalk protocols can be implemented by a wide variety of different devices and transmission media. While all AppleTalk networks use AppleTalk protocols, they do not all use the same transmission standards, media, or connections.

The design of AppleTalk allows you to select the network type that best suits the needs of your organization, while retaining the same AppleTalk services and interface throughout the internet. Because of the high processing requirements of both AppleTalk and the A/UX multitasking operating system, AppleTalk under A/UX requires a coprocessing card. For the low-cost, medium speed LocalTalk cable system, a LocalTalk card is required. You also have the option of connecting your A/UX Macintosh to the higher performance Ethernet network, using an EtherTalk card. Contact your Apple representative for a list of cards that are available from various vendors.

Here is a diagram that illustrates the differences between LocalTalk and EtherTalk:

Picture to be added

In this chapter, references to AppleTalk apply to both LocalTalk and EtherTalk, unless specifically stated otherwise.

AppleTalk gives you the capability of putting your A/UX system on a network of computers for the purpose of sharing AppleTalk LaserWriter printers, ImageWriter printers, and other printers available via Ethernet. AppleTalk under A/UX can work concurrently with the B-NET, NFS, and Yellow Pages software described in the previous chapters of this manual.

2. Installing and setting up LocalTalk

This section covers installation and setup of LocalTalk.

2.1 Setting up a LocalTalk network

Instructions for setting up an AppleTalk network with LocalTalk cables and connectors can be found in LocalTalk Cable System Owner's Guide. Check the documentation that accompanies your card for any further cabling requirements.

2.2 Installing AppleTalk with a LocalTalk card under A/UX

For instructions on installing AppleTalk under A/UX, refer to the owner's guide that accompanies your LocalTalk card. Briefly, installation generally consists of the following steps:

- Shut down A/UX and use the powerdown command to turn off the power to your computer.
- 2. Remove the cover of your computer.
- 3. Insert the AppleTalk card into an available NuBus™ slot.
- 4. Replace the cover.
- 5. Attach a LocalTalk cable to the connector on the back of the AppleTalk card.
- 6. Restart your computer and bring A/UX up to single-user mode.
- 7. Install the software from any accompanying 3.5-inch disks.
- 8. Run the /etc/newunix and autoconfig commands to build a new kernel that integrates the device driver for the

AppleTalk card.

9. Reboot A/UX and bring it back to multi-user mode.

Specific, detailed installation instructions are included in the owner's guide for your AppleTalk product.

3. Installing and setting up AppleTalk with an EtherTalk card under A/UX

This section describes how to install EtherTalk 2.0 (A/UX). The installation procedure saves the old LocalTalk and ToolBox files in a safe place, and you have the option of de-installing EtherTalk to restore the old version of AppleTalk, recovering the original LocalTalk and ToolBox files.

3.1 Installation requirements

These are the requirements for installing EtherTalk 2.0:

- BNET kernel already installed on your system, as described in Chapter 2 of A/UX Network System Administration
- A/UX Version 1.1 already installed on your system, as described in the A/UX Installation Guide. You can verify the A/UX release version by entering the command /etc/releaseID.
- Superuser privileges

3.2 Installation procedure

Follow these steps to install EtherTalk 2.0:

- 1. Log in as root
- 2. Enter this command to get a full listing of the currently active processes:

A display similar to this appears:

PID	TTY	TIME	COMMAND
0	?	0:00	swapper
1	?	0:01	init
2	?	0:00	vhand
62	?	0:00	toolboxd

3. Shut down the /etc/toolboxdaemon process with the kill command. For the above example, enter this command:

kill 62

If EtherTalk is running, shut it down also.

4. Change to the root directory by entering this command:

cd /

5. Place the EtherTalk floppy disk in Drive 0 (the right-hand drive on systems with two floppy disk drives), then enter this command:

cpio -iBvdu < /dev/rfloppy0</pre>

A series of messages appears on the screen as the system copies the /etc/install.d files.

6. When the system prompt reappears, enter this command to eject the EtherTalk floppy disk:

eject 0

Place the ToolBox #1 disk in Drive 0 and again enter the cpio command:

cpio -iBvdu < /dev/rfloppy0</pre>

A series of messages appears as the system copies the toolbox files.

- 8. When the system has finished copying the ToolBox #1 files, it ejects the floppy disk and prompts you to insert the ToolBox #2 disk. Remove the ToolBox #1 disk and insert the ToolBox #2 disk
- 9. When the system ejects the ToolBox #2 disk, remove it and enter this command:

/etc/newunix ethertalk

10. When the system prompt re-appears, enter this command:

/etc/autoconfig -I -M /etc/master -S /etc/startup

This command may take several minutes to complete.

A/UX Network System Administration 030-3366-A

11. In /etc/inittab, change the line

Cat0:2:off:/etc/at_nbpd >/dev/syscon 2>&1

to

at0:2:0mce:/etc/at_nbpd >/dev/syscon 2>&1

If you already have AppleTalk or LocalTalk on your system, the /etc/inittab file may already include the above change.

12. Use shutdown, then restart your system to complete the installation.

3.3 De-installation

To deinstall EtherTalk and return to LocalTalk, enter this command at the system prompt:

/etc/newunix noethertalk

The rest of this chapter describes how to print on an AppleTalk network in A/UX.

4. Using AppleTalk to print in A/UX

When you use AppleTalk to print in A/UX, you can print files using any of these three methods:

- Using the 1p spooler to print from an A/UX shell. To use
 AppleTalk with the 1p spooler, follow the directions in the
 following section. After setting up the 1p spooler, users can use
 the at_cho_prn command to choose their default AppleTalk
 destination printer.
- Printing from an A/UX shell using the atprint command if you are working under EtherTalk or the at_printer command if you are working under LocalTalk. The atprint and at_printer commands bypass the lp spooler and send output directly to the printer.
- Using Print from the File menu and the Printer dialog box to print from a Macintosh application. You must first select a default printer using the Chooser from the Apple menu. When printing from a Macintosh application, you can print files as you normally would when using the Macintosh OS. If you are going

Installing and Administering AppleTalk 050-5366-A

to print files from Macintosh applications only, you do not have to configure or use the 1p spooler.

Your computer is set up so that an ImageWriter II (iw2), functioning as a serial device, is the default 1p printer. To change the default 1p printer to an AppleTalk printer, follow the steps in the following sections.

Note: To use an ImageWriter II with AppleTalk, you need a special conversion kit (see your Apple representative for details). Without this kit, an ImageWriter II can function only as a serial printer.

To use a LaserWriter with AppleTalk, change the switch setting on the LaserWriter printer from serial to AppleTalk.

4.1 Setting up the printer spooler

To use the A/UX 1p spooler to print jobs across the AppleTalk network, you must provide certain information to 1p. Follow these steps:

- 1. Log in as the root user or as 1p.
- 2. Move to the printer spooler directory by entering

cd /usr/spool/lp

3. Use the ADD_AT printer-name command to set up printing on the AppleTalk network. For example, enter the command

This command shuts down your printer spooler, configures it to print to all LaserWriter and ImageWriter printers on the AppleTalk network, sets the default destination printer to atalk_printer, and then restarts the printer spooler. After you give the ADD_AT atalk_printer command, the following messages appear on your screen:

scheduler stopped /usr/lib/lpadmin: destination "atalk_printer" non-existent destination "atalk_printer" now accepting requests destination "AppleTalk" now accepting requests printer "atalk_printer" now enabled IP service atalk_printer now installed

Note that the ADD_AT command sets the default destination to the printer name specified (in this example, atalk_printer). When atalk_printer is the default printer and you use lp or atprint or at_printer to print a file, your file will be printed on the first available AppleTalk printer in your current AppleTalk zone, unless you have chosen a default printer with the at_cho_prn command. See "Choosing a printer for each user" for a description of the at_cho_prn command.

If you do not want the default destination set to atalk_printer, "comment out" one line; that is, put a number sign (#) as the first character in the line in the ADD_AT script. Change the following line from

/usr/lib/lpadmin -d\${Printer}

to

/usr/lib/lpadmin -d\${Printer}

This allows you to use a serial printer as the system default printer.

To verify that the printer spooler is running, enter

lpstat -t

The response should be similar to this:

scheduler is running
system default destination: atalk_printer
members of Class AppleTalk:
 atalk_printer

device for iw2: /dev/iw2
device for atalk_printer: /usr/spool/lp/transcript/atalk_printer-log
iw2 accepting requests since Jan 14 11:08
atalk_printer accepting requests since Jan 20 11:32
AppleTalk accepting requests since Jan 20 11:32

printer iw2 is idle. enabled since Jan 14 11:08 printer stalk_printer is idle. enabled since Jan 20 11:32

Each user can now use lp, atprint, or at_printer to send files to a printer on an AppleTalk network, and at_cho_prn to set a specific default AppleTalk printer for that user.

4.2 Printing files with 1p

You can print to either LaserWriter or ImageWriter printers using 1p. Note that ImageWriter printers expect ASCII input and LaserWriter printers expect PostScript input. If 1p receives ASCII input and a user has not specified an ImageWriter as the default printer choice (using at_cho_prn) the interface file for the AppleTalk printer class invokes commands which automatically perform PostScript conversion.

If you will be using the 1p spooling system to print ASCII or PostScript files on LaserWriter printers attached to the AppleTalk network, use either of these commands:

lp file
or cat file | lp

To use the first command, your file must be readable by 1p. Use the second command (cat file | lp) if the file access permissions of your file do not give read permission to group or others.

To print formatted troff output files on LaserWriter printers, use the command

troff -Tpsc file | psdit | lp

A/UX Network System Administration 030-3366-A

psroff file

When you use the psroff command to format text, the file is automatically run through psdit and sent to lp.

To use the 1p spooler to print ASCII files on ImageWriter printers attached to the AppleTalk network, you must first choose an ImageWriter as your default printer using the at_cho_prn command. After selecting an ImageWriter printer, use the command

cat file | lp

When you print using lp, your file will be sent to the printer you chose with at_cho_prn. If you did not use at_cho_prn to choose a printer and if the class AppleTalk is your default destination, your file will be printed on the first available LaserWriter printer in your current AppleTalk zone.

See cat(1) and lp(1) in A/UX Command Reference for more details on using these commands. See /usr/lib/ps for the Adobe PostScript programs available with A/UX.

4.3 Printing files with atprint and at_printer
The EtherTalk atprint command and the LocalTalk at_printer
command send data directly to a Printer Access Protocol (PAP) server,
such as a LaserWriter or ImageWriter. This script is useful when there is a
single user on a system or when the 1p spooler is not configured.

The atprint and at_printer scripts are faster than the spooler because they send files directly to the output device. However, with the spooler, print requests from an individual user can be queued; in contrast, with atprint and at_printer, requests can be issued only one at a time.

The following three examples are for printing under EtherTalk; to print under A/UX on a LocalTalk printer, substitute at_printer for atprint:

To print troff output files to a LaserWriter under EtherTalk, use this command:

```
troff -Tpsc file | psdit | atprint
```

To print plain ASCII text to a LaserWriter under EtherTalk, use this command:

```
enscript -p- file | atprint
```

To print plain ASCII text to an ImageWriter under EtherTalk, use this command:

```
atprint file
```

All of the above commands send file to the default printer chosen with at_cho_prn. If you did not use at_cho_prn to choose a printer and if the class AppleTalk is your default destination, your file will be printed on the first available LaserWriter printer in your current AppleTalk zone.

4.4 Choosing a printer for each user

Each user may choose his or her own default printer for the 1p spooler or atprint or at_printer commands by using the at_cho_prn command. See "Printing Files Created by a Macintosh Application" for information on choosing a default printer for Macintosh applications.

Here is the procedure for using at_cho_prn to choose the default printer:

- 1. Check that the LaserWriter or ImageWriter is turned on.
- 2. Enter this command:

at cho prn

In response, the system first displays a list of available AppleTalk zones. If you have only one AppleTalk zone, then the list of printers in that zone is shown immediately. If more than one AppleTalk zone is available to you, you will see a display similar to this one:

Zone List:

- 1: Twilight zone
- 2: The Land of Oz
- 3: Never never zone

ZONE number (0 for current zone)?

Choose a number corresponding to the AppleTalk zone from which

you wish to select a printer. For example, to select a printer from the Never never zone, enter

3

After you choose an AppleTalk zone, you will see a list of all the printers in the zone. For example, after you select a zone, you see information about every printer in that AppleTalk local zone:

ITEM	OBJECT	TYPE	ZONE	MET	MODE	SOCKET
1:	MrEd	LaserWriter	*	0x1111	0x11	0xca
2:	Flipper	LaserWriter	*	0x1111	0x12	0xcb

ITEM number (0 to make no selection)?

Enter a number corresponding to the printer you wish to make the default when you send files to printers on the AppleTalk network. Enter 0 if you do not want to change your current printer default designation. For example, to select the printer with the name MrEd, enter:

1

for ITEM 1. This creates an entry in the file /usr/lib/PrinterChoices with your user identification number (user ID) and the chosen default printer.

The headings displayed by at_cho_prn for each printer mean the following:

ITEM	An identification number for the entire entry.
OBJECT	The printer's name.
TYPE	The type of printer, either LaserWriter or ImageWriter.
ZONE	The zone in which the printer resides. An asterisk (*) indicates the local zone.
NET	The network number if more than one AppleTalk network is connected; a 4-digit hexadecimal number.
NODE	The printer's node number on the network; a 2-digit hexadecimal number. This number is assigned dynamically when the network is set up.

Installing and Administering AppleTalk oso-sses-A

10-11

SOCKET The socket number for this service.

If the printer to which you wish to send files is not in the list of printers, make sure that you have selected the correct zone and try entering the at_cho_prn command again. See at_cho_prn(1) in A/UX Command Reference for more information.

Note that the default printer designation you set with at_cho_prn does not affect the default printer designation for Macintosh applications. When you print from a Macintosh application, files are sent to the printer you selected from the Chooser.

4.5 Printing files created by a Macintosh application
If you have started a Macintosh application, you can print a file from the application just as you normally would from the Macintosh OS. First set your default destination printer by selecting Chooser from the Apple menu.

If there is only one zone on your AppleTalk network, choose a printer type by clicking a printer icon, and then choose a printer by clicking on a printer name. If there is more than one zone on your AppleTalk network, choose a printer type by clicking a printer icon, then choose an AppleTalk zone by clicking a zone name, and then choose a printer from that zone by clicking a printer name. Close the Chooser by clicking the close box in the top-left corner of the Chooser dialog box. This printer will remain your destination printer for all Macintosh applications until you change it again using the Chooser.

Once you have set your default destination printer from the Chooser, you can print the file you are working on by choosing Print from the File menu. When the Macintosh printer dialog box appears, click OK to send your file to the printer.

Note that the default printer designation you set with the Chooser does not affect the default printer designation you set with the at_cho_prn command.

If you have created a file in a Macintosh application and you have quit that application, you can print the file from the A/UX shell by entering

launch -p [-it] application file

where application is the name of the Macintosh program and file is the name of the file created with that program. Your file will be sent to the

printer you selected from the Chooser. For example, to print a MacPaint® document named picture1, enter this command:

launch -p MacPaint picturel

You will then see the Macintosh printer dialog box. Click OK to send your file to the printer. See Getting Started with A/UX for more information about printing Macintosh files from the A/UX Toolbox.

4.6 Printing files from term

You can print files to printers on the AppleTalk network from term. First set your default destination printer by selecting Chooser from the Apple menu.

If there is only one zone on your AppleTalk network, choose a printer type by clicking a printer icon, and then choose a printer by clicking on a printer name. If there is more than one zone on your AppleTalk network, choose a printer type by clicking a printer icon, then choose an AppleTalk zone by clicking a zone name, and then choose a printer from that zone by clicking a printer name. Close the Chooser by clicking the close box in the top-left corner of the Chooser dialog box. This printer will remain your destination printer for all Macintosh applications, including term, until you change it again using the Chooser.

Once you have set your default destination printer from the Chooser, you can print information from your terminal screen window by selecting the text you want to print and then choosing Print Selection from the File menu. You will then see the Macintosh printer dialog box. Click OK to send your file to the printer. See Getting Started with A/UX for more information about printing Macintosh files from term.

5. Summary of AppleTalk commands

at_cho_prn Chooses a user's default printer on the LocalTalk

network. See at_cho_prn(1) in A/UX Command

Reference.

at nbpd Maintains the local node's Name Binding Protocol table.

at_lookup Looks up network-visible entities (NVEs) registered on

the EtherTalk network. For LocalTalk networks, this command is at_nvelkup. See at_lookup(1) in

AIUX Command Reference.

Installing and Administering AppleTalk 630-3366-A

atprint (EtherTalk uses atprint, LocalTalk uses

at_printer) Copies data to a remote Printer Access Protocol (PAP) server. See atprint(1) in A/UX

Command Reference.

at_server Generic PAP server. See at_server(1) in A/UX

Command Reference.

at_status Returns the status of a PAP server. See

at_status(1) in A/UX Command Reference.

appletalk Configures and allows you to view AppleTalk network

interfaces. See appletalk(1M) in A/UX System

Administrator's Reference.

/etc/newunix Prepares for a new kernel configuration. See

newunix(1M) in A/UX System Administrator's

Reference.

6. Troubleshooting

This section discusses general error conditions that affect printing on an AppleTalk network. First determine whether the error indicates a hardware or software problem.

To identify hardware problems:

• Check that your LocalTalk or Ethernet cabling is correct. See AppleTalk Personal Network for information on how to fix cabling problems.

To identify network software problems:

• Use the appletalk -s command to observe nonzero values and to report AppleTalk network statistics. See appletalk(1M) in A/UX System Administrator's Reference.

To identify local software problems:

• Check the /etc/inittab file for the at0 entry. If your AppleTalk network appears to be down, edit /etc/inittab so that it contains this line:

at0:2:respawn:/etc/at_nbpd >/dev/syscon 2>&1 # set to "respawn" for AppleTalk

A/UX Network System Administration

(This appears on one line in /etc/inittab.)

If this line is missing from /etc/inittab or if the third field in this line is set to off, at_lookup (for EtherTalk) or at_nvelkup (for LocalTalk) will return this error:

at_lookup (or at_nvelkup): error in lookup (status = -1)
and at_cho_prn will return this error:

could not select default printer

• Check to see that the /dev/appletalk file exists. If you see this error message on your screen

NBPD open failed: No such file or directory then the at_nbdp daemon is either failing to respawn or is respawning too rapidly. Check to see that the /dev/appletalk file exists. If /dev/appletalk does not exist, create a new kernel with the autoconfig command and specify the -I option.

• Use at_cho_prn to verify that zones appear. If the Chooser or at_cho_prn command does not return a zone list in an internet environment, check to see that the bridge is up by entering

appletalk -s

A bridge and network number of zero indicate that the local bridge is down. Contact your AppleTalk system administrator for assistance.

Check to make sure the AppleTalk printer is available. If you
attempt to print to an AppleTalk LaserWriter or ImageWriter printer
and the request fails, make sure the printer is still available by
selecting it again with the at_cho_prn command.

NAME

appletalk — configure and view AppleTalk network interfaces

SYNOPSIS

appletalk [-d] [-i interface] [-n] [-r[RTS-attempts]] [-s] [-u] [-z] [-b interface]

DESCRIPTION

Use appletalk to configure and view AppleTalk network interfaces and the AppleTalk network. You can use appletalk at any time to view network interface parameters, or to bring an AppleTalk interface up or down. The current version of A/UX only supports the DDP interface defined in appletalkrc(4).

These are the available arguments:

-i interface	Defines the interface to configure or view; this			
	parameter is a string such as localtalk0 or			
	ethertalkO. The appletalk default is			
	the interface defined in appliet alkrc(4).			

Displays the AppleTalk interface initial and current node addresses. If the AppleTalk network has not been activated, these addresses will be zero. The initial address is the address that this interface first attempts to assign during the LAP (AppleTalk Link Access Protocol) dynamic node assignment phase. Once the system find a unique address, it saves that address and uses it as the initial node address of this interface. The current address is the unique LAP address assigned to this interface.

-u Brings online the interface specified in appletalkrc(4)).

Takes offline the interface specified in appletalkrc(4). You must be the superuser to use this flag option.

-s Displays statistics and error counts for this interface. If the AppleTalk network is active, DDP statistics and error counts will also be displayed.

-d

-z

-r RTS-attempts Displays the number of RTS attempts to make when transmitting an ALAP frame. Supplying

an optional numeric parameter sets the number of RTS attempts to RTS-attempts. You must be

the superuser to set RTS attempts.

This forces the command to ignore the zone name hint and assume there is no zone name available for the node at start-up. As a result, if there are multiple zones on the cable, the system displays a menu of valid zone names for the node. The system puts the node into the zone that you select. This option is applicable only to an EtherTalk (Extended AppleTalk) in-

terface.

-b interface The interface is the hardware interface to be associated with an EtherTalk interface; it is a

sociated with an Emerial interface; it is a string such as ae0. In order for this option to work, you must use it along with the -u option. This option is useful when the node has multiple EtherNet boards installed. It associates the EtherTalk interface with the specified hardware

interface. The default interface is ae0.

EXAMPLES

```
appletalk -i localtalk0 -u
appletalk -s
appletalk -i ethertalk0 -b ael -u
```

The first command brings the interface localtalk0 online; the second displays statistics and error counts for the interface specified in appletalkrc(4). The third brings online interface ethertalk0 on the hardware interface ael.

FILES

```
/etc/appletalk
/etc/appletalkrc
/dev/appletalk/ddp/socket
/dev/appletalk/lap/*/control
/etc/at_nbpd
```

SEE ALSO

appletalkrc(4), "Installing and Administering AppleTalk," in A/UX Network System Administration.

NAME

at_cho_prn — choose a default printer on the AppleTalk network for current user.

SYNOPSIS

at_cho_prn [type[@zone]]

DESCRIPTION

at_cho_prn displays a list of default printer selections, and saves the name of the printer that you select. The printer that you select with at_cho_prn becomes the default printer for the command atprint(1). The at_cho_prn command checks the network to determine which printers are registered on that network. If you don't enter the zone part of the argument on the command line, at_cho_prn lists all the zones in the internet and prompts you to choose a zone in which you'd like to select your default printer. The default is the current AppleTalk zone.

After you specify the zone, at_cho_prn lists the printers (of type type) available in that zone. If you don't use the type argument on the command line, at_cho_prn displays all entities of the types LaserWriter and ImageWriter.

Next, the system prompts you to select a printer by entering the appropriate number from the printer list display. The system stores information about this printer in /usr/lib/PrinterChoices, keyed by the user's ID.

EXAMPLE

The command:

at_cho_prn LaserWriter@*
produces output similar to this:

ITEM NET-ADDR OBJECT: TYPE
1: 56bf.af.fc AnnLW:LaserWriter

2: 56bf.ac.cc TimLW:LaserWriter

ITEM number (0 to make no selection)?

NET-ADDR is the AppleTalk internet address of the printer's listener socket, printed in hex.

OBJECT: TYPE is the name of the registered printer and its type.

at_cho_prn(1)

at_cho_prn(1)

FILES

/usr/bin/at_cho_prn
/usr/lib/PrinterChoices

SER ALSO

atlookup(1), at_print(1), at_status(1); Inside AppleTalk; "Installing and Administering AppleTalk," in AIUX Network System Administration; "AppleTalk Programming Guide," in AIUX Network Applications Programming.

NAME

atlookup — look up NVEs registered in the AppleTalk network

SYNOPSIS

atlookup[-d][-rnn][-snn][-x][object[:type[@zone]]]

DESCRIPTION

The atlookup command uses Name Binding Protocol (NBP) to look up names and addresses of the specified network visible entities (NVEs). The default is to look up all the entities (of all types) in the current zone. Specifying the object, type, and/or zone on the command line limits the scope of lookup. The command line arguments are interpreted as follows:

-d Print network address using decimal numbers.

-rnn Retry lookup nn times. The default is to try the lookup eight times.

-ann Retry lookup every nn seconds. By default, retries are spaced one second apart.

are spaced one second apart.
 Print the 8-bit ascii characters on output as hexade-

Print the 8-bit ascii characters on output as hexadecimal numbers of the form XX (where X is a hex digit).

object Name of the object to be looked up.type Type of the objects to be looked up.

zone Zone in which the lookup is to be performed. You can use an asterisk, *, instead of specific zone name to indicate the current zone name.

The object and type may contain wildcard characters. The symbol

'=' indicates a wildcard lookup. For wildcard lookups to work correctly with all nodes, the wildcard character, if specified, must be the only character in the string. However, nodes implementing Extended AppleTalk Name Binding Protocol also honor a single embedded wildcard character. Under this scheme, one wildcard character may appear anywhere in the string, and will match one or more characters. Note, however, that although an embedded '=' is acceptable in object and type fields for atlookup, only the

nodes implementing Extended AppleTalk protocols will respond

to such a query, hence the list of NVE's thus obtained may be incomplete. Information about the NVEs is displayed in a table format, one line per NVE, containing the *object*, type, and zone names, and the network, node, and socket numbers.

EXAMPLE

This command queries for all NVEs registered on the local AppleTalk zone:

atlookup

queries for all NVEs registered on the local AppleTalk zone. The output appears as follows:

Found 2 entries in zone MyZone 5587.c6.aa doc1:Macintosh II 5586.d4.da doc2:Macintosh II

If this is an Extended AppleTalk network, the command

atlookup d=1:=@*

would yield

Found 1 entries in zone MyZone 5587.c6.aa doc1:Macintosh II

FILES

/usr/bin/atlookup

SEE ALSO

at_cho_prn(1), at_print(1), at_status(1); Inside AppleTalk. atp(3N) atp(3N)

```
NAME
    atp_close, atp_getreq, atp_open, atp_sendreq,
    atp_sendrsp — AppleTalk Transaction Protocol (ATP)
    interface
SYNOPSIS
    #include <at/appletalk.h>
    #include <at/atp.h>
    int atp_open(socket)
    at_socket *socket;
    int atp_close (fd)
    int fd;
    int atp_sendreq(fd, dest, buf, len, userdata, xo, tid,
    resp, retry, comp, param);
    int fd;
    at_inet t *dest;
    char *buf;
    int len, userdata, xo;
    u short *tid;
    at_resp_t *resp;
    at_retry_t *retry;
    int (*comp)(), param;
    int atp_getreq(fd, src, buf, len, userdata, xo, tid,
    bitmap, comp, param);
    int fd;
    at_inet_t *src;
    char *buf;
    int *len, *userdata, *xo;
    u_short *tid;
    u char bitmap;
    int *nresp, (*comp)(), param;
    int atp_sendrsp(fd, dest, xo, tid, resp, comp, param);
    int fd;
    at_inet_t *dest;
    int xo;
    u_short tid;
    at_resp_t *resp;
    int (*comp)(), param;
```

atp(3N) atp(3N)

DESCRIPTION

The ATP interface provides applications with access to the services of the AppleTalk Transaction Protocol, a transport layer protocol.

These routines use the following structures, defined in <at/appletalk.h>.

The at_inet_t structure specifies the AppleTalk internet address of a DDP AppleTalk socket endpoint:

```
typedef struct at_inet {
    u_short net;
    at_node node;
    at_socket socket;
} at_inet_t;
```

The at_retry_t structure specifies the retry interval and maximum count for a transaction:

The members of this structure are:

interval The interval in seconds before ATP retries a request.

retries The maximum number of retries for this ATP request.

If retries is AT_INF_RETRY, the request will be repeated infinitely.

The at_resp_t, defined in <at/atp.h>, specifies buffers to be used for response data:

```
typedef struct at_resp {
    u_char bitmap;
    struct iovec resp[AT_ATP_TRESP_MAX];
    int userdata[AT_ATP_TRESP_MAX];
} at_resp_t;
```

The members of this structure are:

bitmap The bitmap of responses expected, and for which buffers are allocated.

resp An iovec structure describing the response buffers and their lengths.

userdata An array of 32-bit words holding the user bytes for each ATP response.

The atp_open routine opens an ATP AppleTalk socket and returns a file descriptor for use with the remaining ATP calls.

socket

A pointer to the static DDP AppleTalk socket number to open. If the socket number is zero, a socket is dynamically assigned, and the socket number is returned in socket.

The atp_close routine closes the ATP AppleTalk socket identified by the file descriptor fd.

The atp_sendreq routine sends an ATP request to another socket. This call blocks until a response is received.

fd The ATP file descriptor to use in sending the request.

dest The AppleTalk internet address of the AppleTalk

socket to which the request should be sent.

buf Specifies the request data buffer.

len Specifies the size of request data buffer size.

userdata Contains the user bytes for the ATP request header.

xo Should be true (1) if the request is to be an exactly-

once (XO) transaction.

tid On return, contains the transaction identifier for this

transaction; tid can be NULL if the caller is not in-

terested in the transaction identifier.

The atp_sendreq routine requires a pointer to an at_resp_t structure containing two arrays for the response data: resp, an eight-entry iovec array, and userdata, an eight-entry array. The field iov base in each iovec entry points to a buffer to contain response data. The field iov len specifies the length of the buffer. The field bitmap indicates the responses expected; on return, it indicates the responses received.

On return, each iov_len entry indicates the length of the actual response data. If the number of responses is less than expected, either an EOM was received or the retry count was exceeded. In the latter case, an error is returned. Each userdata entry in resp contains the user data for the respective ATP response packet. The retry pointer specifies the ATP request retry timeout in seconds and the maximum retry count. If retry is NULL, the default timeout, AT_ATP_DEF_INTERVAL, and the default retries, AT_ATP_DEF_RETRIES, are used. The retries field of retry can

be set to AT_INF_RETRY, in which case the transaction will be repeated infinitely.

The atp_getreq routine receives an ATP request sent from another AppleTalk socket. It is completed when a request is received.

fd The ATP file descriptor to use in receiving the request.

src The AppleTalk internet address of the AppleTalk socket from which the request was sent.

Specifies the data buffer in which to store the incoming request.

len Specifies the data buffer size in which to store the incoming request.

userdata On return, contains the user bytes from the ATP request header; userdata can be NULL if the caller is not interested in the userdata.

will be true (1) if the request is to be an exactly once (XO) transaction.

tid Contains the transaction identifier for this transaction.

bitmap Indicates the responses expected by the requester.

Because the transaction may require a response, the xo, tid, and

bitmap parameters are always returned.

atp_sendrsp sends an ATP response to another AppleTalk

atp_sendrsp sends an ATP response to another AppleTalk socket. All response data is passed in one at_sendrsp call. In the case of an XO transaction, the call does not return until a release is received from the requester or the release timer expires. In the latter case, an error is returned.

fd The ATP file descriptor to use in sending the response.

dest The AppleTalk internet address of the AppleTalk socket to which the response should be sent.

tid Contains the transaction identifier for this transaction.

The atp_sendrsp routine requires a pointer to an at_resp_t structure containing two arrays for the response data: resp, an eight-entry iovec array, and userdata, an eight-entry array. The field iov_base in each iovec entry points to a buffer containing

atp(3N) atp(3N)

response data. The field *iov_len* specifies the length of the response data. Each userdata entry in resp contains the user data to be sent with the respective ATP response packet. The field bit-map indicates the responses to be sent.

ERRORS

All routines return -1 on error with a detailed error code in error. For additional errors returned by the underlying DDP and AppleTalk Link Access Protocol (ALAP) modules, see ddp(3N) and lap(3N).

[EBADF] fd is not a valid file descriptor (all).

[ENOTTY] fd is not a TTY, that is, not a special device

(all).

[EINTR] The request was interrupted by signal (all).

[EAGAIN] The request failed due to a temporary resource

limitation; try again. When this error occurs,

no XO transaction is initiated (all).

[EINVAL] Invalid dest, len, resp, or retry parameter

(atp_sendreq).

Invalid len parameter (atp getreq).

Invalid dest or resp parameter

(atp_sendrsp).

[ENOENT] An attempt to send a response to a nonexistent

transaction (atp_sendrsp).

[ETIMEDOUT] The request exceeded the maximum retry count

(atp_sendreq).

[EMSGSIZE] The response is larger than the buffer, or more

responses were received than expected. Truncated to available buffer space

(atp sendreq).

The request buffer is too small for request data;

truncated (atp_getreq).

The response is too large; maximum is

AT_ATP_DATA_SIZE

(atp_sendrsp).

WARNINGS

The parameters comp and param allow asynchronous sending and receiving of ATP requests. Currently, asynchronous requests are not supported, so set these parameters to NULL to indicate syn-

atp(3N) atp(3N)

chronous operation.

The length of each response buffer, specified in *lov_len*, is overwritten by the actual response length when atp_sendreq returns.

SEE ALSO

ddp(3N), lap(3N), nbp(3N), pap(3N), rtmp(3N), appletalk(7); Inside AppleTalk; "AppleTalk Programming Guide," in A/UX Network Applications Programming.

NAME

atprint - copy data to a remote PAP server

SYNOPSIS

atprint [-c [-u mid]] [object[:type[@zone]]]

DESCRIPTION

The atprint command opens a PAP (Printer Access Protocol) AppleTalk connection to a remote PAP server, such as a Laser-Writer®, and then copies its standard input to the remote server until it reaches an end-of-file. You can use this command to send "raw" PostScript® code to a LaserWriter. (This command won't engage in a dialog with a LaserWriter about fonts, or load PostScript header files in the same manner as a Macintosh® Operating System does).

The destination PAP server is chosen as follows: If the user has specified a PAP server using object, type and zone fields on the command line, that PAP server is looked up and used. If the user has not specified a choice on the command line, but has used -c option, the command looks for the default print server chosen by the invoking user via at_cho_prn(1). In absence of either, by default, the command looks for the system-wide default print server set via the Macintosh toolbox chooser interface.

You can use the wildcard character '=' for either or both of object and type. In case of wildcard lookup, the command establishes a connection with the first available server and copies the file there. If type is left out, it uses LaserWriter by default. An asterisk, '*', for zone indicates the local zone. Networks without bridges do not have zones and should always use the default, *.

The option -u uid overrides a user's current user ID with the specified uid when choosing a default printer from the file /user/lib/PrinterChoices. This option is valid only in conjunction with the -c option. Typically, this flag option is used only by 1p interface programs.

The atprint command outputs a message indicating which server it is connected to before it transfers data.

EXAMPLE

The command

atprint < test.ps

copies the PostScript file test.ps to the first available Laser-Writer.

WARNING

The atprint command does not interpret contents of input files. To print properly on a PostScript printer, ASCII files must be pre-processed through pstext(1) or enscript(1), and troff-formatted files must be pre-processed through psdit(1).

FII.ES

/usr/bin/atprint
/usr/lib/PrinterChoices

SEE ALSO

at_cho_prn(1), atlookup(1), at_status(1), enscript(1), pstext(1), psdit(1); Inside AppleTalk; Inside Postscript/LaserWriter; "AppleTalk Programming Guide," in A/UX Network Applications Programming; "Installing and Administering AppleTalk," in A/UX Network System Administration.

```
at_status(1)
```

at_status(1)

NAME

at status — return status from a PAP server

SYNOPSIS

at_status object[:type[@zone]]

DESCRIPTION

The at_status command gets the status string from an AppleTalk® PAP (Printer Access Protocol) server such as a Laser-Writer®. These are the at_status command parameters:

object

The name of the PAP server. You must enter this parameter, and wildcards are not permitted. If the name contains spaces, put quotes around the name. Here is an example:

at_status 'Sharons Print Shop:LaserWriter'

type

The type name of the server. Wildcards are not permitted. If you don't enter the type, the system de-

faults to LaserWriter.

zone

The zone of the PAP server. Wildcards are not permitted. If you don't enter the zone, the system defaults to *, your local zone.

FILES

/usr/bin/at_status

SEE ALSO

at_cho_prn(1), atlookup(1), atprint(1), Inside AppleTalk.

•

ddp(3N) ddp(3N)

NAME

ddp_open, ddp_close — AppleTalk Datagram Delivery Protocol (DDP) interface

SYNOPSIS

```
#include <at/appletalk.h>
#include <at/ddp.h>
int ddp_open(socket)
at_socket *socket;
int ddp_close(fd)
int fd;
```

DESCRIPTION

The DDP interface, when included in applications, provides access to the AppleTalk DDP operations.

The ddp_open routine opens a static or dynamic DDP socket and returns an AppleTalk socket file descriptor that can be used to read and write DDP datagrams.

socket

A pointer to the DDP socket number to open. If the AppleTalk socket number is 0, or if socket is NULL, a DDP socket is dynamically assigned. If non-NULL, the socket number is returned in socket.

An error condition results if there are no more dynamic DDP sockets available, if the maximum number of open files has been exceeded at a process or system level, or if the network is offline.

Only the superuser can open a static DDP socket.

The ddp_close routine closes the DDP socket identified by the AppleTalk file descriptor fd.

Datagrams are always read and written with the long DDP header format, using standard A/UX read(2) and write(2) system calls. If the datagram is directed to a LocalTalk™ interface on the same network, the DDP protocol module sends it with a short DDP header. The long header DDP datagram is defined by this structure in <at/ddp.h>:

ddp(3N) ddp(3N)

```
at_net src_net;
at_node dst_node;
at_node src_node;
at_socket dst_socket;
at_socket src_socket;
u_char type;
u_char data[AT_DDP_DATA_SIZE];
} at_ddp_t;
```

When a datagram is written, only the fields checksum, dst_net, dst_node, dst_socket, type, and data need to be set. The length field is the DDP packet length and hop count field. The hop count is in the 6 most significant bits of this field; the length is in the 10 least significant bits.

The checksum field contains the DDP checksum. When datagrams are writen, a checksum is computed only if this field is nonzero.

Datagrams can be sent and received asynchronously using standard A/UX facilities: select(2N); O_NDELAY fcnt1(2); or FIONREAD, FIONBIO, and FIOASYNC ioctls (see ioct1(2)).

ERRORS

All routines return -1 on error with detailed error code in errno:

An fournes fettin -1 on end with deather end code in errifo.	
[EACCES]	A nonsuperuser attempted to open a static AppleTalk socket (ddp_open).
[EADDRINUSE]	The static socket is in use, or all dynamic sockets are in use (ddp_open).
[EINVAL]	An attempt was made to open an invalid AppleTalk socket number (ddp_open).
[EMSGSIZE]	A datagram is too large.
[ENETDOWN]	The network interface is down (all).
[ENOBUFS]	DDP is out of buffers, so unable to return

Routines also return any additional error codes returned by the underlying LAP module (see lap(3N)) and by standard A/UX open(2), close(2), read(2), write(2), and ioctl(2) system calls.

configuration information.

ddp(3N) ddp(3N)

FILES

/dev/appletalk/ddp/*

SEE ALSO

close(2), fcntl(2), ioctl(2), open(2), read(2), select(2N), write(2), atp(3N), ddp(3N), lap(3N), nbp(3N), pap(3N), rmtp(3N), fcntl(5), termio(7), appletalk(7); Inside AppleTalk; "AppleTalk Programming Guide," in A/UX Network Applications Programming.

April, 1989 030-5604-B

• • •

lap(3N) lap(3N)

NAME

lap_default — AppleTalk Link Access Protocol (LLAP/ELAP) interface

SYNOPSIS

char *lap_default()

DESCRIPTION

The lap_default routine returns a character pointer to the LAP interface name of the default interface as defined in /etc/appletalkrc. It returns NULL on error.

ERRORS

If an error occurs, lap_default returns NULL, with a detailed error code in error.

[ENOENT]

No AppleTalk interface exists.

FILES

/dev/appletalk/lap/*/...
/etc/appletalkrc

SEE ALSO

atp(3N), ddp(3N), nbp(3N), pap(3N), rtmp(3N), appletalkrc(4), appletalk(7); Inside AppleTalk; "AppleTalk Programming Guide," in A/UX Network Applications Programming.

April, 1989 030-5604-B

nbp(3N) nbp(3N)

```
NAME
   nbp_parse_entity, nbp_make_entity,
   nbp_confirm, nbp_lookup, nbp_register,
    nbp_remove — AppleTalk Name Binding Protocol (NBP)
   interface.
SYNOPSIS
    #include <at/appletalk.h>
    #include <at/nbp.h>
    int nbp_parse_entity(entity, str);
    at entity t*entity;
    char*str;
    intnbp_make_entity(entity, object, type, zone);
    at_entity_t*entity; char*object, *type, *zone;
    intnbp_confirm(entity, dest, retry);
    at_entity_t*entity;
    at_inet_t*dest
    at_retry_t*retry;
    intnbp_lookup(entity, buf, max, retry);
    at_entity_t*entity;
    at nbptuple t*buf;
    intmax;
    at_retry_t*retry;
    intnbp_register(entity, fd, retry);
    at_entity_t*entity;
    intfd;
    at_retry_t*retry;
    intnbp_remove (entity, fd);
    at_entity_t*entity;
    intfd;
DESCRIPTION
    The NBP interface provides applications with access to the Ap-
    pleTalk Name Binding Protocol operations. These routines use
    these structures defined in <at/appletalk.h>:
                         struct at_inet {
                  u_short
                                      net:
                  at_node
                                      node:
                  at socket
                               socket;
           } at_inet_t;
```

May, 1989

nbp(3N) nbp(3N)

The AppleTalk Name Binding Protocol operations also use these structures defined in <at/nbp.h>:

```
typedef
             struct at_nvestr {
      char
                    len;
                    str[AT_NVE_STR_SIZE];
      char
} at_nvestr_t;
typedef
             struct at_entity {
      at_nvestr_t object;
      at_nvestr_t
                   type;
      at_nvestr_t zone;
} at_entity_t;
typedef struct at_nbptuple {
      at_inet_t
                      enu addr;
      u_char
                    enu_enum;
      at_entity_t enu_entity;
} at nbptuple t;
```

at_inet_t specifies the AppleTalk internet address of a DDP socket endpoint.

at <u>retry_t</u> specifies the retry interval and maximum count for a transaction. The members of this structure are:

interval The interval in seconds before NBP retries a request.

retries The maximum number of retries for this NBP request. If retries is AT_INF_RETRY, then the request will be repeated infinitely.

at_nvestr_t specifies the an NBP entity string. This structure is used to describe the three strings that make up an entity name: object, type, and zone. The members of this structure are:

len The length of the string in bytes.

str The character data for this string.

nbp(3N) nbp(3N)

An entity name is described by the structure at_entity_t, which consists of the above three strings: object, type, and zone.

nbp_parse_entity constructs an NBP entity name from a NULL-terminated "C" string of the form object:type, object:type@zone, type@zone, or just type. The entity name is placed into the at entity t structure entity. This routine returns zero on success.

nbp_make_entity constructs an NBP entity name from object, type, and zone strings. The strings are NULL-terminated "C" strings. The entity name is placed into the at_entity_t structure entity. Use the object, type and zone character strings to construct the entity name. This routine returns zero on success.

nbp_confirm sends a confirmation request to the specified node to see if an entity name is still registered at the specified AppleTalk internet address.

entity A pointer to the at_entity_t structure containing the entity name. No metacharacters are allowed in the entity name strings.

dest The AppleTalk internet address to confirm. If the name is still registered on the node but at a different socket number, the socket number in dest is updated.

The pointer that specifies the NBP request retry timeout in seconds and the maximum retry count. If retry is NULL, the system uses the default values: a one-second timeout and eight retries. Setting the retries field of retry to AT_INF_RETRY causes the transaction to be repeated infinitely.

nbp_lookup returns a list of registered name-address pairs via an NBP lookup.

entity

A pointer to the at_entity_t structure containing the entity name to be looked up.

April, 1989 030-5604-B

buf An array of at_nbptuple_t to receive entity tuples.

max The maximum number of entity tuples to accept. If more than max tuples are received before the lookup retry is exceeded, the lookup terminates. On success, nbp_lookup returns the number of entity tuples actually received.

retry

This pointer specifies the NBP request retry timeout in seconds and the maximum retry count. If retry is NULL, the system uses teh default values: a one-second timeout and eight retries. Setting the retries field of retry to AT_INF_RETRY causes the transaction to be repeated infinitely.

nbp_register adds the specified name-socket pair to the list of registered names on this node.

entity A pointer to the <u>at_entity_t</u> structure containing the entity name to be registered. The zone field of entity is always ignored.

Fd indicates an AppleTalk file descriptor that is registered with the given name. No metacharacters are allowed in the strings.

retry

A pointer that specifies the NBP request retry timeout in seconds and the maximum retry count. If retry is NULL, the system uses the default values: a one-second timeout and eight retries. Setting the retries field of retry to AT_INF_RETRY causes the transaction to be repeated infinitely.

nbp_remove removes the specified entity name from the list of registered names on this node.

entity is a pointer to the at_entity_t structure containing the entity name to be registered. The zone field of entity is always ignored.

Fd indicates an AppleTalk file descriptor that is registered with the given name.

No metacharacters are allowed in the strings.

nbp(3N)nbp(3N)

SEE ALSO

ddp(3n)

DIAGNOSTICS

All routines return -1 on error, with a detailed error code in errno:

[EINVAL]

Invalid entity name (all).

[ETIMEDOUT] Request exceeded maximum retry count (nbp_confirm, nbp_lookup, and nbp_register).

WARNINGS

Strings in entity names and entity tuples are not NULL terminated.

April, 1989 030-5604-B

· . .

NAME

newunix - prepare for new kernel configuration

SYNOPSIS

/etc/newunix [bnet] [nfs] [nonet] [toolbox]
[notoolbox] [localtalk] [nolocaltalk] [ethertalk]
[noethertalk] [tc] [notc] [slip] [noslip] [asttty]
[noasttty]

DESCRIPTION

The newunix command begins the process of configuring a new kernel by installing (or uninstalling) the appropriate scripts and driver-object files needed by autoconfig(1M). To complete the kernel configuration process, run autoconfig after newunix. The appropriate argument to newunix depends on the type of kernel desired:

bnet

basic networking

nfs

Network File System

nonet

non-networking

toolbox

A/UX toolbox

notoolbox

no toolbox capabilities

The arguments specified also depend on the optional peripherals and interfaces desired:

localtalk

LocalTalk support

nolocaltalk

no LocalTalk support

ethertalk

EtherTalk support

noethertalk

no EtherTalk support

asttty

support for serial port expansion board

```
noasttty
     no support for serial port expansion board
tc support for the tape cartridge
notc
     no support for the tape cartridge
     support for the Serial Line Internet Protocol
noslip
     no support for the Serial Line Internet Protocol
```

EXAMPLES

To prepare an NFS kernel, enter this command:

```
/etc/newunix nfs
```

Use this command to add the software that supports the tape controller to the kernel:

```
/etc/newunix tc
```

If after adding the tape controller support, you decide to remove it, enter this command: from the kernel, use

```
/etc/newunix notc
```

In all three of the above examples, after running newunix, run autoconfig to create a new kernel. Then enter this command to reboot your system to begin using the new kernel:

```
autoconfig -I -S /etc/startup
```

See autoconfig(1M) for complete details on running autoconfig.

FILES

```
/etc/boot.d/*
                                  driver object files
/etc/install.d/*
                                  installation scripts
/etc/master.d/*
                                  script files
/etc/startup.d/*
                                  startup programs
/etc/uninstall.d/*
                                  uninstallation scripts
                                  initialization scripts
/etc/init.d/*
```

SEE ALSO

```
autoconfig(1M), finstall(1M);
"Installing and Administering AppleTalk," in A/UX Local System
Administration.
```

```
NAME
    paps_open, paps_get_next_job, paps_status,
    paps_close, pap_open_nve, pap_read,
    pap_read_ignore, pap_status_nve, pap_write,
    pap_close — AppleTalk Printer Access Protocol (PAP)
    interface
SYNOPSIS
    #include <at/appletalk.h>
    #include <at/pap.h>
    #include <at/nbp.h>
    int paps_open()
    int paps_get_next_job(sd)
    int sd;
    int paps_status(sd, status)
    int sd;
    char *status;
    int paps close (sd)
    int sd;
    int pap_open_nve (entity, retry, times, name)
    at_entity_t *entity, *name;
    at_retry_t *retry;
    int times;
    int pap_read(sd, data, len)
    int sd, len;
    char *data;
    int pap_read_ignore(sd)
    int sd;
    char *pap_status_nve (entity, retry)
    at_entity_t *entity;
    at_retry_t *retry;
    int pap_write(sd, data, len, eof, flush)
int sd, len;
int eof, flush;
    char *data;
    int pap_close (sd)
    int sd;
```

pap(3N) pap(3N)

DESCRIPTION

The PAP interface provides applications with access to the AppleTalk Printer Access Protocol operations. The interface routines can be divided in to two sets: one set provides services for a PAP client, the other for a PAP server. These are the routines for the PAP server: paps_open

paps_get_next_job
paps_status

paps_close

These are the the routines for the PAP client:

pap_open_nve
pap_read
pap_read_ignore
pap_status_nve
pap_write

The paps_open routine opens a PAP server AppleTalk socket for a PAP server. The caller may then use nbp_register (see nbp(3N)) to register an NVE on the socket and paps_status to post a status string on it.

The paps_get_next_job routine is called by a server when it is ready to respond to a new PAP client. It returns a PAP client AppleTalk-socket descriptor that is set up for PAP reading from and writing to the client that has been waiting the longest.

sd A PAP-server AppleTalk socket descriptor from a previous paps_open.

Upon successful completion, a PAP client AppleTalk socket descriptor is returned.

The paps_status routine changes the status string associated with an open PAP-server AppleTalk socket. This is the string returned to a PAP-client from an pap_status_nve call.

sd An open PAP-server AppleTalk socket returned from an paps_open call.

A pointer to a null-terminated character string containing the status string being posted. Strings longer than 255 characters are truncated.

Upon successful completion, a value of 0 is returned.

status

pap(3N) pap(3N)

The paps_close routine closes an open PAP server AppleTalk socket.

sd Descriptor for the AppleTalk socket that is to be closed.

It returns 0 upon successful completion.

The at_pap_open_nve routine opens a PAP client AppleTalk socket to a server. It searches first for a server with a registered name that matches that described by the *entity* parameter. The *object* and *type* fields of *entity* may contain wildcards (=). Although some PAP servers may be unavailable, pap_open_nve tries to access all available PAP servers the number of times specified by *times* until it finds one that will accept a client connection.

entity

A pointer to entity struct containing object, type and zone. nbp_make_entity (See nbp(3N)) may be used to construct an entity struct from individual constituent strings.

Pointer to struct containing number of retransmissions and interval between retransmissions for the requests to lookup this NVE. If this argument is NULL (recommended), standard default values will be used for both the parameters (interval = 1 sec; retries = 8). The backoff field of the struct may be left uninitialized.

times The number of times to search for a PAP server. During searching, a list of all available PAP servers is made, and a connection is attempted to each server in turn. This parameter specifies the number of times that all available PAP servers will be polled before the connection gives up. It does not specify the number of printers to be polled. If times is -1, it tries indefinitely.

name If this is non-NULL, the object name, type and zone name of the PAP server connected to it will be returned in the at entity t struct to which it points. If NULL, nothing is returned.

Upon successful completion, this routine returns a PAP client AppleTalk socket connected to the server requested.

April, 1989 030-5604-B

The pap_read routine reads data from a client PAP socket opened by a pap_open_nve or paps_get_next_job call.

sd A PAP client AppleTalk socket descriptor from a previous open.

data The address of the data to be returned. The maximum data length returned is 512 bytes.

length The maximum length to be read.

Upon successful completion, the number of bytes read is returned. A value of 0 is returned when an end-of-file is reached.

The pap_read_ignore routine issues a PAP read request and ignores any returned data. This is used to allow LaserWriters to function when they want to return "status" messages.

sd The streams file descriptor returned by an earlier pap_open_nve.

The pap_status_nve routine locates a PAP server and returns pointer to its status string.

entity A pointer to entity struct containing an object, type and zone. The routine nbp_make_entity (See nbp(3N)) may be used to construct an entity struct from individual constituent strings.

retry

Pointer to struct containing the number of retransmissions and interval between retransmissions for the requests to lookup this NVE. If this argument is NULL (recommended), standard default values will be used for both the parameters (interval = 1 sec; retries = 8). The backoff field of the struct may be left uninitialized.

Upon successful completion, a value of 0 is returned; if the printer's status can not be recovered, -1 is returned.

The pap_write routine sends the data passed to it to the other end of a PAP client session.

A valid PAP client AppleTalk socket descriptor from a call to pap open nve or paps get_next_job.

data A pointer to the data being written.

len The length of the data being written; this must not exceed 512 bytes.

pap(3N) pap(3N)

Boolean flag indicating if EOF indication is to be sent to the other end of PAP session (after the data has been sent) to indicate that no more data will be sent. Setting eof to true implies flush.

Boolean flag indicating if data for all waiting PAP writes is to be sent to the remote end. Because PAP runs on top of ATP, PAP writes are queued up until either a complete ATP response is available (about 4 KB) or an end-of-message is sent. This call sends an ATP end-of-message, which causes all waiting PAP writes to be sent to the other end. This should be done if a higher level protocol (for example, a handshake with a Laser-Writer) needs to do a write followed by a read.

Upon successful completion, a value of 0 is returned.

The pap_close routine closes an open PAP client AppleTalk socket.

sd Descriptor for the AppleTalk socket that is to be closed.

It returns 0 upon successful completion. Otherwise, if the AppleTalk socket is no longer open, -1 is returned.

ERRORS

All routines return -1 on error with a detailed error code in errno:

[EINVAL] An invalid argument was passed.

[ENETDOWN] The network interface is down.

[ESHUTDOWN] The requested AppleTalk socket has already

been closed.

[ETIMEDOUT] The connection is timed out.

See open(2), close(2), read(2), write(2), and ioctl(2) for additional error codes; see also errors returned by the underlying NBP, ATP, DDP, and LAP modules.

SEE ALSO

atp(3N), ddp(3N), lap(3N), nbp(3N), rtmp(3N),

Inside AppleTalk;

"AppleTalk Programming Guide," in A/UX Network Applications Programming.

NAME

rtmp_netinfo — identify AppleTalk node and bridge addresses

SYNOPSIS

```
#include <at/appletalk.h>
int rtmp_netinfo(fd, addr, bridge)
intfd;
at_inet_t *addr, *bridge;
```

DESCRIPTION

addr

This routine allows the caller to determine node addresses. It uses the structure at inet t defined in <at/appletalk.h>:

```
typedef struct at_inet {
    at_net net;
    at_node node;
    at_socket socket;
} at_inet_t;
```

at_inet_t specifies AppleTalk socket internet address. The parameters are used as follows:

fd An AppleTalk socket descriptor. If this parameter is -1, it is ignored, otherwise, socket field in addr upon return would contain the socket number corresponding to fd.

Pointer to a struct at_inet_t. If this pointer is non-NULL, AppleTalk node address would be returned in the struct to which it points. If fd is not -1, socket field of this struct would be filled, otherwise it would be 0. This param is ignored if it is NULL.

bridge Pointer to a struct at_inet_t. If this pointer is non-NULL, AppleTalk address of a bridge known to DDP will be returned in the struct to which it points. It is ignored if NULL. The socket field is meaningless and always contains 0 on return.

Either addr or bridge must be non_NULL. rtmp_netinfo will return -1 if both are NULL.

The function returns 0 if successful, -1 otherwise.

April, 1989 030-5604-B

rtmp(3N)

rtmp(3N)

SEE ALSO
Inside AppleTalk;
"AppleTalk Programming Guide," in A/UX Network Applications
Programming.

April, 1989 030-5604-B

zip(3N) zip(3N)

NAME

zip_getmyzone, zip_getzonelist, zip_locallist
— AppleTalk Zone Information Protocol (ZIP) interface

SYNOPSIS

```
#include <at/appletalk.h>
#include <at/atp.h>
#include <at/nbp.h>
#include <at/zip.h>
int zip_getmyzone(zone)
at_nvestr_t *zone;
int zip_getzonelist(start, buf)
int start;
at_nvestr_t *zone[];
int zip_getlocalzones(start, zones)
int start;
at_nvestr_t *zones[];
```

DESCRIPTION

The ZIP interface provides applications with access to the AppleTalk Zone Information Protocol operations.

The zip_getmyzone routine obtains the zone name for the local network. In case of ALAP, this involves sending a ZIP request to a local bridge to get the zone name of the default network. In case of ELAP, the request is completed on the node itself.

zone

A pointer to the zone name. The zone string is defined by the following structure (see <at/nbp.h>):

```
typedef struct at_nvestr {
          u_char len;
          u_char str[AT_NVE_STR_SIZE];
} at_nvestr_t;
```

len The size of the string in bytes.

str The zone name.

This routine returns 0 upon success.

The zip_getzonelist routine obtains a complete list of all the zone names defined in the internet. This routine sends a ZIP request to a bridge for the list of zone names in the internet. The list is placed into the supplied buffer as concatenated at_nvestr_t structures.

April, 1989 030-5604-B zip(3N) zip(3N)

The starting index for the get zone list request. The start index is the value of the index at which to start including zone names in the response. It is used to obtain a zone list that may not fit into one ATP response packet. The start index should initially be 1. While zip_getzonelist returns a value greter than zero, the caller must reissue zip_getzonelist calls, specifying a start index of the previous start index plus the previous return value of zip_getzonelist.

A buffer to hold this list of zone names. Each zone name is an at_nvestr_t structure. The size of the zone list buffer (in bytes) must be at least AT_ATP_DATA_SIZE. Upon successful completion, this routines returns the number of zone names in the list. When all zones in the bridge's Zone Information Table have been returned, this routine returns 0.

The zip_getlocalzones routine usage and behavior are the same as zip_getzonelist except that it returns the list of zones on the local EtherTalk cable rather than all the zones on the internet.

DIAGNOSTICS

start

All routines return -1 on error with a detailed error code stored in errno:

[EINVAL]

Invalid parameter (all).

[ENETUNREACH]

A bridge node could not be found to process the request (all).

Routines also return any error codes returned by the underlying ATP, DDP, or ALAP layers.

WARNING

The returned zone strings are not null-terminated.

SEE ALSO

ddp(3N), lap(3N), atp(3N),

Inside AppleTalk;

"AppleTalk Programming Guide," in A/UX Network Applications Programming.