A/UX® 2.0 Release Notes

031-0117

APPLE COMPUTER, INC.

Apple Computer, Inc.
20525 Mariani Ave.
Cupertino, California 95014
(408) 996-1010

Simultaneously published in the United States and Canada.

031-0117

LIMITED WARRANTY ON MEDIA AND REPLACEMENT

If you discover physical defects in the manual or in the media on which a software product is distributed, Apple will replace the media or manual at no charge to you provided you return the item to be replaced with proof of purchase to Apple or an authorized Apple dealer during the 90-day period after you purchased the software. In addition, Apple will replace damaged software media and manuals for as long as the software product is included in Apple's Media Exchange Program. While not an upgrade or update method, this program offers additional protection for up to two years or more from the date of your original purchase. See your authorized Apple dealer for program coverage and details. In some countries the replacement period may be different; check with your authorized Apple dealer.

ALL IMPLIED WARRANTIES ON THIS MANUAL, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO NINETY (90) DAYS FROM THE DATE OF THE ORIGINAL RETAIL PURCHASE OF THIS PRODUCT.

Even though Apple has reviewed this manual, **APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS MANUAL IS SOLD "AS IS," AND YOU, THE PURCHASER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS MANUAL, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Tables

Introduction

This document consists of two sections. The first section contains information that was not available in time for the printing of the A/UX® 2.0 manual set. You can add this section to the *A/UX Installation Guide* behind the *Release Notes* tab. In addition, the first section contains a survey that will help us continue to prepare complete, easy-to-use documentation.

The second section contains update pages to the A/UX 2.0 reference manuals.

Late-breaking information

This section presents late-breaking information that was not available in time for the printing of the A/UX 2.0 manual set. Be sure to check the *README* floppy disk for additional timely information that may supplement or supersede the contents of this document.

A correction to the *A/UX Installation Guide*

The *A/UX Installation Guide* incorrectly refers to a System Folder on the *A/UX Startup* floppy disk. This System Folder does not exist.

Using a Macintosh CD-ROM under A/UX Release 2.0

To use a CD-ROM that contains a Macintosh® HFS volume under A/UX Release 2.0, follow the instructions that come with the CD-ROM drive for installing the CD-ROM software in the Macintosh Operating System (OS) System Folder.

▲ **Warning** Do not install any CD-ROM software in the A/UX System Folder. ▲

Next, turn on the CD-ROM drive and start up the A/UX operating system. Insert the CD-ROM disc into the CD-ROM drive and log in as you normally would. The CD-ROM volume will appear on your desktop as it would under the Macintosh OS. If you want to eject the current CD-ROM and use another, drag the CD-ROM icon to the Trash and insert the next CD-ROM. You must log out and log in once again for the second CD-ROM to appear as an icon on your desktop.

- *Note:* To use a CD-ROM that contains a Macintosh HFS volume, you must turn on the CD-ROM drive prior to starting up the A/UX operating system.

Using a UNIX file system CD-ROM under A/UX Release 2.0

If you plan to use a CD-ROM drive to mount UNIX® file systems only, you do not need to install CD-ROM software as described in the previous paragraph. Simply log in as `root` or become the superuser. Insert the CD-ROM disc in the CD-ROM drive and use the `mount` command to mount the file system with read-only access. To mount a different CD-ROM, you must unmount the file system, eject the CD-ROM, insert the new CD-ROM disc, and mount it.

- *Note:* To use a CD-ROM that contains a UNIX file system, you do not have to turn on the CD-ROM drive prior to starting up the A/UX operating system.

Connecting devices with LocalTalk cables and connectors

If you are using LocalTalk® cables and connectors to connect a serial device to a Macintosh computer, make sure that the serial port to which you will connect the device doesn't have a `getty` process running. A `getty` process is left running on a port if, for instance, you disconnect a modem from the port without disabling logins on the port.

Connecting a device with LocalTalk cables and connectors to a port with an active `getty` process can cause invalid data to be transmitted over the LocalTalk network.

To make sure that a `getty` process is not running on the port, use the `setport` command to disable a modem as described in Chapter 7, "Adding and Managing Modems," in *Setting Up Accounts and Peripherals for A/UX.*

Putting away icons on the desktop

To return an icon from the desktop to its original folder, choose Put Away from the File menu. This allows you to put away an icon that, for example, because of access permissions, you otherwise would not be able to drag to its original folder.

The System Folder

A/UX Release 2.0 uses a special version of System 6.0.5. Some files that A/UX uses are not compatible with the Macintosh OS. Conversely, some files that make up the Macintosh OS are not compatible with A/UX Release 2.0.

▲ **Warning** Randomly interchanging files between the Macintosh OS System Folder and the A/UX System Folder is strongly discouraged. ▲

Tables R-1 and R-2 refer to the 6.0.5 Macintosh System Software. Those items marked "A/UX only" are either new for A/UX or have replaced their Macintosh OS counterparts and should not be run under the Macintosh OS. Those items checked "Macintosh OS only" should not be run under A/UX. Those items marked "Either" can be run under A/UX and the Macintosh OS.

■ **Table R-1** Files in the System Folder

Name	A/UX only	Macintosh OS only	Either
+Layers	x	—	—
+MemMgrFix	x	—	—
+NMgrFix	x	—	—
32-Bit QuickDraw	—	—	x
AppleShare	x	—	—
AppleTalk ImageWriter	—	—	x
AUX Resources	x	—	—
Backgrounder	x	—	—
CloseView	—	x	—
Color	—	—	x
DA Handler	—	—	x
Easy Access	—	x	—
Finder	x	—	—
General	—	—	x
ImageWriter	—	—	x
Key Layout	—	—	x
Keyboard	x	—	—
Laser Prep	—	—	x
LaserWriter	—	—	x
LaserWriter IISC	—	x	—
LQ AppleTalk ImageWriter	—	—	x
LQ ImageWriter	—	—	x
MacTCP	x	—	—
Map	—	—	x
Monitors	—	—	x
Mouse	—	x	—
MultiFinder	x	—	—
Patch.0178	x	—	—
Patch.067C	x	—	—

[continued]

■ **Table R-1** Files in the System Folder [continued]

Name	A/UX only	Macintosh OS only	Either
Portable	—	x	—
Print	x	—	—
PrintMonitor	—	—	x
Responder	—	x	—
Sound	—	—	x
StartupDevice	—	x	—
System	x	—	—
Video Card Utility	x	—	—
_DTInit	x	—	—

■ **Table R-2** Macintosh utilities

Name	A/UX Only	Macintosh OS Only	Either
Apple File Exchange	—	—	x
MacroMaker	—	x	—
Font/DA Mover	—	—	x
Installer	—	x	—
HDSC Setup	—	x	—
TeachText	—	—	x
Disk First Aid	—	x	—

The Keycaps Desk Accessory

The Keycaps Desk Accessory displays the Macintosh Plus keyboard regardless of the type of the actual keyboard attached to the system. This is because the Apple Desktop Bus™ Manager is not supported in A/UX 2.0; as a result, the system assumes that a Macintosh Plus keyboard is attached. The displayed keycaps are correct for the extended keyboard; all the keys shown function correctly.

The Mouse CDEV

The Mouse CDEV, which allows you to set the mouse tracking and double-click speed under the Macintosh OS, is not included in this release of the A/UX operating system. Copying the Mouse CDEV from the Macintosh OS into the A/UX System Folder will have no effect.

The Keyboard CDEV

The Keyboard CDEV allows you to select a different key repeat rate and delay until repeat, but such selections have no effect. The Keyboard CDEV is included to support international keyboards.

The MacTCP CDEV

The IP address, net, subnet, and node for your system are displayed for your information. Although the MacTCP® CDEV allows you to modify displayed settings, including the selection of LocalTalk or Ethernet, no change is actually made.

Using the correct MacTCP for A/UX 2.0

A/UX 2.0 uses a special version of MacTCP that is installed in the A/UX System Folder on the standard distribution. If you are using MacX™, be careful that you do *not* copy the version of MacTCP that is included on the *MacX Communications* floppy disk. The MacTCP that is on the *MacX Communications* floppy disk works with the Macintosh OS only, and will cause MacTCP clients, such as MacX, to fail if run under the A/UX operating system.

Switching from EtherTalk to LocalTalk

To switch the AppleTalk® interface from EtherTalk™ to LocalTalk, first make certain that AppleTalk is not being used for an AppleShare® data transfer or for printing. Next, shut down AppleTalk, edit the `/etc/appletalkrc` file, restart AppleTalk and log out. When you log in again the new interface will be active.

- *Note:* Failure to log out and in again will cause unpredictable Chooser results.

The following procedure describes the steps in detail.

1. **To bring down the EtherTalk interface enter**
   ```
   /etc/appletalk -d
   ```
2. **Edit the file `/etc/appletalkrc`, changing the line**
   ```
   interface = ethertalk0
   ```
   to
   ```
   interface = localtalk0
   ```
3. **Make sure the `tty1` line is deactivated. Note that you can skip this step if the `tty1` line is already turned off in the `/etc/inittab` file.**
4. **Edit the `/etc/inittab` file, changing the line**
   ```
   01:2:respawn:/etc/getty tty1 at_9600 #port...
   ```
   to
   ```
   01:2:off:/etc/getty tty1 at_9600 #port...
   ```
5. **To remove the unwanted `getty` process on `tty1` enter**
   ```
   /etc/init q
   ```
   Make sure the LocalTalk cable is securely connected to the printer port on the back of your machine.
6. **To bring up the LocalTalk interface enter**
   ```
   appletalk -u
   ```

Switching from LocalTalk to EtherTalk

If you have an EtherTalk NB card but are currently running LocalTalk, you can manually switch to EtherTalk by following this procedure:

1. **To bring down the EtherTalk interface enter**
   ```
   /etc/appletalk -d
   ```
2. **Edit the file `/etc/appletalkrc`, changing the line**
   ```
   interface = localtalk0
   ```
   to
   ```
   interface = ethertalk0
   ```

3. **Remove the LocalTalk cable from the printer port on the back of your machine.**

▲ **Caution** Make absolutely certain that the LocalTalk cable isn't connected to the printer port when the `getty` process is active. Having the `getty` process active on the printer port with a LocalTalk cable attached can cause invalid data to be transmitted over the network. ▲

If you wish to reconfigure the printer port as a serial terminal port, refer to "Deinstalling AppleTalk" in Chapter 6, "Administering AppleTalk," in *A/UX Network System Administration.*

4. **To bring up the EtherTalk interface enter**
   `/etc/appletalk -u`

The UNIX environment

Using the `mac` include files

Macintosh-specific headers are located in `/usr/include/mac`. If you use, for example, `#include <osutils.h>` in your C program, use the `-I/usr/include/mac` option on the compiler command line. This option will generally be needed because the `mac` headers include other headers that are located in `/usr/include/mac`. If you want to use the include files `<sys/video.h>` and `<sys/uinter.h>`, you must include the following files in the order shown:

```
#include <mac/types.h>
#include <mac/osutils.h>
#include <mac/segload.h>
#include <mac/files.h>
#include <mac/quickdraw.h>
#include <mac/devices.h>
```

```
#include <sys/video.h>
#include <mac/events.h>
#include <sys/types.h>
#include <sys/uinter.h>
```

Running the name server

Commands such as `remsh`, `rcp`, and `rlogin` do case-sensitive matches on names returned from the name server, `named`. Therefore, the contents of the `.rhosts` file in your home directory and the `/etc/hosts.equiv` file must exactly match the contents of the name server's database.

Data encryption

A/UX 2.0 encryption software uses the standard AT&T System V Release 2 DES-like encryption algorithm, as supplied by AT&T.

The Korn shell

If your login shell is the Korn shell, when you log in and open your first CommandShell window, you may see the following message at the top of the window:

```
Warning: no access to tty; thus no job control in this
shell...
```

This message refers to the shell that is used to launch the Macintosh environment, not the shell running in the CommandShell window.

If you are updating from a previous A/UX release

Using file systems created prior to A/UX 2.0

Although A/UX 2.0 comes preconfigured with a BSD root file system, System V file systems, used with previous releases of the A/UX operating system, are still supported. Before you can mount file systems under A/UX 2.0 that were created prior to A/UX 2.0, you must run `fsck` on them. This is so that `fsck` can mark such file systems to conform to a new mechanism for distinguishing between file systems that need to be checked and file systems that do not need to be checked.

This mechanism works as follows. When a file system is mounted, the file system is marked as "in use." When the file system is unmounted properly, it is marked as "clean." If you experience a system crash, the file system will still be marked "in use" when you restart. For file systems that are listed in `/etc/fstab`, the system may automatically invoke `fsck` to check the file system and mark it "clean," before mounting it. See `fsck`(1M) and `fstab`(4) for further details.

- *Note:* You can mount as read-only file systems that are not marked "clean," including pre-A/UX 2.0 file systems. File systems on CD-ROM discs created prior to A/UX 2.0 must be mounted read-only.

Using `dump.bsd` archives created prior to A/UX 2.0

The format of `dump.bsd`(1M) archives has been modified to support the new file system type. Therefore, old archives must be restored using the following options:
`restore -T5.2 -o` *restore-options*

Kernel configuration

The procedure for installing and configuring a new kernel has been simplified. See `newconfig`(1M) for details.

A/UX device drivers

Kernel data structures, such as the `user` structure and the `proc` structure, have changed size and location within the kernel. Therefore, drivers that were developed for releases prior to A/UX 2.0 may not work under A/UX 2.0.

The Slot Library

A/UX Release 2.0 fully supports the Slot Manager. Table R-3 shows the routines that have been removed from the Slot Library because they duplicate Slot Manager functionality.

■ **Table R-3** Routines that have been removed from the Slot Library

Name	Name	Name
`slot_byte`	`slot_structure`	`slot_address`
`slot_data`	`slot_word`	`slot_bytelane`
`slot_long`	`slot_seg_violation`	`slot_calc_pointer`
`slot_resource`	`slot_catch`	`slot_rom_data`
`slot_resource_list`	`slot_ignore`	`slot_header`

See `slots`(3X) for information on the routines that now make up the Slots Library.

Changes in terminology

Some terminology has changed for A/UX Release 2.0. The *stand-alone shell* (`sash`) has been modified and is now called *A/UX Startup.* The ordering, size, and names of partitions have changed. The *SASH Partition* is now called the *MacPartition.* The partition name *Root&Usr* is now called *A/UX Root&Usr slice 0.* The *Swap* partition is now called *A/UX Swap slice 1.*

The `at_nbpd` daemon

The `at_nbpd` daemon, which was part of previous A/UX releases and was described in `appletalk`(7), is obsolete and no longer required. You should remove any `/etc/inittab` entry that references `at_nbpd`.

Customer survey

From: A/UX Group, MS 75-5A
Developer Technical Publications
Apple Computer, Inc.
20400 Stevens Creek Blvd.
Cupertino, CA 95014

To: A/UX customers

Please take a minute to answer this survey, and tell us what you think about our efforts to bring you complete, easy-to-use A/UX documentation. You can mail your comments to the address at the top of this survey, send them via AppleLink to DEV.TECH.PUBS, send them via email to auxpubs@apple.com, or fax comments to us at:

Developer Technical Publications,
A/UX Group,
408-974-1211

Thanks for helping us write better manuals!

Your Name:

Organization:

Address:

Phone Number:

May we call you for follow-up questions? Yes ❒ No ❒

What is your job title and what is your primary job?

Would you classify yourself as a:

❒ User ❒ System Administrator ❒ Programmer ❒ Other_____
(please specify)

How do you use A/UX?

What is your level of expertise?

	novice	**average**	**programmer**	**expert**
Macintosh				
A/UX				
UNIX				

If you are using an older release of A/UX, did you purchase the entire set of books for the last release?

❒ yes ❒ no

How often do you refer to the books?

❒ never
❒ seldom
❒ frequently

Which books do you use most often? For what?

Do you use the on-line man pages, or do you refer to the hard-copy reference manuals, or both?

What books do you use least often?

Could you use additional copies of some of the A/UX books? Which ones? Why?

What product feature(s) would you like to see explained in more detail?

If you have any specific comments on the A/UX books, please fax them to us or send them to one of the addresses noted above.

Reference manual update

This section is of interest to those who have purchased any of the A/UX 2.0 reference manuals. This section contains both new and replacement manual pages for the reference manuals. With the exception of `startmac`(1M), `syslogd`(1M), `read_disk`(8), `waitpid`(2), and `syslog`(3), which are new, the replacement manual pages in this section reflect changes since the initial printing of the reference manuals. In addition, two manual pages, `at_server`(1) and `directory`(3P) should be removed from the manual set. The affected manual pages are listed below under the manual in which they reside.

A/UX Command Reference, Section 1(A-L)

`at_server`(1)	Remove; replace with `at_cho_prn`(1) page 1 and `awk`(1) page 1.
`cc`(1)	Replace with `cb`(1) page 2 through `cc`(1) page 7.
`changesize`(1)	Replace with `cflow`(1) page 3 through `checkcw`(1) page 1.
`cmdo`(1)	Replace with `cmdo`(1) page 1 through `cmp`(1) page 1.
`CommandShell`(1)	Replace with `comm`(1) page 1 through `compact`(1) page 1.
`derez`(1)	Replace with `delta`(1) page 4 through `deroff`(1) page 1.
`enscript`(1)	Replace with `enable`(1) page 1 through `enscript`(1) page 3.
`f77`(1)	Replace with `expr`(1) page 3 through `f77`(1) page 3.
`fcnvt`(1)	Replace with `fcnvt`(1) page 1 through `fgrep`(1) page 1.

A/UX Command Reference, Sections 1(M-Z) and 6

`rez`(1)	Replace with `rev`(1) page 1 through `rez`(1) page 3.
`rlog`(1)	Replace with `rlog`(1) page 1 through `rlogin`(1) page 1.
`sysline`(1)	Replace with `sysline`(1) page 1 through `systemfolder`(1) page 1.
`whatis`(1)	Replace with `whatis`(1) page 1 and `whereis`(1) page 1.

A/UX Programmer's Reference, Sections 2 and 3(A-L)

`waitpid`(2)	New; replacements for `wait3`(2N) page 1 and `write`(2) page 1 are provided.
`directory`(3)	Replace with `dim`(3F) page 1 through `dprod`(3F) page 1.
`directory`(3P)	Remove; deleted (see previous step).

A/UX Programmer's Reference, Sections 3(M-Z), 4, and 5

`slots`(3X)	Replace with `sleep`(3C) page 1 through `slots`(3X) page 3.
`syslog`(3)	New; replacement pages for `sysconf`(3P) page 2 through `system`(3F) page 1 are provided.

A/UX System Administrator's Reference

`adduser`(1M)	Replace with `acctwtmp`(1M) page 1 through `adduser`(1M) page 3.
`ff`(1M) and `finc`(1M)	Replace with `exterr`(1M) page 1 through `finc`(1M) page 1.
`frec`(1M) and `fsck`(1M)	Replace with `frec`(1M) page 1 through `fsdb`(1M) page 1.
`ftpd`(1M)	Replace with `fsstat`(1M) page 1 through `ftpd`(1M) page 3.
`kconfig`(1M) and `keyset`(1M)	Replace with `kconfig`(1M) page 1 through `keyset`(1M) page 1.
`newfs`(1M)	Replace with `newconfig`(1M) page 3 through `newunix`(1M) page 1.
`sadc`(1M)	Replace with `sa2`(1M) page 1 through `sadc`(1M) page 3.
`shutdown`(1M)	Replace with `shutdown`(1M) through `slattach`(1M) page 1.
`startmac`(1M)	New; replacements for `sprayd`(1M) page 1 through `StartMonitor`(1M) page 1 are provided.
`syslogd`(1M)	New; insert after `sysinitrc`(1M) page 1.
`tzic`(1M)	Replace with `tzic`(1M) page 1 through `umount`(1M) page 1.
`appletalk`(7)	Replace with `intro`(7) page 3 and `appletalk`(7) page 1.
`read_disk`(8)	New; insert after `launch`(8) page 3.

NAME

at_cho_prn — choose a default printer on the AppleTalk® internet

SYNOPSIS

at_cho_prn [*type*[*@zone*]]

DESCRIPTION

The at_cho_prn command displays a list of printer selections and saves the name of the printer that you select. 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 the zone in which you'd like to select your default printer.

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. The system prompts you to select a printer by entering the appropriate number from the printer list display.

EXAMPLES

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 hexadecimal.

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

FILES

/usr/bin/at_cho_prn

SEE ALSO

atlookup(1), atprint(1), atstatus(1); *Inside AppleTalk;* "Installing and Administering AppleTalk," in *A/UX Network System Administration;* "AppleTalk Programming Guide," in *A/UX Network Applications Programming*.

NAME

awk — pattern scanning and processing language

SYNOPSIS

awk [-f *file...*] [-Fc] [*prog*] [*parameters*] [*file...*]

DESCRIPTION

awk scans each input *file* for lines that match any of a set of patterns specified in *prog*. With each pattern in *prog* there can be an associated action that will be performed when a line of a *file* matches the pattern. The set of patterns may appear literally as *prog*, or in a file specified as -f *file*. The *prog* string should be enclosed in single quotes (') to protect it from the shell.

parameters, in the form x=... y=... etc., may be passed to awk.

Files are read in order; if there are no files, the standard input is read. The filename - means the standard input. Each line is matched against the pattern portion of every pattern-action statement; the associated action is performed for each matched pattern.

An input line is made up of fields separated by white space. (This default can be changed by using FS; see below). The fields are denoted $1, $2, ...; the variable $0 refers to the entire line.

A pattern-action statement has the form:

pattern { *action* }

A missing *action* means print the line; a missing *pattern* always matches. An *action* is a sequence of statements. A statement can be one of the following:

```
if ( conditional ) statement [ else statement ]
while ( conditional ) statement
for ( expression ; conditional ; expression ) statement
break
continue
{ [ statement ] ... }
variable = expression
print [ expression-list ] [ >expression ]
printf format [ , expression-list ] [ >expression ]
next
exit
```

FILES

/usr/bin/cb

SEE ALSO

cc(1), indent(1).
The C Programming Language by B. W. Kernighan and D. M. Ritchie (Prentice-Hall, Inc., New Jersey, 1978).

BUGS

Punctuation that is hidden in preprocessor statements will cause indentation errors.

NAME

cc — C compiler

SYNOPSIS

cc [-A *factor*] [-B *string*] [-c] [-C] [-D*symbol* [=*def*]] [-E] [-fm68881] [-F] [-g] [-I*dir*] [-l*x*] [-L *dir*] [-n] [-o *outfile*] [-O] [-p] [-P] [-R] [-s] [-S] [-t [p012al]] [-T] [-U*symbol*] [-v] [-W *c*,*arg1*[,*arg2*...]] [-X] [-Z*flags*] [-#] ... *file* ...

DESCRIPTION

cc is a front-end program that invokes the preprocessor, compiler, assembler, and link editor, as appropriate. The default is to invoke each one in turn.

Arguments whose names end with .c are taken to be C source programs. They are compiled, and each object program is left in a file in the current directory, whose name is that of the source, with .o substituted for .c. In the same way, arguments whose names end with .s are taken to be assembly source programs and are assembled to produce a .o file. By default, the named files are loaded to produce an output file named a.out. If a single C program is compiled and loaded all at once, the .o file is deleted.

FLAG OPTIONS

The following flag options are interpreted by cc. (Other flag options may be passed to the assembler and the linker. See ld(1) for link-editor flag options and as(1) for assembler options.)

-A *factor*

Expand the default symbol table allocations for the compiler, assembler, and link editor. The default allocation is multiplied by the *factor* given.

-B *string*

Construct pathnames for substitute preprocessor, compiler, assembler, and link-editor passes by concatenating *string* with the suffixes cpp, comp, optim, as, and ld. If *string* is empty, it is taken to be /lib/. For example, versions of the C compiler, assembler, and link editor can be found in the directory /usr/lib/big. These tools operate just like their standard counterparts, except that their symbol tables are very large. If you receive an overflow error message when you compile your program with the standard versions, you may wish to switch to the alternate versions using

`cc -B /usr/lib/big/ -o` *filename filename*`.c`

You should have 4 MB or more of main memory in order to use the big versions of these programs safely.

`-c` Suppress the link-editing phase of the compilation and force an object file to be produced even if only one program is compiled.

`-C` Pass along all comments except those found on `cpp`(1) directive lines. The default strips out all comments.

`-D`*symbol*[=*def*]
Define the external *symbol* to the preprocessor and give it the value *def* (if specified). If no *def* is given, *symbol* is defined as 1. This mechanism is useful for conditional compilation using preprocessor control lines.

`-E` Run only `cpp`(1) on the named C programs and send the result to the standard output.

`-fm68881`
Generate inline code for the MC68881 floating-point coprocessor. This is the default.

`-F` Do not generate inline code for the MC68881 floating-point coprocessor.

`-g` Generate additional information needed for the use of `sdb`(1).

`-I`*dir* Search for `#include` files (whose names do not begin with `/`) in *dir* before looking in the directories on the standard list. Thus, `#include` files whose names are enclosed in `" "` (double quotes) are initially searched for in the directory of the `.c` file currently being compiled, then in directories named in `-I` flag options, and finally in directories on a standard list. For `#include` files whose names are enclosed in <>, the directory of the `.c` file is not searched.

`-l`*x* Same as `-l` in `ld`(1). Search the library `libx.a`, where *x* is up to 7 characters long. A library is searched when its name is encountered, so the placement of `-l` is significant. By default, libraries are located in `LIBDIR`. If you plan to use the `-L` option, that option *must* precede `-l` on the command line.

`-L` *dir* Same as `-L` in `ld`(1). Change the algorithm of searching for `lib`*x*`.a` to look in *dir* before looking in `LIBDIR`. This option is effective only if it precedes the `-l` option on the command line.

`-n` Arrange for the loader to produce an executable which is linked in such a manner that the text can be made read-only and shared (nonvirtual) or paged (virtual).

`-o` *outfile*
Same as `-o` in `ld`(1). Produce an output object file, *outfile*. The default name of the object file is `a.out`.

`-O` Invoke an object-code optimizer. The optimizer moves, merges, and deletes code, so symbolic debugging with line numbers could be confusing when the optimizer is used. For this reason, use of the `-g` flag option disables the `-O` flag option. This option may not work properly on code containing `asm` directives.

`-p` Arrange for the compiler to produce code that counts the number of times each routine is called. Also, if link-editing takes place, replace the standard startoff routine by one that automatically calls `monitor`(3C) at the start and arranges to write out a `mon.out` file at normal termination of execution of the object program.

`-P` Run only `cpp`(1) on the named C programs and leave the result on corresponding files suffixed `.i`.

`-R` Have assembler remove its input file when finished.

`-s` Same as `-s` in `ld`(1). Strip line-number entries and symbol-table information from the output of the object file.

`-S` Compile, but do not assemble, the named C programs and leave the assembly-language output on corresponding files suffixed `.s`.

`-t [p012al]`
Find only the designated preprocessor passes whose names are constructed with the *string* argument of the `-B` flag option, that is, (`p`), compiler (`0` and `1`), optimizer (`2`), assembler (`a`), and link editor (`l`). In the absence of a `-B` option and its argument, *string* is taken to be `/lib/n`. Using `-t` with no argument is equivalent to `-tp012`.

`-T` Truncate symbol names to 8 significant characters. Many modern C compilers, as well as the proposed ANSI standard for C, allow arbitrary-length variable names. `cc` follows this convention. The `-T` option is provided for compatibility with earlier systems.

`-U`*symbol*
Undefine *symbol* to the preprocessor.

`-v` Print the command line for each subprocess executed.

`-W`*c*`,`*arg1*`[,`*arg2*`...]`
Hand off the argument(s) *argi* (where *i* = 1,2, ..., *n*) to pass *c*, where *c* is one of `[p012al]` indicating preprocessor, compiler first pass, compiler second pass, optimizer, assembler, or link editor, respectively. For example,

```
-Wa,-m
```

invokes the `m4` macro preprocessor on the input to the assembler. (The `-m` flag option to `as` causes it to go through `m4`.) This must be done for a source file that contains assembler escapes.

`-X` Ignored by A/UX® for the Motorola 68020 and 68030 host processors.

`-Z`*flags* Special *flags* to override the default behavior (see NOTES in this section). Currently recognized flags are:

- `c` Suppress returning pointers in both a0 and d0.
- `n` Emit no code for stack-growth. This is the default.
- `m` Use Motorola SGS-compatible stack growth code.
- `p` Use `tst.b` stack probes.
- `E` Ignore all environment variables.
- `I` Emit inline code for the MC68881 floating-point coprocessor. This is the default.
- `l` Suppress selection of a loader command file.
- `t` Do not delete temporary files.
- `S` Compile to be SVID-compatible. Link the program with a library module that calls `setcompat`(2) with the `COMPAT_SVID` flag set. Define only the `SYSV_SOURCE` feature test mac-

ro.

P Compile for the POSIX environment. Link the program with a library module that calls setcompat(2) with the COMPAT_POSIX flag set. Define only the POSIX_SOURCE feature test macro.

B Compile to be BSD-compatible. Link the program with a library module that calls setcompat(2) with the COMPAT_BSD flag set. Define only the BSD_SOURCE feature test macro.

-# Without actually starting the program, echo the names and arguments of subprocesses that would have started. This is a special debug option.

Other arguments are taken to be link-editor flag-option arguments, C-compatible object programs (typically produced by an earlier run of cc), or libraries of C-compatible routines. These programs, together with the results of any compilations specified, are link edited (in the order given) to produce an executable program with the name a.out unless the -o flag option of the link editor is used.

FILES

/usr/bin/cc	
file.c	input file
file.o	object file
file.s	assembly language file
a.out	link-edited output
/usr/tmp/mc68?	temporary
/lib/cpp	preprocessor
/lib/comp	compiler
/lib/optim	optimizer
/bin/as	assembler, as(1)
/bin/ld	link editor, ld(1)
/lib/libc.a	standard library, see (3)
/lib/libposix.a	POSIX library, see (2P) and (3P)
/lib/libbsd.a	BSD library
/lib/libsvid.a	SVID library
/usr/lib/shared.ld	loader command file for shared text or paged programs

`/usr/lib/shlib.ld`	loader command file for shared text or paged programs using shared libraries
`/usr/lib/unshared.ld`	loader command file for unshared text programs
`/usr/lib/unshlib.ld`	loader command file for unshared text programs using shared libraries
`/lib/crt0.o`	run-time startoff
`/lib/crt1.o`	run-time startoff with shared library support
`/lib/crt2.o`	run-time startoff used with `crt1.o` for shared library support
`/lib/crtn.o`	run-time startoff used with `crt1.o` and `crt2.o` for shared library support
`/lib/mcrt0.o`	run-time startoff for profiling

SEE ALSO

`as`(1), `dis`(1), `ld`(1), `setcompat`(2).
The C Programming Language by B. W. Kernighan and D. M. Ritchie, (New Jersey, Prentice-Hall: 1978); "`cc` Command Syntax" in *A/UX Programming Languages and Tools, Volume 1*; "A/UX POSIX Environment," in *A/UX Programming Languages and Tools, Volume 1*.

DIAGNOSTICS

The diagnostics produced by the C compiler are sometimes cryptic. Occasional messages may be produced by the assembler or link editor.

WARNINGS

By default, the return value from a C program is completely random. The only two guaranteed ways to return a specific value are to call `exit` explicitly (see `exit`(2)) or to leave the function `main()` with a `return` (*expression*) statement.

NOTES

This version of `cc` is based on the `cc` released with the Motorola SGS and has been changed in the following ways:

- The `-Z` flag option has been added to explicitly control generation of stack-growth code for cross-development environments or generation of stand-alone code. The Motorola SGS looks for an environment variable called `M68000` and generates stack-growth code if the variable is set to `STACKCHECK`. This `cc` defaults to no stack-growth code on the Macintosh II® 68020 and 68030 processors.
- The default is to produce shared text programs. To produce nonshared text programs, you must run `ld` with the `-N` flag option.
- When `cc` is used with the `-g` flag option, the arguments `-u _dbargs -lg` are inserted in the command line for the link phase. This causes the contents of `libg.a` to be linked in. Note that the Motorola SGS only generates the loader argument `-lg`, which is not sufficient to cause loading of the library's contents.
- The `-v` (verbose) flag option has been added to print the command line for each subprocess executed. This helps to isolate problems to a specific phase of the compilation process by showing exactly what `cc` is doing, so that each phase can be run by hand, if necessary.
- The Motorola SGS compiler expects functions that return pointers or structures to return their values in a0 and expects other functions to return their values in d0/d1. Because of the large body of existing code that has inconsistent type declarations. This version of the compiler emits code to return pointers in both a0 and d0 by copying a0 to d0 just prior to returning. This copy operation can be suppressed with the `-Zc` flag option, thus generating slightly smaller code.

FILES

/usr/bin/cflow	
/usr/lib/lpfx	filters line(1) output into dag input
/usr/lib/nmf	converts nm output into dag input
/usr/lib/dag	graph maker
/usr/lib/flip	reverser

SEE ALSO

as(1), cc(1), cpp(1), lex(1), lint(1), nm(1), pr(1), yacc(1).

BUGS

Files produced by lex(1) and yacc(1) cause the reordering of line number declarations which can confuse cflow. To get proper results, feed cflow the yacc or lex input.

NAME

changesize — change the fields of the SIZE resource of a file

SYNOPSIS

/mac/bin/changesize [-v] [-p*prefsize*] [-m*minsize*] [±*option*] *file*

DESCRIPTION

changesize is based on an MPW tool that prints the fields of the SIZE resource of an application and allows the user to modify any of the fields of the SIZE resource. The format of the SIZE resource contains MultiFinder flags followed by the preferred size and minimum size of the application.

FLAG OPTIONS

The following flag options are interpreted by changesize:

-v Print the values of fields in the SIZE resource and then exit without changing anything.

-p*prefsize*
Specify an amount of memory in which the application will run effectively and which MultiFinder attempts to secure upon launch of the application. This value is expressed in units of kilobytes (K).

±*option*
Set or clear the MultiFinder flag specified by *option*. +*option* sets the MultiFinder flag, -*option* clears the flag. Multiple options can be specified at the same time on the command line. The MultiFinder flags that can be modified are:

SaveScreen
For SWITCHER compatibility. Normally, this is set to 0.

SuspendResume
When set, this bit signifies that the application knows how to process suspend/resume events.

OptionSwitch
For SWITCHER compatibility. Normally, this is set to 1.

CanBackground
Receive null events while in the background, if set.

MultiFinderAware
Take responsibility for activating and deactivation

any windows in response to a suspend/resume event, if set.

`OnlyBackground`
Set this flag if your application does not have a user interface and will not run in the foreground.

`GetFrontClicks`
Set this flag is you want to receive the mouse-down and mouse-up events used to bring your application to the foreground when the user clicks in one of the windows of your application while it is suspended.

`ChildDiedEvents`
Normally, applications set this to 0. Debuggers may set this flag to 1.

`32BitCompatible`
Set this flag if your application is 32-bit clean.

EXAMPLES

To print the fields of the size resource, use

```
/mac/bin/changesize -v file
```

To set the `32BitCompatible` flag, clear the `CanBackground` flag, set the preferred memory size to 500 KB, using

```
/mac/bin/changesize +32BitCompatible
      -CanBackground -p500 file
```

NOTES

This command is not supported in 24-bit mode and must be run from the command line while logged into the Macintosh environment with the 32-bit session type.

checkcw(1)

See cw(1)

NAME

cmdo — build commands interactively

SYNOPSIS

cmdo [*command-name*]

DESCRIPTION

cmdo helps you build A/UX commands using specialized Macintosh dialog boxes. The dialog boxes make it easy to select options, choose files, and access help information, as well as build compound command lines.

Commands with many options and parameters may employ one or more nested dialog boxes.

cmdo may be run in one of three ways:

1. Double-click a command's icon from the Finder.
2. Enter a complete cmdo command line:

   cmdo *command-name*

3. Type a partial command line containing only the name of the command within a CommandShell window, then select Commando from the Edit menu. The Command-key equivalent for selecting Commando from the menu is COMMAND-K.

The first two methods of invoking cmdo execute the command. The third method pastes the command line arguments onto the command line so that more commands, flag options, and arguments can be added to create the desired command line. A command line can even be created with cmdo consisting of many commands piped together, also known as a compound command line. Pressing the RETURN key executes the command.

EXAMPLES

```
cmdo ls
```

displays the ls Commando dialog box.

FILES

/etc/cmdo

SEE ALSO

CommandShell(1).

NOTES

This command is not supported in 24-bit mode and must be run from the command line while logged into the Macintosh environment with the 32-bit session type.

NAME

cmp — compare two files

SYNOPSIS

cmp [-l] [-s] *file1 file2*

DESCRIPTION

The two files are compared. (If *file1* is -, the standard input is used.) Under default flag options, cmp makes no comment if the files are the same; if they differ, it announces the byte and line number at which the difference occurred. If one file is an subset of the other, that fact is noted.

FLAG OPTIONS

The following flag options are interpreted by cmp:

-l Print the byte number (decimal) and the differing bytes (octal) for each difference.

-s Print nothing for differing files; return codes only.

EXAMPLES

```
cmp alpha beta
```

will report if the files are different and at what point they differ, such as:

```
alpha beta differ: char 33, line 2
```

FILES

/bin/cmp

SEE ALSO

bdiff(1), comm(1), diff(1), diff3(1), diffmk(1).

DIAGNOSTICS

Exit code 0 is returned for identical files, 1 for different files, and 2 for an inaccessible or missing argument.

NAME

comm — select or reject lines common to two sorted files

SYNOPSIS

comm [-[123]] *file1 file2*

DESCRIPTION

comm reads *file1* and *file2* which should be ordered in ASCII collating sequence (see sort(1)), and produces a three-column output: lines only in *file1* lines only in *file2* and lines in both files. The file name - means the standard input.

Flags 1, 2, or 3 suppress printing of the corresponding column. Thus comm -12 prints only the lines common to the two files; comm -23 prints only lines in the first file but not in the second; comm -123 is a no-op.

EXAMPLES

```
comm -12 filea fileb
```

prints only the lines common to filea and fileb.

```
comm -23 filea fileb
```

prints only lines in the first file but not in the second.

```
comm -123 filea fileb
```

is not an option, as it suppresses all output.

```
comm -3 filea fileb
```

prints only the lines that differ in the two files.

FILES

/usr/bin/comm

SEE ALSO

bdiff(1), cmp(1), diff(1), diff3(1), diffmk(1), sort(1), uniq(1).

NAME

CommandShell — A/UX® Toolbox application for managing command-interpretation windows and moderating access to the A/UX console window

SYNOPSIS

CommandShell [-b *pid*] [-u] [-q]

DESCRIPTION

CommandShell provides a Macintosh user interface to A/UX users. Within CommandShell windows, you can enter A/UX command lines for processing by one of the available shells.

A/UX commands can be entered with the aid of Macintosh copy and paste operations. The source of text that can be copied to the Clipboard includes any previous command lines in the same window as well as any text available in other windows (including other Macintosh application windows). The CommandShell windows are scrollable, so you can make previous commands or resultant outputs available for copying.

A/UX commands can also be built semiautomatically by entering the command name at the start of a line and then choosing the Commando menu item from the Edit menu of CommandShell. After entering information in the dialog box, the command line is changed to include all the flag options and arguments that were generated with the help of the Commando dialog. This same function is also available through the cmdo command, which is described in cmdo(1).

When CommandShell is started normally, one window is displayed by default unless you have saved a previous window layout with more than one window. More windows can be created by choosing New from the File menu or pressing COMMAND-N. A new window appears in front of the existing window or windows. The title bar of each window is numbered in sequence according to its creation order. Normally you can create up to 15 windows.

When you create a new window, it appears in front of and to the lower right of the previous window. This position obscures the windows behind the front window. You can use the tiling commands to view the contents of all the windows. For specific tiling commands, see the section describing the Window menu.

Besides managing command-interpretation windows, CommandShell moderates access to the A/UX console window. This window is one of the places where the A/UX environment and the Macintosh desktop environment meet. Kernel error messages are routed to this window so they do not disturb the bit-mapped display of Macintosh applications.

The Macintosh user interface is an integral part of the A/UX boot process, supported in part by the A/UX console window of CommandShell. This special window is the place where all boot messages are routed and the place where you enter responses in the event that one of the boot processes requires user input. The A/UX console window is unlike any other CommandShell window because it can never be closed (although it can be hidden or obscured). During the boot process, CommandShell disables many of the functions in its menus (see the -b flag option).

All the status messages that are normally directed to the system console during startup no longer appear. Many of these messages are not useful to anyone besides the system administrator. A progress bar is presented to users in place of the startup messages. When multiuser mode is entered, the progress bar is replaced by a login dialog box.

If you wish to inspect the boot messages that resulted when the system was last booted, you can choose the A/UX System Console window from the Window menu of CommandShell. Within multiuser mode, a CommandShell process has to be started manually through a command request or automatically through the Login preferences that you maintain at the time you log in to the system (see Login(1M)).

To view the A/UX console window, perform these steps:

1. Choose CommandShell from the Apple menu.
2. Choose A/UX System Console from the Window menu.
3. Use the scroll bar to scroll back the contents of the window.

Processes that run as part of a startup script, such as /etc/sysinitrc, or as another part of the booting process may occasionally ask for user input. For example, suppose you add another Ethernet card to your system. Then suppose that while rebooting, the system needs to request address information about the new card. You will not see a prompt for this information directly. Instead an alert box will appear in front of the A/UX

boot progress bar. The alert box will say that an A/UX process requires input. Clicking OK in the alert box will cause the the A/UX console window to appear and the alert box to disappear. The A/UX console window will contain messages prompting you for input. At these junctures, you will be permitted to provide input into the window. At all other times, the window is for reading purposes only.

The alerting process that has just been described is referred to as a *notification system*. A similar notification system has been created for the handling of A/UX kernel messages. Alert boxes are displayed that encourage you to inspect the text of the A/UX error messages that has been tucked away in the normally-hidden A/UX console window.

Here is what you should do in response to this form of notification:

1. After reading the alert box, click OK.
2. Make `CommandShell` the selected application if it is not already the application in control of the active window.
3. Inspect the A/UX kernel errors by choosing A/UX System Console from the Window menu of `CommandShell`. When chosen, this menu item makes active the normally hidden A/UX console window. If it was hidden, it will now be visible. It should contain all the error messages generated since the last time the system was booted (if you scroll back, you can see the old messages). It will also let you respond to any prompts for input.

If after checking the window messages in the A/UX console window, you see that the error information regards system or network performance (such as a number of retries before successful transmission of a network packet), then no further corrective action is required. Sometimes, however, the error message may indicate a serious error condition, such as "`file system full`," "`file system corrupt`," or "`fork failed: too many processes.`" In these cases your current work is subject to loss if the error condition persists.

You can change the way the system notifies you of a process needing input or a system message when `CommandShell` is not the active application. To set notification preferences, choose Notification Levels from the Preferences menu. Choosing

Notification Levels displays a dialog box in which you can select to be notified by an alert box, a flashing icon in the menu bar, or both. User preferences are normally stored in the `.cmdshellprefs` files located in the home directories of each user (see "*Managing CommandShell preferences*"). The default notification is to display an alert box because console messages may indicate a fundamental system problem that you should know about immediately.

When `CommandShell` is run in the normal way and is the active application, a Quit menu item is not available under the File menu (see the `-q` flag option for an alternative way of running `CommandShell`). Instead of a Quit function, which would leave you with no way to view or enter responses directed to the system console, `CommandShell` has the menu item Close All Windows. This menu item kills all shell processes and closes all of the windows, except the A/UX console window that accepts console messages. If you simply wish to set aside `CommandShell` windows without killing active processes, choose Hide Command Shell Windows from the Apple menu.

The combined A/UX and desktop environment supported with `CommandShell` is unique in terms of the flexibility you have for performing file manipulation chores. To delete a file, you can drag it in its iconic form on the desktop to the Trash icon and choose Empty Trash from the Special menu. Alternately, you can open a `CommandShell` window to run the A/UX command `rm`. The latter way may be the quicker way to perform a sophisticated operation. For example, consider the task of removing all files in the current folder (or directory) containing the letters "`.tmp`." The easy way is to use a command line such as

```
rm *.tmp*
```

Since A/UX is a multitasking system, other windows besides the active one receive processor attention. But when you use a command, only the active window (the window with horizontal lines in the title bar) receives the instruction. All mouse and keyboard inputs are reliably intercepted by the active window. However, if the active application has no open windows, character sequences may be lost.

The terminal emulation capabilities of CommandShell are most closely analogous to those of `mac2cs` in `/etc/termcap`.

MENU OPTIONS

CommandShell displays menus titled File, Edit, Window, Fonts, Commands, and Preferences in the menu bar at the top of the screen, plus the Apple menu at the far left of the menu bar. To choose a menu item, position the pointer on the menu title, press and hold the mouse button, and move the pointer to the menu item while holding down the mouse button. Release the mouse button when the pointer highlights the desired item.

Many menu items can be chosen from the keyboard by holding down COMMAND (not CONTROL) and typing a character. The character required is shown beside the Command-key symbol in the menu. Such Command-key equivalents may be entered as lowercase; you don't need to hold down SHIFT as well.

The following sections describe the actions performed by the various CommandShell menu items.

The Apple Menu

At the far left of the menu bar, the Apple symbol is the title of a menu that contains some general Macintosh desk accessories and some menu items specifically related to CommandShell. The menu items related to CommandShell include:

About CommandShell
: Display a dialog box that gives version information.

Hide Command Shell Windows
: Hide CommandShell windows temporarily without killing active processes.

CommandShell
: Make CommandShell the selected application and make its associated window that was most recently active the currently active window once again (if any were opened before).

The File Menu

The menu items in the File menu allow you to do such things as create and close windows and select printing options.

New
: Create a new window. The windows are numbered sequentially according to their creation order. The Command-key equivalent for the New menu item is COMMAND-N.

Open...
: Launch a UNIX command or launch an editor if the highlighted file is a text file. The

Command-key equivalent for Open menu item is COMMAND-O.

Close — Close the active window. Before you close a window, make sure that you write the contents of the window to a disk if you want to save your work. The Command-key equivalent for the Close menu item is COMMAND-W.

Save Selection — Save the contents of a `CommandShell` window in an A/UX file. The text you want saved must be selected (the text appears highlighted).

Save Preferences — Save all window settings, layout information, and notification level settings.

Restore From Preferences — Restore window settings and layout to that specified in the preferences file. Open any saved windows that have been closed. Run initial command in windows that do not already have a command running.

Page Setup — Display a dialog box that lets you set the paper size, orientation, and reduction or enlargement for subsequent printing actions.

Print Selection — Print selected text from the active window. Use the Chooser desk accessory, available in the Apple menu, to specify which printer to use. Use the Page Setup menu item, just described, to specify paper size, orientation, and scale.

Close All Windows — Close all windows at once. All the windows disappear. Before you close the windows, make sure that you write the contents of each window to a disk if you want to save your work. If you don't write the contents to a disk, they are lost.

The Edit Menu

The menu items in the Edit menu help you do such things as move text around and perform certain global formatting actions.

Undo — Reverse the most recent text change. If you choose Undo a second time, the change is reinstated. The Command-key equivalent for the Undo menu item

is COMMAND-Z.

Cut — Copy the currently selected text in the active window to the Clipboard and then delete it from the window. This menu item is used with desk accessories only; otherwise it is disabled. The Command-key equivalent for the Cut menu item is COMMAND-X.

Copy — Copy the currently selected text in the active window to the Clipboard without deleting it from the window. The Command-key equivalent for the Copy menu item is COMMAND-C.

Paste — Inserts the contents of the Clipboard at the current text cursor location. The Command-key equivalent for the Paste menu item is COMMAND-V.

Clear — Delete the currently selected text from the active window. This menu item is used with desk accessories only; otherwise it is disabled. The Command-key equivalent for the Clear menu item is DELETE.

Select All — Select the entire document shown in the active window. The Command-key equivalent for the Select All menu item is COMMAND-A.

Commando — Build commands semiautomatically. Choose this after entering the command name at the start of a line. Afterward, a dialog box is displayed that depicts all the features of the command. Upon exiting the dialog, the command line that you started to specify at the outset is changed to include all of the flag options and arguments that were generated with the help of the Commando dialog. The Command-key equivalent for the Commando menu item is COMMAND-K.

The Window Menu

The menu items in the Window menu help you arrange and display `CommandShell` windows. The menu is divided into three parts. The upper part of the menu contains menu items that help you arrange windows in various formats. The middle part contains menu items that help you size and order the windows. The lower part contains a list of all windows currently available in

CommandShell. When you choose one of the window names in the lower list, CommandShell makes the corresponding window the active window. The names of currently available windows are listed in the order they were originally created.

The menu items in the top part of the menu do the following:

Tile — Position windows in a right-to-left, then top-to-bottom sequence. You must have more than one window on the desktop to use this menu option. The Command-key equivalent for the Tile menu item is COMMAND-T.

Tile Horizontal — Position windows from top to bottom on the screen in their creation order. The windows are stretched to fit the width of the screen. The height of each window is adjusted to accommodate the number of windows.

Tile Vertical — Position windows from left to right on the screen in their creation order. The windows are stretched to fit the height of the screen. The width is adjusted to accommodate the number of windows.

Standard Positions — Reposition the windows in the original stacked order, from front to back.

The items in the middle part of the menu do the following:

Standard Size — Resize a window to its original dimensions. The Command-key equivalent for the Standard Size menu item is COMMAND-S.

Full Height — Stretch a window to the full height of the screen. The Command-key equivalent for the Full Height menu item is COMMAND-F.

Zoom Window — Make the window larger. The window automatically is resized to fit the whole screen. You can return a window to its previous size by choosing the Zoom Window menu item again.

Hide *window-name* — Make *window-name* temporarily disappear. The window is no longer visible, but is still available. To show a window that has been hid-

den, choose the window name from the Window menu again. The window appears in front of other open windows and becomes the active window. The Command-key equivalent for the Hide Terminal item is COMMAND-H.

Show All Windows
: Show all windows that have been hidden.

Last Window
: Makes the previously active window the active window once again, making it visible if it was hidden, and making it the recipient for any keyboard or mouse inputs. Repeating this selection, returns the windows to their original states. The Command-key equivalent for the Last Window menu item is COMMAND-L.

Rotate Window
: Move the rear window to the front of all the other windows. The Command-key equivalent for the Rotate Window menu item is COMMAND-R.

The menu items in the bottom part of this menu are window names for all currently available `CommandShell` windows. Selecting a window name makes it the active window, which also makes it visible as the front window. Among any other windows listed is the ever-present A/UX console window.

A/UX System Console
: Make the A/UX console window the active window. This window is used to view console messages. The Command-key equivalent for the A/UX System Console menu item is COMMAND-0 (zero).

The Fonts Menu

The Fonts menu lets you choose the type of font and the point size of the font for text entered or displayed in the active `CommandShell` window.

The Commands Menu

The menu items in the Commands menu help you set defaults for recording information and allow you to clean up the screen.

Don't Record Lines Off Top/Record Lines Off Top
: When `CommandShell` is invoked, it is set to record a preset number of lines as they scroll

out of view. If you do not want to store the lines for possible review later, you can stop the recording of lines for a particular window. If the default setting is left unchanged, the menu item Don't Record Lines Off Top appears in the Commands menu. Choose it so the lines are not recorded. If the lines are not being recorded, the menu item Record Lines Off Top appears in the menu. Choose it to record the lines off the top of the window as they scroll out of view from a particular window.

Clear Lines Off Top
: Erase recorded lines and make them no longer available for review within a particular window. The scroll bar disappears in the active `CommandShell` window.

Redraw Screen
: Clean up the screen if textual output affects the bitmapped display.

The Preferences Menu

The menu items in the Preferences menu help you select how you want to be notified of system messages, choose your default window settings, and allow you to set your preferred window configuration.

Notification Levels...
: Set the notification level for console messages. A dialog box appears in which you choose how you are notified of console messages. The choices are an alert box, a flashing icon in the menu bar, or both.

New Window Settings...
: Specify the default title prefix, window cascade origin, window size, font name, font size, and number of lines saved off the top of the window. A dialog box appears for these specifications.

Active Window Settings...
: Specify the settings for the active window. A dialog box appears allowing you to specify window title, size, and position; whether or not to save lines off the top; and the initial command to run in the window when it opens.

Managing CommandShell preferences

Preferences are normally saved in the `.cmdshellprefs` file located in the home directory. To maintain more than one set of preferences, you can establish a different filename as the file for storing preferences. For example, to allow one set of preferences (window sizes and so forth) to be saved for use with a large display device and another set of preferences to be saved for use with a smaller display device, you can reset the `CommandShell` variable that controls the file that is used to maintain these settings. The name of this variable is `CMDSHELLPREFS`. This variable can be set to be something other than `.cmdshellprefs`. When reset, the variable is normally assigned the name of a file relative to the user's home directory, although an absolute pathname is also acceptable. The choice of an absolute pathname is particularly helpful when your home directory is served to you through a file server which you access through several different A/UX systems. In such a case, each system can contain a preferences file that is customized for its own hardware.

FLAG OPTIONS

When invoked without any flag options, `CommandShell` starts with one active window on the desktop. The window is titled "CommandShell 1." The following flag options modify this default behavior:

`-b` *pid*

Used at boot time to start `CommandShell` in a background layer without any windows. With this option, `CommandShell` accepts window management commands only as far as permitting the displaying of the A/UX console window. You do this by choosing CommandShell from the Apple menu and then choosing A/UX System Console from the Window menu. No command-interpretation windows can be displayed. The treatment of kernel errors or requests for input are treated as has been described previously. *pid* is the process ID of `macsysinit` (see `brc`(1M)). After taking control of the system console, `CommandShell` sends a signal to this PID to exit. This signal heralds the continuation of all the remaining startup processes, which can continue with the assurance that alert boxes will be produced as necessary to notify users of the need for interaction with messages directed to the system console.

-u Specify that a user has logged in and start CommandShell in a background layer. When CommandShell is made the active application, the user's preferred (or default) CommandShell window layout is established. The default is to open a single CommandShell window. When started this way, individual user preferences can be stored using Save Preferences in the File menu.

-q Do not include a Quit menu item in the File menu and bring CommandShell into the foreground.

FILES

/mac/bin/CommandShell	object file
/mac/bin/%CommandShell	resource fork
$HOME/.cmdshellprefs	default preferences file

SEE ALSO

StartMonitor(1M), brc(1M), startmac(1M), startmsg(1M), termcap(4).

NAME

compact, uncompact, ccat — compress and uncompress files

SYNOPSIS

compact [*name...*]

uncompact [*name...*]

ccat [*file...*]

DESCRIPTION

compact compresses the named files using an adaptive Huffman code. If no filenames are given, the standard input is compacted to the standard output. compact operates as an on-line algorithm. Each time a byte is read, it is encoded immediately according to the current prefix code. This code is an optimal Huffman code for the set of frequencies seen so far. It is unnecessary to prefix a decoding tree to the compressed file since the encoder and the decoder start in the same state and stay synchronized. Furthermore, compact and uncompact can operate as filters. In particular, the command sequence

```
| compact | uncompact |
```

operates as a (very slow) no-op.

When an argument *file* is given, it is compacted and the resulting file is placed in *file*.C. *file* is unlinked. The first two bytes of the compacted file code the fact that the file is compacted. This code is used to prohibit recompaction.

The amount of compression to be expected depends on the type of file being compressed. Typical values of compression are: Text (38%), Pascal Source (43%), C Source (36%) and Binary (19%). These values are the percentages of file bytes reduced.

uncompact restores the original file from a file compressed by compact. If no filenames are given, the standard input is uncompacted to the standard output.

ccat cats the original file from a file compressed by compact, without uncompressing the file.

RESTRICTION

The last segment of the filename must contain fewer than thirteen characters to allow space for the appended .C.

d-file	Created during the execution of `delta`; removed after completion of `delta`.
`/usr/bin/bdiff`	Program to compute differences between the "gotten" file and the *g-file*.

Note: All files of the form *?*`-file` are explained in the "SCCS Reference" in *A/UX Programming Languages and Tools, Volume 2*. The naming convention for these files is also described there.

SEE ALSO

`admin`(1), `bdiff`(1), `cdc`(1), `get`(1), `help`(1), `prs`(1), `rmdel`(1), `sccs`(1), `sccsfile`(4).
"SCCS Reference" in *A/UX Programming Languages and Tools, Volume 2*.

DIAGNOSTICS

Use `help`(1) for explanations.

NAME

derez — decompile a resource file

SYNOPSIS

derez [*option*]... *resource-file* [*resource-description-file*]...

DESCRIPTION

derez creates a text representation (*resource description*) of a compiled resource file according to the resource type declarations in the *resource-description-file*s.

The *resource-file* parameter specifies the name of the file containing the compiled resources. You must specify a resource file; derez never reads the standard input.

The *resource-description-file* parameter specifies one or more files containing the type declarations used by the resource file. derez can provide meaningful output only if you provide the type declarations.

The type declarations in the resource description file follow the same format as that used by the resource compiler, rez. The type declarations for standard Macintosh resources are contained in the files types.r and systypes.r in the directory /mac/lib/rincludes. If you do not specify a resource description file, the output consists of data statements giving the resource data in hexadecimal form, without any additional format information.

The resource description is written to standard output. If the output of derez is used as input to rez with the same resource description files, it produces the same resource file that was originally input to derez. derez is not guaranteed to be able to run a declaration backward: if it can't, it produces a data statement instead of the appropriate resource statement.

derez ignores all include (but not #include), read, data, and resource statements found in the *resource-description-file*. (It still parses these statements for correct syntax.) Appendix C in the *A/UX Toolbox: Macintosh ROM Interface*, "Resource Compiler and Decompiler," describes the format of resource type declarations.

The resource description consists of resource and data statements that can be understood by rez.

If no errors or warnings are detected, `derez` runs silently. Errors and warnings are written to standard error (see `intro`(3S) in *A/UX Programmer's Reference*).

`derez` returns one of the following status values.

0 No errors

1 Error in parameters

2 Syntax error in file

3 I/O or program error

You may specify one or more of the following options.

`-c[ompatible]`
Generate output that is backward-compatible with `rez` 1.0.

`-d[efine]`*macro*`[=`*data*`]`
Define the macro variable *macro* to include the value *data*. If *data* is omitted, then *macro* is set to the null string (note that this still means that macro is defined). The `-d` flag option is the same as writing

```
#define macro [data]
```

at the beginning of the input. The `-d` flag option may be repeated any number of times.

`-e[scape]`
Do not escape characters that are normally escaped (such as: `\0xff`). Instead, print these characters as extended Macintosh characters. Note that not all fonts have defined all the characters.

Normally, characters with values between 0x20 and 0xD8 are printed as Macintosh characters. With the `-e` option, however, all characters (except null, newline, tab, backspace, form feed, vertical tab, and rubout) are printed as characters, not as escape sequences.

`-i`*pathname*(s)
Search the specified directories for `include` files. You may specify more than one pathname. The paths are searched in the order they appear on the command line.

To decompile an A/UX Toolbox resource file, use this pathname:

```
/:mac:lib:rincludes
```

`-m[axstringsize]` ***n***
Set the maximum string size to ***n***; ***n*** must be in the range 2–120. This setting controls how wide the strings can be in the output.

`-o[nly]` *type-expr*`[(`*ID1*`[:`*ID2*`])]`
`-o[nly]` *type-expr*`[`*resourceName*`]`
Read only resources of resource type *type-expr*. If an ID, range of IDs, or resource name is given, read only those resources for the given type. This option may be repeated.

Note: type-expr is an expression, so straight quotes (') might be needed. If an ID, range of IDs, or name is given, the entire option parameter must be quoted. For example,

```
derez -only "'MENU'(1:128)" ...
```

See also the EXAMPLES later in this section.

Note: The `-only` flag option cannot be specified together with the `-skip` flag option.

`-o[nly]` *type*
Read only resources of the specified type. This is a simpler version of the above option. No quotes are needed to specify a literal type as long as it starts with a letter. Do not use escape characters or other special characters. For example,

```
derez -only MENU ...
```

`-p` Display progress and version information.

`-rd`
Suppress warning messages if a resource type is redeclared.

`-s[kip]` *type-expr* `[(`*ID1*`[:`*ID2*`])]`
`-s[kip]` *type-expr* `[`*resourceName*`]`
Skip resources of type *type-expr* in the resource file. For example, you can save execution time by skipping `CODE` resources. The `-s` option may be repeated any number of times.

Note: type-expr is an expression, so straight quotes (') might be needed. If an ID, range of IDs, or name is given, the entire option parameter must be quoted. See the note under `-only` *type-expr* earlier in this section.

`-s[kip]` *type*
Skip resources of the specified type. This is a simpler version of the `-s` option. No quotes are needed to specify a literal as long as it starts with a letter.

`-u[ndef]` *macro*
Undefine the macro variable *macro*. This is the same as writing

```
#undef macro
```

at the beginning of the input file. (See Appendix C in *A/UX Toolbox: Macintosh ROM Interface* for a description of macro variables.) It is meaningful to undefine only the preset macro variables. This option may be repeated.

EXAMPLES

The command

```
derez -i /:mac:lib:rincludes sample types.r > sample.r
```

decompiles the resource file `%sample`, using the definitions in the file `/mac/lib/rincludes/types.r` and writes the output to the file `sample.r`. If it has access to the type definitions, `derez` generates more meaningful output.

```
derez -o MENU -i /:mac:lib:rincludes sample types.r
```

displays all of the MENU resources in `%sample`. The type definition for MENU resources is in the file `types.r`.

FILES

`/mac/bin/derez`

SEE ALSO

A/UX Toolbox: Macintosh ROM Interface.

NOTES

This command is not supported in 24-bit mode and must be run from the command line while logged into the Macintosh environment with the 32-bit session type.

NAME

deroff — remove nroff/troff, tbl, and eqn constructs

SYNOPSIS

deroff [-m*x*] [-w] [*file*...]

DESCRIPTION

deroff reads each of the *files* in sequence and removes all troff(1) requests, macro calls, backslash constructs, eqn(1) constructs (between .EQ and .EN lines, and between delimiters), and tbl(1) descriptions, perhaps replacing them with white space (blanks and blank lines), and writes the remainder of the file on the standard output. deroff follows chains of included files (.so and .nx troff commands); if a file has already been included, a .so naming that file is ignored and a .nx naming that file terminates execution. If no input file is given, deroff reads the standard input.

The -m flag option may be followed by an m, s, or l. The -mm flag option causes the macros be interpreted so that only running text is output (that is, no text from macro lines.) The -ml flag option forces the -mm flag option and also causes deletion of lists associated with the mm macros.

If the -w flag option is given, the output is a word list, one *word* per line, with all other characters deleted. Otherwise, the output follows the original, with the deletions mentioned above. In text, a *word* is any string that contains at least two letters and is composed of letters, digits, ampersands (&), and apostrophes ('); in a macro call, however, a *word* is a string that *begins* with at least two letters and contains a total of at least three letters. Delimiters are any characters other than letters, digits, apostrophes, and ampersands. Trailing apostrophes and ampersands are removed from words.

EXAMPLES

The command

```
deroff textfile
```

removes all nroff, troff, and macro definitions from textfile.

FILES

/usr/bin/deroff

NAME

enable, disable — enable or disable LP printers

SYNOPSIS

enable *printers*
disable [-c] [-r[*reason*]] *printers*

DESCRIPTION

enable activates the named *printers*, enabling them to print requests taken by lp(1). Use lpstat(1) to find the status of printers.

disable deactivates the named *printers*, disabling them from printing requests taken by lp(1). By default, any requests that are currently printing on the designated printers will be reprinted in their entirety either on the same printer or on another member of the same class. Use lpstat(1) to find the status of printers. Flag options useful with disable are:

-c Cancel any requests that are currently printing on any of the designated printers.

-r [*reason*]
Associates a *reason* with the deactivation of the printers. This reason applies to all printers mentioned up to the next -r flag option. If the -r flag option is not present or the -r flag option is given without a reason, then a default reason will be used. *reason* is reported by lpstat(1).

FILES

/usr/bin/enable
/usr/bin/disable
/usr/spool/lp/*

SEE ALSO

lp(1), lpstat(1).
A/UX Local System Administration

NAME

enscript — convert text files to POSTSCRIPT format for printing

SYNOPSIS

enscript [-12BGghKklmoqRr] [-L*lines*] [-f*font*] [-F*hfont*] [-b*header*] [-p*out*] [*spoolopts*] [*files*]

DESCRIPTION

enscript reads plain text files, converts them to POSTSCRIPT format, and spools them for printing on a POSTSCRIPT printer. Fonts, headings, and limited formatting options may be specified.

For example:

```
enscript -paleph boring.txt
```

processes the file called boring.txt for POSTSCRIPT printing, writing the output to the file aleph.

```
enscript -2r boring.c
```

prints a two-up landscape listing of the file called boring.c on the default printer (see below).

Font specifications have two parts: A font name as known to POSTSCRIPT (for example, Times-Roman, Times-Roman BoldItalic , Helvetica, Courier), and a point size (1 point=1/72 inch). So, Courier-Bold8 is 8 point Courier Bold, Helvetical2 is 12 point Helvetica.

The environment variable ENSCRIPT may be used to specify defaults. The value of ENSCRIPT is parsed as a string of arguments before the arguments that appear on the command line. For example

```
ENSCRIPT='-fTimes-Roman8'
```

sets your default body font to 8 point Times Roman.

The possible options are:

-2 set in two columns.

-1 set in one column (the default).

-r rotate the output 90 degrees (landscape mode). This is good for output that requires a wide page or for program listings when used in conjunction with –2.

```
enscript -2r files
```

is a nice way to get program listings.

`-R` don't rotate, also known as portrait mode (the default).

`-G` print in gaudy mode: causes page headings, dates, page numbers to be printed in a flashy style, at some slight performance expense.

`-l` simulate a line printer: make pages 66 lines long and omit headers.

`-B` omit page headings.

`-b`*header*
sets the string to be used for page headings to *header*. The default header is constructed from the file name, its last modification date, and a page number.

`-L`*lines* set the maximum number of lines to output on a page. `enscript` usually computes how many to put on a page based on point size, and may put fewer per page than requested by *lines*.

`-f`*font* sets the font to be used for the body of each page. Defaults to `Courier10`, unless two column rotated mode is used, in which case it defaults to `Courier7`.

`-F`*hfont* sets the font to be used for page headings. Defaults to `Courier-Bold10`.

`-p`*out* causes the POSTSCRIPT file to be written to the named file rather than being spooled for printing. As a special case, `-p -` will send the POSTSCRIPT to the standard output.

`-g` enables the printing of files containing non-printing characters. Any file with more than a small number of non-printing characters is suspected of being garbage and is not printed unless this option is used.

`-m` sends mail via `mail`(1) after the files have been printed. By default, no mail is sent upon normal completion of the print request.

`-o` If `enscript` cannot find characters in a font, the missing characters are listed.

`-q` causes `enscript` to be quiet about what it is doing. `enscript` won't report about pages, destination, omitted characters, and so forth. Fatal errors are still reported to the standard error output.

-k enables page prefeed (if the printer supports it). This allows simple documents (e.g., program listings in one font) to print somewhat faster by keeping the printer running between pages.

-K disable page prefeed (the default).

-h suppress printing of job burst page.

ENVIRONMENT

ENSCRIPT	string of options to be used by enscript.
PSLIBDIR	path name of a directory to use instead of /usr/lib/ps for enscript prologue and font metric files.
PSTEMPDIR	path name of temporary directory to use instead of XPSTEMDIRX of spooled temporary files.
LPDEST	the name of a printer for lp to use. If LPDEST is not set, enscript will spool to a printer class named PostScript.

FILES

/usr/bin/enscript	
/usr/lib/ps/*.afm	font metrics files.
/usr/lib/ps/enscript.pro	prologue for enscript files.

SEE ALSO

cancel(1), lp(1), lpr(1), lprm(1), lpstat(1), pr(1), ps630(1), getopt(3).

FEATURES

Options and the ENSCRIPT environment string are parsed in getopt(3) fashion.

BUGS

Long lines are truncated. Line truncation may be off by a little bit as printer margins vary. There should be a "wrap" option and multiple (truncated or wrapped) columns.

BUGS

After argument processing by the shell, `expr` cannot tell the difference between an operator and an operand except by the value. If `$a` is an =, the command:

```
expr  $a  =  '='
```

looks like:

```
expr  =  =  =
```

as the arguments are passed to `expr` (and they will all be taken as the = operator). The following works:

```
expr  X$a  =  X=
```

NAME

f77 — Fortran 77 compiler

SYNOPSIS

f77 [-l] [-66] [-A *factor*] [-c] [-C] [-E] [-f] [-F] [-g] [-I[24s]] [-m] [-o*output*] [-O] [-onetrip] [-p] [-R] [-S] [-u] [-U] [-w] *file* ...

DESCRIPTION

f77 is the Fortran 77 compiler; it accepts several types of *file* arguments:

Arguments whose names end with .f are taken to be Fortran 77 source programs; they are compiled and each object program is left in the current directory in a file whose name is that of the source, with .o substituted for .f. However, if a single Fortran source program is compiled and loaded all at one time, the .o file is deleted. By default, the process produces an executable file, named a.out, in the current directory

Arguments whose names end with .r or .e are taken to be EFL source programs; these are first transformed by the EFL preprocessor, then compiled by f77, producing .o files.

In the same way, arguments whose names end with .c or .s are taken to be C or assembly source programs and are compiled or assembled, producing .o files.

The following flag options have the same meaning as in cc(1) (see ld(1) for link editor flag options):

-A *factor* Expand the default symbol table allocations for the assembler and link editor. The default allocation is multiplied by the factor given.

-c Suppress link editing and produce .o files for each source file.

-f In systems without floating-point hardware, use a version of f77 that handles floating-point constants and links the object program with the floating-point interpreter.

-g Generate additional information needed for the use of sdb(1)

-o*output* Name the final output file *output*, instead of a.out.

-O	Invoke an object code optimizer.
-p	Prepare object files for profiling (see prof(1)).
-S	Compile the named programs and leave the assembler language output in corresponding files whose names are suffixed with .s. (No .o files are created.)

The following flag options are specific to f77:

-onetrip	Perform all DO loops at least once. (Fortran 77 DO loops are not performed at all if the upper limit is smaller than the lower limit.)
-1	Same as -onetrip.
-66	Compile as a Fortran 66 program.
-C	Generate code for run-time subscript range-checking.
-E	The remaining characters in the argument are used as an EFL flag argument whenever processing a .e file.
-F	Apply EFL preprocessor to relevant files and put the result in files whose names have their suffix changed to .of. (No .o files are created.)
-I[24s]	Change the default size of integer variables (only valid on machines where the normal integer size is not equal to the size of a single precision real). -I2 causes all integers to be 2-byte quantities, -I4 (default) causes all integers to be 4-byte quantities, and -Is changes the default size of subscript expressions (only) from the size of an integer to 2 bytes.
-m	Apply the M4 preprocessor to each EFL source file before transforming with the efl(1) processor.
-U	Do not "fold" cases. F77 is normally a no-case language (i.e., a is equal to A). The -U flag option causes f77 to treat upper and lower cases separately.
-u	Make the default type of a variable *undefined*, rather than using the default Fortran rules.
-w	Suppress all warning messages. If the flag option is -w66, only Fortran 66 compatibility warnings are suppressed.

Other arguments are taken to be link editor flag option arguments, f77-compatible object programs (typically produced by an earlier run), or libraries of f77-compatible routines. These programs, together with the results of any compilations specified, are linked (in the order given) to produce an executable program with the default name a.out.

FILES

/usr/bin/f77	
file.[fresc]	input file
file.o	object file
a.out	linked output
./fort[*pid*].?	temporary
/usr/lib/f77comp	compiler
/lib/c2	optional optimizer
/usr/lib/libF77.a	intrinsic function library
/usr/lib/libI77.a	Fortran I/O library
/lib/libc.a	C library; see Section 3 in *A/UX Programmer's Reference.*

SEE ALSO

asa(1), cc(1), efl(1), fpr(1), fsplit(1), ld(1), m4(1), prof(1), sdb(1).
"f77 Reference" in *A/UX Programming Languages and Tools, Volume 1.*

DIAGNOSTICS

The diagnostics produced by f77 itself are self-explanatory. Occasional messages may be produced by the link editor ld(1).

NAME

fcnvt — convert a resource file to another format

SYNOPSIS

fcnvt [-v] [-i *input-format*] -s [-f] *input-file output-file*
fcnvt [-v] [-i *input-format*] -d [-f] *input-file output-file*
fcnvt [-v] [-i *input-format*] -t [-f] *input-file output-file*
fcnvt [-v] [-i *input-format*] -p [-f] *input-file output-file*
fcnvt [-v] [-i *input-format*] -b [-f] *input-file output-file*
fcnvt [-v] [-i *input-format*] -m [-f] *input-file output-file*

DESCRIPTION

fcnvt converts a file (*input-file*) from one file format to another (*output-file*). The command line options and their meanings are:

-i *input-format*
Specify the file format of the file to be converted. If an input-file format is not specified, the AppleSingle format is assumed. Supported formats are:

single
AppleSingle (see the -s flag option)

double
AppleDouble (see the -d flag option)

triple
Plain Triple (see the -t flag option)

pair
Plain Pair (see the -p flag option)

hex
BinHex 4.0 (see the -b flag option)

bin
MacBinary (see the -m flag option)

-v Specify verbose mode. In verbose mode, fcvnt displays information as it processes each file.

-s Create an AppleSingle-format output file. This format is the default. With AppleSingle format, all contents and file information are kept in a single file. AppleSingle format is best used for non-text data and executable Macintosh® object files. Directory listings look much cleaner because each Macintosh file maps to a single A/UX file with no % prefix.

`-d` Create an AppleDouble-format output file. With AppleDouble format, the contents of the data fork are stored in one file, known as the *data file*; resources and file attribute information are stored in a separate file, known as the *header file*. The header file has the same name as the data file but is prefixed with a percent sign (`%`). The AppleDouble format is best used for text data and data to be shared with UNIX utilities, because the data fork is available as an isolated file.

`-t` Convert the input file into Plain Triple file format. This format is used by the `macget` and `macput` public-domain file-transfer programs. Three files are created with suffixes attached to help tell them apart. The files *output-file*`.info`, *output-file*`.data`, and *output-file*`.rsrc` contain identification information, the data fork, and the resource fork, respectively.

`-p` Convert the input file into Plain Pair file format. This option is the same as the `-t` option except that *output-file*`.info` is not created.

`-b` Create a BinHex 4.0-format output file. The input-file is encoded into ASCII characters, permitting ASCII transfer of a binary file.

`-m` Create a MacBinary-format output file. This format is commonly used when transferring Macintosh files via a telecommunication link, using a protocol such as XMODEM, XMODEM7, Kermit, CompuServe A or CompuServe B.

`-f` Allow `fcnvt` to overwrite an existing file with the same name as the new file. If you specify an output filename that is the same as an existing filename without specifying the `-f` flag option, `fcnvt` takes no action and returns an error message.

If the initial transfer is made using a terminal emulator program, the file created is likely to be in a text-only format, BinHex 4.0 format, or MacBinary format, if not just a copy of the resource fork of the Macintosh file. In any of those cases, `fcnvt` can be used to convert the file to either AppleSingle or AppleDouble format, assuming you know their starting format.

Note that file transfers made using terminal emulators are likely to strip away the Macintosh type and creator attributes for the file. (Each of these attributes is one four-character string.) See

settc(1) to restore those attributes, once you know what they are supposed to be.

FILES

/mac/bin/fcnvt

SEE ALSO

settc(1).

fgrep(1)

See grep(1)

NAME

rev — reverse characters within each line of text

SYNOPSIS

rev [*file*]...

DESCRIPTION

rev copies the named files to the standard output, reversing the order of characters in every line. If no file is specified, the standard input is copied.

FILES

/usr/ucb/rev

NAME

rez — compile resources

SYNOPSIS

rez [*option*]... [*resource-description-file*]...

DESCRIPTION

rez creates a resource file according to a textual series of statements in the resource-description language developed for Macintosh resources. The resource-description language is described in an appendix of *A/UX Toolbox*.

The *resource-description-file* parameter represents the names of one or more files containing resource descriptions. If no filenames are specified, rez accepts keyboard input.

The data used to build the resource file can come directly from one or more resource-description files, from other text files (through `#include` and `read` directives in the resource-description file), and from other resource files (through the `include` directive in the resource-description file). The type declarations for standard Macintosh resources are contained in the files `types.r` and `systypes.r` located in the directory `/mac/lib/rincludes`.

rez includes macro processing, full expression evaluation, built-in functions, and system variables.

rez never sends output to standard output. By default, rez writes to a file named `rez.Out` in the current directory. You can specify a different output file with the `-o` option.

If no errors or warnings are detected, rez runs silently. Errors and warnings are written to standard error.

rez returns one of the following status values.

0 No errors

1 Error in parameters

2 Syntax error in file

3 I/O or program error

You may specify one or more of the following options.

`-align[word|longword]`
Align resources along word or longword boundaries. This may allow the Resource Manager to load these resources faster. The `-align` option is ignored when the `-a` option is in

effect.

-a[ppend]
Output from rez is appended to the output file instead of replacing the output file.

Note: rez overwrites any existing resource of the same type and ID without a warning message. rez cannot append resources to a resource file in which the Read Only bit is set. Also, rez cannot replace a resource file that has a protection bit set. See also the -ov option below.

-c[reator] *creator-expr*
Set the creator of the output file. (The default value is ????.)

-d[efine] *macro*[=*data*]
Define the macro variable *macro* to have the value *data*. If *data* is omitted, then *macro* is set to the null string (this still means that *macro* is defined). The -d option is the same as writing

```
#define macro [data]
```

at the beginning of the input.

-i*pathname(s)*
Search the specified *pathnames* for include files. You may specify more than one pathname. The paths are searched in the order they appear in the command line.

To reach the include files provided with the A/UX Toolbox, use this pathname

```
rez -i /:mac:lib:rincludes
```

-o *output-file*
Place the output in *output-file*. Specify the name of the associated data file; rez automatically affixes a percent sign (%) to the name of the header file containing the resources. The default output file is rez.Out.

-ov
Override the protected bit when replacing resources with the -a option.

-p[rogress]
Write version and progress information to diagnostic output.

`-rd`
Suppress warning messages if a resource type is redeclared.

`-ro`
Set the `mapReadOnly` flag in the resource map.

`-s` *pathname(s)*
Search the specified *pathnames* for resource include files.

`-t[ype]` *type-expr*
Set the type of the output file. The default value is `APPL`.

`-u[ndef]` *macro*
Undefine the macro variable *macro*. This is the same as writing

```
#undef macro
```

at the beginning of the input. It is meaningful to undefine the preset macro variables only. (See Appendix C of the document *A/UX Toolbox* for a description of macro variables.)

EXAMPLES

```
rez -i /:mac:lib:rincludes sample.r -o sample
```

generates a resource file for `sample`, based on the descriptions in `sample.r` and the include files in the directory `/mac/lib/rincludes`. Place the output in an AppleDouble header file named `%sample`.

FILES

`/mac/bin/rez`

SEE ALSO

`derez`(1).

NOTES

This command is not supported in 24-bit mode and must be run from the command line while logged into the Macintosh environment with the 32-bit session type.

NAME

rlog — display log messages and other information about RCS files

SYNOPSIS

rlog [-L] [-R] [-h] [-t] [-d*dates*] [-l[*lockers*]] [-r*revisions*] [-s*states*] [-w[*logins*]] *files*

DESCRIPTION

rlog displays information about RCS files. Files ending in ,v are RCS files, all others are working files. If a working file is given, rlog tries to find the corresponding RCS file, first in directory ./RCS and then in the current directory, as explained in co(1).

rlog displays the following information for each RCS file: RCS filename, working filename, head (that is, the number of the latest revision on the trunk), access list, locks, symbolic names, suffix, total number of revisions, number of revisions selected for display, and descriptive text. This is followed by entries for the selected revisions in reverse chronological order for each branch. For each revision, rlog displays revision number, author, date and time, state, number of lines added or deleted (with respect to the previous revision), locker of the revision (if any), and log message. Without options, rlog displays complete information. The options below restrict this output.

FLAG OPTIONS

The following flag options are interpreted by rlog:

-L Ignores RCS files that have no locks set; convenient in combination with -R, -h, or -l.

-R Prints only the name of the RCS file; convenient for translating a working filename into an RCS filename.

-h Prints only RCS filename, working filename, head, access list, locks, symbolic names, and suffix.

-t Prints the same as -h, plus the descriptive text.

-d*dates* Prints information about revisions with a checkin date and time in the ranges given by the semicolon-separated list of *dates*. A range of the form *d1*<*d2* or *d2*>*d1* selects the revisions that were deposited between *d1* and *d2* (inclusive). A range of the form <*d* or *d*> selects all revisions dated *d* or earlier. A range of the form *d*< or >*d* selects all revisions dated

d or later. A range of the form *d* selects the single, latest revision dated *d* or earlier. The date and time strings *d*, *d1*, and *d2* are in the free format explained in co(1). Quoting is normally necessary, especially for < and >. Note that the separator is a semicolon.

-l[*lockers*]
Prints information about locked revisions. If the comma-separated list *lockers* of login names is given, only the revisions locked by the given login names are displayed. If the list is omitted, all locked revisions are displayed.

-r*revisions*
Prints information about revisions given in the comma-separated list *revisions* of revisions and ranges. A range *rev1*-*rev2* means revisions *rev1* to *rev2* on the same branch, *-rev* means revisions from the beginning of the branch up to and including *rev*, and *rev-* means revisions starting with *rev* to the end of the branch containing *rev*. An argument that is a branch means all revisions on that branch. A range of branches means all revisions on the branches in that range.

-s*states* Prints information about revisions whose state attributes match one of the states given in the comma-separated list *states*.

-w[*logins*] Prints information about revisions checked in by users with login names appearing in the comma-separated list *logins*. If *logins* is omitted, the user's login is assumed.

rlog displays the intersection of the revisions selected with the options -d, -l, -s, and -w, intersected with the union of the revisions selected by -b and -r.

EXAMPLES

The following are some examples of using rlog.

```
rlog -L -R RCS/*,v
rlog -L -h RCS/*,v
rlog -L -l RCS/*,v
rlog RCS/*,v
```

The first command displays the names of all RCS files in the sub-

directory RCS which have locks. The second command displays the headers of those files, and the third displays the headers plus the log messages of the locked revisions. The last command displays complete information.

DIAGNOSTICS

The exit status always refers to the last RCS file operated upon, and is 0 if the operation was successful, 1 if otherwise.

DISCLAIMER

This reference manual entry describes a utility that Apple understands to have been released into the public domain by its author or authors. Apple has included this public domain utility for your convenience. Use it at your own discretion. Often the source code can be obtained if additional requirements are met, such as the purchase of a site license from an author or institution.

IDENTIFICATION

Author: Walter F. Tichy, Purdue University, West Lafayette, IN 47907.

SEE ALSO

ci(1), co(1), ident(1), rcs(1), rcsdiff(1), rcsintro(1), rcsmerge(1), rcsfile(4), sccstorcs(1M).

Walter F. Tichy, "Design, Implementation, and Evaluation of a Revision Control System," in *Proceedings of the 6th International Conference on Software Engineering*, IEEE, Tokyo, Sept. 1982.

NAME

rlogin — remote login

SYNOPSIS

rlogin *rhost* [-8] [-ec] [-l *username*]

DESCRIPTION

rlogin connects your terminal on the current local host system *lhost* to the remote host system *rhost* via a local network. On the remote side, you get whatever shell is set up for that account.

Each host has a file /etc/hosts.equiv which contains a list of *rhosts* with which it shares account names. (The hosts names must be the standard names as described in remsh(1N)). When you rlogin as the same user on an equivalent host, you don't need to give a password. Each user may also have a private equivalence list in a file .rhosts in his login directory. Each line in this file should contain a *rhost* and a *username* separated by a space, giving additional cases where logins without passwords are to be permitted. If the originating user is not equivalent to the remote user, then a login and password will be prompted for on the remote machine as in login(1). To avoid security problems, the .rhosts file must be owned by either the remote user or root. Note that, for security reasons, root is an exception to the above; a superuser on an equivalent host must still supply the password to login as root unless the root account has its own private equivalence list in a file .rhosts in the root directory. Note that a .rhosts file for a root account is not recommended where secure systems are required.

Your remote terminal type is the same as your local terminal type (as given in your environment TERM variable). All echoing takes place at the remote site, so that (except for delays) the rlogin is transparent. Flow control via CONTROL-S and CONTROL-Q and flushing of input and output on interrupts are handled properly.

The -8 flag option allows an eight-bit data path; otherwise parity bits are stripped.

Tilde (~) is the default escape character. A line of the form "~." (where "~" is the escape character), disconnects the current job from the remote host.

The escape sequence "~CONTROL-Z" stops the rlogin process and returns control to the local machine where the rlogin was initiated. This applies only if the initiating shell allows job control

NAME

sysline — display system status on status line of a terminal

SYNOPSIS

sysline [-b] [-c] [-d] [-D] [-e] [-h] [-H *remote*] [-i] [-j] [-l] [-m] [-p] [-q] [-r] [-s] [+*N*]

DESCRIPTION

sysline runs in the background and periodically displays system status information on the status line of the terminal. Not all terminals have a status line. Those that do include the h19, c108, aaa, vt100, tvi925/tvi950 and Freedom 100. Of these, only the h19 termcap entry supports the status line.

Note: The Macintosh monitor does not have a status line.

If no flags are given, sysline displays the time of day, the current load average, the change in load average in the last 5 minutes, the number of users (followed by u), the number of runnable process (followed by r) the number of suspended processes (followed by s), and the users who have logged on and off since the last status report. Finally, if new mail has arrived, a summary of it is printed. If there is unread mail in your mailbox, an asterisk will appear after the display of the number of users. The display is normally in reverse video (if your terminal supports this in the status line) and is right-justified to reduce distraction. Every fifth display is done in normal video to give the screen a chance to rest.

If you have a file named .who in your home directory, then the contents of that file is printed first. One common use of this feature is to alias chdir, pushd, and popd to place the current directory stack in ~/.who after it changes the new directory.

The following flag options may be used on the command line:

-b	Beep once every half hour and twice every hour, just like those obnoxious watches you keep hearing.
-c	Clear the status line for 5 seconds before each redisplay.
-d	Debug mode – print status line data in human-readable format
-D	Print out the current day/date before the time.

`-e`	Print out only the information. Do not print out the control commands necessary to put the information on the bottom line. This flag option is useful for putting the output of `sysline` onto the mode line of an `emacs` window.
`-H` *remote*	Print the load average on the remote host *remote*. If the host is down, or is not sending out `rwhod` packets, then the down time is printed instead.
`-h`	Print out the host machine's name after the time.
`-l`	Don't print the names of people who log in and out.
`-m`	Don't check for mail.
`-p`	Don't report the number of process which are runnable and suspended.
`-r`	Don't display in reverse video.
`+N`	Update the status line every N seconds. The default is 60 seconds.
`-q`	Don't print out diagnostic messages if something goes wrong when starting up.
`-i`	Print out the process ID of the `sysline` process onto standard output upon startup. With this information you can send the alarm signal to the `sysline` process to cause it to update immediately. `sysline` writes to the standard error, so you can redirect the standard output into a file to catch the process id.
`-s`	Print "short" form of line by left-justifying. *iff* escapes are not allowed in the status line. Some terminals (the tvi's and Freedom 100's, for example) do not allow cursor movement (or other "intelligent" operations) in the status line. For these terminals, `sysline` normally uses blanks to cause right-justification. This flag option will disable the adding of the blanks.
`-j`	Force the sysline output to be left-justified even on terminals capable of cursor movement on the status line.

If you have a file `.syslinelock` in your home directory, then `sysline` will not update its statistics and write on your screen, it will just go to sleep for a minute. This is useful if you want to disable `sysline` momentarily. Note that it may take a few seconds from the time the lock file is created until you are guaranteed that `sysline` will not write on the screen.

FILES

`/usr/ucb/sysline`	
`/etc/termcap`	database of terminal capabilities
`/etc/utmp`	names of people who are logged in
`/dev/kmem`	contains process table
`/usr/spool/rwho/whod.*`	`who/uptime` information for remote hosts
`${HOME}/.who`	information to print on bottom line
`${HOME}/.syslinelock`	when it exists, `sysline` will not print

SEE ALSO

`ps(1)`, `pstat(1)`.

BUGS

If you interrupt the display, you may find your cursor missing or stuck on the status line. The best thing to do is to reset the terminal.

If there is too much for one line, the excess is thrown away.

NAME

`systemfolder` — create a personal System Folder

SYNOPSIS

`systemfolder [-f]`

DESCRIPTION

`systemfolder` allows a user to create a personal Macintosh® System Folder in his or her home directory. If a personal System Folder already exists, it is updated with any files in the global System Folder that are not present in the personal Folder.

FLAG OPTIONS

`systemfolder` interprets one flag option, the force option.

`-f` By default, the System file is not updated if it already exists in the personal System Folder. If this option is used, this forces the update of the System file.

FILES

```
$HOME/System Folder
/mac/sys/System Folder/*
```

NAME

whatis — display a brief description for the named manual page entry

SYNOPSIS

whatis *command* ...

DESCRIPTION

whatis looks up a given command and gives the header line from the manual section. You may then run the man(1) command to get more information. If the line ends with

name (*section*) ...

you may type

man *section name*

to get the documentation for it. For instance, if you type

```
whatis ed
```

you will see

```
red ed(1)          - text editor
```

This tells you that the *section* for ed is 1. To see the manual entry for ed, you may then type

```
man 1 ed
```

and the entire entry will be printed.

FILES

/usr/ucb/whatis
/usr/lib/whatis

SEE ALSO

apropos(1), man(1), whereis(1), which(1).

NAME

whereis — locate source, binary, and online help file for a command

SYNOPSIS

whereis [-b] [-m] [-s] [-u] [-B *dir* [-f]] [-M *dir* [-f]] [-S *dir* [-f]] *name* ...

DESCRIPTION

whereis locates source/binary and manual sections for specified files. The supplied names are first stripped of leading pathname components and standard extensions for source files and manual entries, for example, .c and .1m. Prefixes of s. resulting from use of source code control are also dealt with. whereis then attempts to locate the desired program in a list of standard places. If any of the -b, -s, or -m flags are given, then whereis searches only for binaries, sources, or manual sections respectively (or any two thereof). The -u flag may be used to search for unusual entries. A file is said to be unusual if it does not have one entry of each requested type. Thus whereis -m -u * asks for those files in the current directory which have no documentation.

Finally, the -B, -M, and -S flags may be used to change or otherwise limit the places where whereis searches. The -f file flag is used to terminate the last such directory list and signal the start of file names.

EXAMPLES

The following finds all the files in /usr/bin which are not documented in /usr/man/man1 with source in /usr/src/cmd:

```
cd /usr/ucb
whereis -u -M /usr/man/man1 -S /usr/src/cmd -f *
```

FILES

```
/usr/bin/whereis
/usr/src/*
/usr/catman/man/*
/bin
/lib
/etc
/usr/bin
/usr/lib
/usr/etc
/usr/ucb
```

NAME

wait3 — wait for child process to stop or terminate

SYNOPSIS

```
#include <sys/wait.h>

int wait3(status, options, 0)
union wait *status;
int options;
```

DESCRIPTION

wait3 provides an interface for programs which must not block when collecting the status of child processes. The *status* parameter is defined as above. The *options* parameter is used to indicate the call should not block if there are no processes which wish to report status (WNOHANG), and/or that children of the current process that are stopped due to a SIGTTIN, SIGTTOU, SIGTSTP, or SIGSTOP signal should also have their status reported (WUNTRACED).

When the WNOHANG option is specified and no processes wish to report status, wait3 returns a *pid* of 0. The WNOHANG and WUNTRACED options may be combined by ORing the two values.

The declaration of "union wait" is found in <sys/wait.h>. The third argument, 0, is a placeholder. The "normal case" is the same as wait(2).

RETURN VALUE

wait3 returns −1 if there are no children not previously waited for; 0 is returned if WNOHANG is specified and there are no stopped or exited children.

SEE ALSO

exit(2), wait(2).

NAME

waitpid — wait for child process to stop or terminate

SYNOPSIS

```
#include <sys/types.h>
#include <sys/wait.h>

pid_t waitpid(pid, stat_loc, options)
pid_t pid;
int *stat_loc, options;
```

DESCRIPTION

The waitpid function allows the calling process to obtain status information regarding one of its direct descendants (child processes), created by calling fork(2). Unless an error occurs, the calling process is suspended until one of the specified children stops or terminates. waitpid provides additional functionality over wait(2): control over which children will be waited for, non-blocking status collection and collection of the status of children that are stopped.

The *pid* argument specifies the set of child processes from which status may be returned.

If *pid* is equal to −1, all children of the calling process are included in the set.

If *pid* is greater than zero, the set consists solely of the child process whose process ID is equal to *pid*.

If *pid* is equal zero, all children whose process group ID matches that of the calling process are included in the set.

If *pid* is less than −1, the set consists of all child processes whose process group ID is equal to the absolute value of *pid*.

If *stat_loc* (taken as an integer) is not NULL, 16 bits of information called *status* are stored in the low order 16 bits of the location pointed to by *stat_loc*. *status* can be used to differentiate between stopped and terminated child processes. If the child process terminated, *status* identifies the cause of termination and passes useful information to the parent. This is accomplished in the following manner:

If the child process stopped, the high order 8 bits of *status* will contain the number of the signal that caused the process to stop and the low order 8 bits will be set equal to 0177.

If the child process terminated due to an `exit` call, the low order 8 bits of *status* will be zero and the high order 8 bits will contain the low order 8 bits of the argument that the child process passed to `exit`(2).

If the child process terminated due to a signal, the high order 8 bits of *status* will be zero and the low order 8 bits will contain the number of the signal that caused the termination. In addition, if the low order seventh bit (i.e., bit 200) is set, a "core image" will have been produced; see `signal`(3).

If *pid* is equal to −1 and *options* is zero, the behavior of `waitpid` is identical to `wait`(2). Otherwise, *options* consists of the logical OR of one or both of the following flags:

`WNOHANG` Return immediately, even if there are no children to wait for. In this case, a return value of zero indicates that no children have terminated (or stopped, if `WUNTRACED` is also set).

`WUNTRACED` Return the status of stopped children. If the child process has stopped due to the delivery of a `SIGTTIN`, `SIGTTOU`, `SIGTSTP`, or `SIGSTOP` signal, its status may be collected by setting this flag.

If `WUNTRACED` is set and the *status* of a stopped child process is reported, the high order 8 bits of *status* shall contain the number of the signal that caused the process to stop and the low order eight bits shall be set to the octal value 0177.

RETURN VALUE

If `waitpid` returns successfully, i.e., one of the specified children has stopped or terminated, the return value shall equal the child's process ID. If `WNOHANG` is specified and there are no stopped or terminated children, the return value will be set to zero. Otherwise, an error has occurred: the value of −1 shall be returned and `errno` shall be set to indicate the error.

ERRORS

If any of the following conditions occur, `waitpid` will return −1 and set `errno` to the corresponding value:

`[ECHILD]` The calling process has no existing unwaited-for child processes.

`[EINTR]` The calling process was interrupted while executing the `waitpid` function.

[EINVAL] waitpid was called with an invalid *options* value.

SEE ALSO

exit(2), wait(2), wait3(2N).

NAME

write, writev — write on a file

SYNOPSIS

```
int write(fildes, buf, nbytes)
int fildes;
char *buf;
unsigned nbytes;

#include <sys/types.h>
#include <sys/uio.h>

int writev(fildes, iov, ioveclen)
int fildes;
struct iovec *iov;
int ioveclen;
```

DESCRIPTION

write attempts to write *nbytes* bytes from the buffer pointed to by *buf* to the file associated with *fildes*. writev performs the same action, but gathers the output data from the *iovlen* buffers specified by the members of the iovec array: iov[0], iov[1], and so on.

The file descriptor *fildes* is obtained from a creat, open, dup, fcntl, pipe, or socket system call.

On devices capable of seeking, the actual writing of data proceeds from the position in the file indicated by the file pointer. On return from write, the file pointer is incremented by the number of bytes actually written.

On devices incapable of seeking, writing always starts at the current position. The value of a file pointer associated with such a device is undefined.

If the O_APPEND flag of the file status flags is set, the file pointer is set to the end of the file prior to each write.

When writing to a pipe (or FIFO), write requests of PIPE_BUF bytes or less are not interleaved with data from other processes writing to the same pipe. Writes of greater than PIPE_BUF bytes may have data interleaved, on arbitrary boundaries, with writes by other processes.

NAME

dim, ddim, idim — Fortran positive difference intrinsic functions

SYNOPSIS

```
integer a1, a2, a3
a3=idim(a1, a2)

real a1, a2, a3
a3=dim(a1, a2)

double precision a1, a2, a3
a3=ddim(a1, a2)
```

DESCRIPTION

These functions return:

a1-a2	if *a1* > *a2*
0	if *a1* <= *a2*

NAME

opendir, readdir, telldir, seekdir, rewinddir, closedir — directory operations

SYNOPSIS

```
#include <sys/types.h>
#include <dirent.h> or #include <sys/dir.h>

DIR *opendir(filename)
char *filename;

struct direct *readdir(dirp)
DIR *dirp;

long telldir(dirp)
DIR*dir

seekdir(dirp, loc)
DIR*dirp
long loc;

void rewinddir(dirp)
DIR *dirp;

int closedir(dirp)
DIR *dirp;
```

DESCRIPTION

`opendir` opens the directory named by *filename* and associates a directory stream with it. `opendir` returns a pointer to be used to identify the directory stream in subsequent operations. The pointer NULL is returned if *filename* cannot be accessed, or if it cannot allocate enough memory to hold the whole thing.

`readdir` returns a pointer to the next directory entry. It returns NULL upon reaching the end of the directory or detecting an invalid `seekdir` operation.

`telldir` returns the current location associated with the named *directory stream.*

`seekdir` sets the position of the next `readdir` operation on the *directory stream.* The new position reverts to the one associated with the *directory stream* when the `telldir` operation was performed. Values returned by `telldir` are good only for the lifetime of the `DIR` pointer from which they are derived. If the directory is closed and then reopened, the `telldir` value may be invalidated due to undetected directory compaction. It is safe to use a previous `telldir` value immediately after a call to `opendir`

and before any calls to `readdir`.

`rewinddir` resets the position of the named directory stream to the beginning of the directory. It also causes the directory stream to refer to the current state of the directory.

`closedir` closes the named directory stream and frees the structure associated with the `DIR` pointer.

Sample code that searches a directory for an entry *name* is:

```
len = strlen(name);
dirp = opendir(".");
for (dp = readdir(dirp); dp != NULL; dp = readdir(dirp))
    if (dp->d_namlen == len && !strcmp(dp->d_name, name)) {
        closedir(dirp);
        return FOUND;
}
closedir(dirp);
return NOT_FOUND;
```

The result of using a directory stream after an `exec`(2) is undefined. After a `fork`(2), either the parent or the child (but not both) may continue processing the directory stream by using `readdir` or `rewinddir`, or both.

In the POSIX environment, the `FD_CLOEXEC` flag is set on the file descriptor underlying the directory stream.

SEE ALSO

`ls`(1), `open`(2), `close`(2), `fcntl`(2), `getdirentries`(2), `read`(2), `lseek`(2), `dir`(4).

NAME

dprod — Fortran double precision product intrinsic function

SYNOPSIS

```
real a1, a2
double precision a3
a3=dprod(a1, a2)
```

DESCRIPTION

dprod returns the double precision product of its real arguments.

NAME

sleep — suspend execution for interval

SYNOPSIS

```
unsigned sleep(seconds)
unsigned seconds;
```

DESCRIPTION

sleep suspends the current process from execution for the number of *seconds* specified by the argument. The actual suspension time may be less than that requested for two reasons: (1) scheduled wakeups occur at fixed 1-second intervals, (on the second, according to an internal clock) and (2) any caught signal will terminate sleep following execution of the signal catching routine. The suspension time may be longer than requested by an arbitrary amount, due to the scheduling of other activity in the system. The value returned by sleep is the "unslept" amount (the requested time minus the time actually slept) in case the caller had an alarm set to go off earlier than the end of the requested sleep time or in case there is premature arousal due to another caught signal.

The routine is implemented by setting an alarm signal and pausing until it (or some other signal) occurs. The previous state of the alarm signal is saved and restored. The calling program may have set up an alarm signal before calling sleep. If the sleep time exceeds the time before the alarm signal, the process sleeps only until the alarm signal would have occurred and the caller's alarm catch routine is executed just before the sleep routine returns. If the sleep time is less than the time before the calling program's alarm, the prior alarm time is reset to go off at the same time it would have without the intervening sleep.

SEE ALSO

alarm(2), pause(2), signal(3).

NAME

slots — ROM library functions

SYNOPSIS

cc [*flags*] *files* -lslots [*libraries*]

DESCRIPTION

The routines in the slots library provide access to board slot ROM from user processes and slot device drivers. Calls to library routines do not require knowledge of either the board ROM configuration or the ROM addressing requirements.

slot_PRAM_init(*slot, data*)
Read the PRAM init structure for *slot* into the buffer pointed to by *data.*

slot_board_flags(*slot*)
Read and return the board flags for *slot.*

slot_board_id(*slot*)
Read and return the board ID number for *slot.*

slot_board_name(*slot, data, size*)
Read up to *size* bytes of the board name string for *slot* into the buffer pointed to by *data.*

slot_board_type(*slot, data*)
Read and return the unsigned 64 bit or 8 byte board type for *slot* into the buffer pointed to by *data.*

slot_board_vendor_info(*kind, slot, data, size*)
For *slot* get *size* bytes of the vendor information string of type *kind* into the buffer pointed to by *data.*

slot_directory(*slot, data, size*)
For *slot* read the resource directory into the buffer of *size* entries pointed to by *data.*

slot_ether_addr(*slot, data*)
For *slot* read 6 bytes of ethernet address into the buffer pointed to by *data.*

slot_part_num(*slot, data, size*)
For *slot* get *size* bytes of the part number string into the buffer pointed to by *data.*

slot_primary_init(*slot, data*)
For *slot* read the primary init structure into the buffer pointed to by *data.*

`slot_rev_level` (*slot, data, size*)
For *slot* get *size* bytes of the revision level of the ROM into the buffer pointed to by *data.*

`slot_serial_num` (*slot, data, size*)
For *slot* get *size* bytes of serial number string into the buffer pointed to by *data.*

`slot_test` (*slot, string, size, c, s, l*)
For *slot*, `slot_test` returns the value of each Test resource in its respective data type. *string* is a character array, *c* is a `char`, *s* is a `short`, and *l* is a `long`. `char`, `short`, and `long` must be passed to `slot_test` by address. A negative return value indicates that *slot* does not have a Test resource. A positive return value indicates the number of available Test resources.

`slot_vendor_id` (*slot, data, size*)
For *slot* read *size* bytes of vendor ID string into the buffer pointed to by *data.*

SEE ALSO
Building A/UX Device Drivers

NOTE
The slots library is only accessible to processes with superuser privileges due to the required `phys` call to access board ROM.

SEE ALSO

pathconf(3P).

NAME

syslog, openlog, closelog, setlogmask — control system log

SYNOPSIS

```
#include <syslog.h>

openlog(ident, logopt, facility)
char *ident;

syslog(priority, message, parameters ... )
char *message;

closelog()

setlogmask(maskpri)
```

DESCRIPTION

syslog arranges to write *message* onto the system log maintained by syslogd(1M). The message is tagged with *priority*. The message looks like a printf(3) string except that %m is replaced by the current error message (collected from errno). A trailing newline is added if needed. This message will be read by syslogd(1M) and written to the system console, log files, or forwarded to syslogd on another host as appropriate.

Priorities are encoded as a *facility* and a *level*. The facility describes the part of the system generating the message. The level is selected from an ordered list:

LOG_EMERG	A panic condition. This is normally broadcast to all users.
LOG_ALERT	A condition that should be corrected immediately, such as a corrupted system database.
LOG_CRIT	Critical conditions, e.g., hard device errors.
LOG_ERR	Errors.
LOG_WARNING	Warning messages.
LOG_NOTICE	Conditions that are not error conditions, but should possibly be handled specially.
LOG_INFO	Informational messages.
LOG_DEBUG	Messages that contain information normally of use only when debugging a program.

If `syslog` cannot pass the message to `syslogd`, it will attempt to write the message on `/dev/console` if the `LOG_CONS` option is set (see below).

If special processing is needed, `openlog` can be called to initialize the log file. The parameter *ident* is a string that is prepended to every message. *logopt* is a bit field indicating logging options. Current values for *logopt* are:

`LOG_PID`	log the process id with each message: useful for identifying instantiations of daemons.
`LOG_CONS`	Force writing messages to the console if unable to send it to `syslogd`. This option is safe to use in daemon processes that have no controlling terminal since `syslog` will fork before opening the console.
`LOG_NDELAY`	Open the connection to `syslogd` immediately. Normally the open is delayed until the first message is logged. Useful for programs that need to manage the order in which file descriptors are allocated.
`LOG_NOWAIT`	Don't wait for children forked to log messages on the console. This option should be used by processes that enable notification of child termination via `SIGCHLD`, as `syslog` may otherwise block waiting for a child whose exit status has already been collected.

The *facility* parameter encodes a default facility to be assigned to all messages that do not have an explicit facility encoded:

`LOG_KERN`	Messages generated by the kernel. These cannot be generated by any user processes.
`LOG_USER`	Messages generated by random user processes. This is the default facility identifier if none is specified.
`LOG_MAIL`	The mail system.
`LOG_DAEMON`	System daemons, such as `ftpd`(1M), `routed`(1M), etc.

LOG_AUTH	The authorization system: login(1), su(1), getty(1M), etc.
LOG_LPR	The line printer spooling system: lpr(1), lpc(1M), lpd(1M), etc.
LOG_LOCAL0	Reserved for local use. Similarly for LOG_LOCAL1 through LOG_LOCAL7.

closelog can be used to close the log file.

setlogmask sets the log priority mask to *maskpri* and returns the previous mask. Calls to syslog with a priority not set in *maskpri* are rejected. The mask for an individual priority *pri* is calculated by the macro LOG_MASK(*pri*); the mask for all priorities up to and including *toppri* is given by the macro LOG_UPTO(*toppri*). The default allows all priorities to be logged.

EXAMPLES

```
syslog(LOG_ALERT, "who: internal error 23");

openlog("ftpd", LOG_PID, LOG_DAEMON);
setlogmask(LOG_UPTO(LOG_ERR));
syslog(LOG_INFO, "Connection from host %d",
                                CallingHost);

syslog(LOG_INFO|LOG_LOCAL2,"foobar error: %m");
```

SEE ALSO

syslogd(1M)

NAME

system — issue a shell command from Fortran

SYNOPSIS

```
character *N c
call system(c)
```

DESCRIPTION

system causes its character argument to be given to sh(1) as input, as if the string had been typed at a terminal. The current process waits until the shell has completed.

SEE ALSO

sh(1), exec(2), system(3S).

See acctsh(1M)

NAME

adduser — add a user account

SYNOPSIS

adduser [-r *real-name*] [-a *address*] [-x *extension*]
[-p *home-phone*] [-g *group*] [-s *shell*] [-d *dir*] [-h *home*]
[-u *lowest*] [-U *uid*] [-i] [-c] [*login-name*]...

DESCRIPTION

adduser creates an account for each *login-name*. One or more accounts may be added with a single command; command-line options apply to all names given. For each user, a password file entry is generated, and a home directory is created. If no *login-names* are provided, adduser enters interactive mode, prompting for all values that were not specified on the command line.

adduser uses the information given, or appropriate defaults, to generate an entry suitable for inclusion in /etc/passwd. If appropriate, an entry is also generated for /etc/group. For each account created, a brief report is written to the standard output. In the interactive mode, a confirmation is requested before the final changes to /etc/passwd are made.

A new home directory is created, if necessary, and startup files (.cshrc, .kshrc, .login, .logout, and .profile) are copied in to it from /skel. Directory and file permissions are set to read, write, and execute for owner, and read and execute for group (750). The information used to create each account is stored in a README file in each new home directory.

adduser does not permit new users to be added locally to a system that is receiving its password file via the Yellow Pages (YP). Accounts for users already in the YP password database can be added locally, but login name, group ID, and password fields are those given by the YP database, rather than those specified in the adduser command line.

In interactive mode, adduser prompts for a password for each new account. In batch mode, the password field is set to , . . , which causes a password to be set when the account is first used.

FLAG OPTIONS

Command line options allow administrators to override default values. The following flag options are available:

-r *real-name* Specify the real name of a person, for example, "Fred Smith" to be associated with the account.

To preserve embedded spaces, such as the space between the first and last name, place quotation marks around *real-name*. Quotes should be used to protect any blanks in the name.

–a *address* Specify an office address, for example, mail stop or building number.

–x *extension* Specify an office telephone number. For example, 3120. If you wish to surround the extension with parentheses, the entire string must be quoted, and a space must follow the closing parenthesis, as in "(3120) ".

–p *home-phone* Specify a home telephone number. For example, 408-555-1212. If you wish to surround a portion of the telephone number with parentheses, the entire phone number must be quoted and the closing parenthesis must be followed by a space, as in "(408) 555-1212".

–g *group* Specify the initial login group in which each user is to be placed. If omitted, `adduser` creates a unique group for each user. The group name created is of the form `gp`*gid* where *gid* is the next available numeric group ID.

–s *shell* Specify the full pathname of an executable program to use as the shell for each user added. If omitted, the default is `/bin/csh`. Other common choices are `/bin/ksh`, and `/bin/sh`.

–d *dir* Specify the full pathname of the parent of the user's home directory. By default, home directories are created as `/users/`*login-name*. This option causes a directory other than `/users` to be used. The name of the home directory is that of the new account, *login-name*. This option may not be used with the `-h` option.

–h *home* Specify the full pathname of the desired home directory. The *login-name* is not to be considered, and *home* is used as the name of the home directory. This option may not be used with the `-d` option.

-u *lowest*	Specify the desired lower bound for determining a numeric user ID (UID). If omitted, `adduser` uses the first available UID ≥ 200. The *lowest* UID may only be specified from the command line; interactive mode does not prompt for this value. This option may not be used with `-U`.
-U *uid*	Force the numeric UID to be *uid*. The UID may only be forced from the command line; interactive mode does not prompt for this value. This option may not be used with `-u`.
-i	Force an interactive mode, which is normally entered only if *login-name* is omitted. This option forces `adduser` to prompt for a real name, address, extension, home phone, group, shell, or home directory, which was not supplied on the command line.
-c	Create a USEFUL COMMAND folder in the home directory of the user.

FILES

`/etc/gtmp`	Temporary group file
`/etc/ptmp`	Temporary password file
`/etc/ogroup`	Old group file
`/etc/opasswd`	Old password file
`/usr/lib/skel/*`	Standard startup files (`.cshrc, .login, .profile,...`)
`$HOME/README`	Account-information file placed in each new account

SEE ALSO

`csh`(1), `ksh`(1), `sh`(1), `vipw`(1M).

A/UX Network System Administration.

BUGS

Telephone number checking is too restrictive. You should be able to use parentheses in the extension and phone number without having to follow the right parentheses with a space.

NAME

exterr — turn on/off the reporting of extended errors

SYNOPSIS

exterr /dev/*devicename* [*choice*]

DESCRIPTION

exterr turns on (or off) the reporting of extended errors on the specified device.

choice may be y or n (for "'yes" or "no") to turn error reporting on or off, respectively.

If reporting of errors is turned "off" with the argument n, only fatal errors are reported.

The default *choice* is "yes" (y), in which case soft as well as hard errors are reported on the specified device. The precise determination of what error messages are printed is specific to each device driver. In general, though, exterr may be used to reduce the amount of error information displayed. The devicename must be the "raw" one to access the ioctl.

FILES

/bin/exterr

EXAMPLE

```
exterr /dev/xxxx n
```

turns to off the reporting of extended errors for device /dev/*xxxx*.

NAME

ff — list file names and statistics for a System V file system

SYNOPSIS

/etc/ff [-a*n*] [-c*n*] [-i*inode-list*] [-I] [-l] [-m*n*] [-n*file*] [-p*prefix*] [-s] [-u] *device-file*

DESCRIPTION

If the file system specified by *device-file* is a System V file system, ff reads the file system's ilist and directories and saves inode data for files matching the selection criteria. ff outputs the path name for each saved inode, and any other file information you requested with the options described below. Output fields are positional. The output is sorted by inode, with the fields separated by tabs. ff's default output line is:

path-name inumber

If you enable all the options, the output fields are:

path-name inumber size uid

In the following list, *n* is a decimal integer (optionally signed), where +*n* means more than *n*, -*n* means less than *n*, and *n* means exactly *n*. A day is a 24 hour period.

-a n — select if the inode has been accessed in *n* days.

-c *n* — select if the inode has been changed in *n* days.

-i *inode-list* — generate names for only those inodes specified in *inode-list*. (An *inode-list* is a comma-separated list of inode numbers).

-I — do not print the inode number after each path name.

-l — list path names for multiply linked files.

-m *n* — select if the file has been modified in *n* days.

-n *file* — select if the inode has been modified more recently than the argument *file*.

-p *prefix* — add *prefix* to each generated path name. . is the default.

-s — print the file size, in bytes, after each path name.

-u print the owner's login name after each path name.

EXAMPLE

```
ff -I /dev/dsk/c5d0s0
```

generates a list of file names on the specified file system.

```
ff -m -l /dev/dsk/c5d0s0 > /log/incbackup/usr/tuesday
```

produces an index of files and inumbers on the file system which have been modified in the last 24 hours.

```
ff -i 451,76 /dev/rdsk/c5d0s0
```

obtains the path names for inodes 451 and 76 on the file system.

FILES

/etc/ff

SEE ALSO

find(1), finc(1M), frec(1M), ncheck(1M).

BUGS

Generates only a single path name for a multiply linked inode, unless you specify the -l option. When you specify -l, no selection criteria apply to the names generated. It includes all possible names for every linked file on the file system in the output.

On very large file systems, memory may run out before ff does.

NAME

finc — fast incremental backup for System V file systems

SYNOPSIS

finc [-a *n*] [-c *n*] [-m *n*] [-n *file*] *disk-device-file device-file*

DESCRIPTION

For System V file systems only, finc selectively copies the input *disk-device-file* to the output *device-file*. Mount the input *device-file* read-only to ensure an accurate backup, although you can obtain acceptable results in read-write mode. The tape must be previously labeled by labelit (see volcopy(1M)).

We recommend using the ff command to produce an index of the tape's contents before using finc. You can recover files on a finc tape with the frec command.

The argument *n* in the following options is a decimal integer (optionally signed), where +*n* means more than *n*, -*n* means less than *n*, and *n* means exactly *n*. A day is defined as 24 hours.

-a *n* true if the file has been accessed in *n* days.

-c *n* True if the inode has been changed in *n* days.

-m *n* true if the file has been modified in *n* days.

-n *file* true for any file which has been modified more recently than the argument *file*.

EXAMPLE

```
finc -m -2 /dev/rdsk/c5d0s0 /dev/rmt/tc1
```

writes a tape of all files from the specified file-system that have been modified in the last 2 days.

FILES

/bin/finc

SEE ALSO

cpio(1), ff(1M), frec(1M), volcopy(1M).

NAME

frec — recover files from a backup tape

SYNOPSIS

/etc/frec [-p*path*] [-f*reqfile*] *device-file inumber*:*name* ...

DESCRIPTION

frec recovers the files identified by *inumber* from the specified *device-file*. This is a backup tape written by volcopy(1M) or finc(1M). The data for each recovery request is written into the *name* file.

If any directories are missing in the paths of recovery *names* they will be created.

-f*reqfile* specify a file containing recovery requests. The format is

inumber:*name*

with one request per line.

-p*path* specify a *path* for names not beginning with / or ./. The default pathname is your current working directory.

EXAMPLE

```
frec /dev/rmt/tc1 1216:junk
```

recovers a file, with inumber 1216, into the file junk in your current working directory.

```
frec -p /u/work /dev/rmt/tc1 14156:a 1232:b
3141:/u/joe/a.c
```

recovers files with inumbers 14156, 1232, and 3141 into files:

```
/u/work/a
/u/work/b
/u/joe/a.c
```

FILES

/etc/frec

SEE ALSO

cpio(1), ff(1M), finc(1M), volcopy(1M).

BUGS

While creating the intermediate directories contained in a pathname, frec can only recover inode fields for those directories contained on the tape and requested for recovery.

NAME

fsck — check file-system consistency and interactively repair

SYNOPSIS

/etc/fsck -T 5.2 [-y] [-n] [-m*timeout*] [-s*X*] [-S*X*] [-t*file*] [-q] [-D*options...*] [-f] [-p*passtostart*] [*svfs-filesystem...*]

/etc/fsck [-b*block-number*] [-y] [-n] [-m*timeout*] -T 4.2 [-p*passtostart*] [*ufs-filesystem...*]

DESCRIPTION

fsck audits and interactively repairs inconsistent conditions for A/UX® file systems. If *filesystem* is not specified, fsck reads a list of default file systems from the file /etc/fstab. If the file system is consistent, then only the number of files, number of blocks used, and number of blocks free are reported. If the file system is inconsistent, the operator is prompted for confirmation to proceed before each corrective action is attempted. Frequently corrective actions result in some loss of data. The amount and severity of data lost may be determined by examining various parameters such as non-zero filesize. Typically, each consistency correction is gated by the operator's yes or no response. However, if the operator does not have write permission for *filesystem*, fsck merely indicates corrective actions needed.

fsck has more consistency checks than its predecessors check, dcheck, fcheck, and icheck combined.

Checking the raw device is almost always faster and should be used with everything but the root file system. In addition, any file system other than the root file system should be unmounted at the time that it is checked with fsck. (It is not possible to unmount the root file system.)

FLAG OPTIONS

The flag options for the fsck command differ depending on the type of file system.

The following flag options apply to both System V file systems (SVFS) and Berkeley 4.2 file systems (UFS):

-T*file-system-type*
: Indicate the file-system type, for example, 4.2 for a Berkeley 4.2 file system (UFS) or 5.2 for a SVFS file system. If this option is not used, fsck attempts to determine the type.

-y Assume a yes response to all questions asked by `fsck`.

-n Assume a no response to all questions asked by `fsck`. This flag option does not open the file system for writing.

-m*timeout*

Use a Macintosh® user interface. This causes the `StartMonitor` to move the progress bar forward periodically during the boot sequence. In addition, if `fsck` finds a problem with a file system, it calls `/etc/macquery` to post a Macintosh alert box asking the user if he or she would like to repair the file system. If the user clicks the default Repair button in the alert box, `fsck` assumes a yes response to all further questions regarding that file system. If the user clicks Don't Repair, `fsck` assumes a no response to all further questions regarding that file system. If a timeout value greater than 0 is given, the dialog automatically selects the default button after that number of seconds. If *timeout* is not given, the default is 0, indicating that the alert should not automatically time out.

-p*passtostart*

Similar to the `-q` option, but mid-progress phase messages are also suppressed. Besides quietly fixing (''preening'') certain file-system inconsistencies, the `-p` flag option and *passtostart* number provide another way to specify which file systems to check. *passtostart* specifies a threshold value that triggers the checking of a file system depending on its *passno* field in `/etc/fstab` (see `fstab`(4)). The default *passtostart* number is 1. If the value of the pass number is 2, as it normally is for the `fsck` in `/etc/bcheckrc`, only those file systems in `/etc/fstab` with pass numbers of 2 or greater are checked. Only partitions in `fstab` that are mounted `rw` or `ro` are subject to being checked this way. If the superblock state field indicates that the file system was properly unmounted, it is skipped.

The following options are interpreted by `fsck` for a SVFS file system:

-s*X*

Ignore the actual free list and (unconditionally) reconstruct a new one by rewriting the superblock of the file system. The

file system should be unmounted during this operation, and if this is not possible, care should be taken that the system is quiescent and that it is rebooted immediately afterward. This precaution is necessary so that the obsolete main memory copy of the superblock does not continue to be used, or to be written onto the file system.

The `-s`*X* flag option allows for creating an optimal free-list organization. The argument *X* should be in the format *blocks-per-cylinder*`:`*blocks-to-skip*. If *X* is not given, the values used when the file system was created are used. If these values were not specified, then the value `400:7` is used.

`-s`*X*

Conditionally reconstruct the free list. This flag option is like `-s`*X* except that the free list is rebuilt only if no discrepancies were discovered in the file system. Using `-S` forces a no response to all questions asked by `fsck`. This flag option is useful for forcing free-list reorganization on uncontaminated file systems.

`-t`*file*

Use a scratch file if `fsck` cannot obtain enough memory to maintain its tables. If the `-t` option is specified, *file* is used, if needed, as the scratch file. Without the `-t` flag, `fsck` prompts the operator for the name of a scratch file. The file, when chosen, should not be on the file system being checked, and if it is not a special file or did not already exist, it is removed when `fsck` completes.

`-q` Suppress progress messages as well as eliminate the need to confim certain corrective actions. Unreferenced `fifos` are silently removed. If `fsck` detects inconsistencies, counts in the superblock are automatically fixed, and the free list salvaged. Inconsistencies other than these still require confirmation.

`-D`*options*

Check directories for bad blocks. If *options* is empty, the directories are merely checked. The `B` option checks for and clears parity bits in filenames, `C` checks whether all trailing characters in the filename are null, and `CZ` checks and writes nulls to all trailing characters in the filename.

`-f` Use a fast check to check blocks and sizes (Phase 1) and the free list (Phase 5). The free list is reconstructed (Phase 6) if necessary.

The following flag option is unique to `fsck` for a Berkeley 4.2 (UFS) file system:

`-b`*block-number*
Use the block specified immediately after the flag as the superblock for the file system. Block 16 is always an alternate superblock.

Consistency Checks Performed

Inconsistencies checked are as follows:

1. Blocks claimed by more than one inode or the free list are checked.
2. Blocks claimed by an inode or the free list outside the range of the file system are checked.
3. Incorrect link counts are checked.
4. Size checks are performed to check for:
   incorrect number of blocks
   directory size not correctly aligned
5. Checks for bad inode format are performed.
6. Checks for blocks not accounted for anywhere are performed.
7. Directory checks are performed to find:
   files pointing to unallocated inodes
   inode numbers out of range
8. Superblock checks are performed for:
   more than the maximum number of inodes.
   more blocks for inodes than there are in the file system
9. Checks for a bad free-block list format are performed.
10. Checks for incorrect total free block or free inode count, or both, are performed.

Reconnection of dislocated files

Orphaned files and directories (allocated but unreferenced) are, with the operator's confirmation, reconnected, if not empty, by placing them in the `lost+found` directory. The user is notified if the file or directory is empty or not. If it is empty, `fsck` silently removes them and forces the reconnection of nonempty directories. The name assigned is the inode number. The only restriction is that the directory `lost+found` must preexist in the root of the file system being checked and must have empty slots in

which entries can be made. This is accomplished by making a `lost+found` directory, copying a number of files to the directory, and then removing the files before `fsck` is executed. See `mklost+found`(1M).

EXAMPLES

```
fsck /dev/rdsk0/c0d0s0
```

checks the consistency of the file system referred through `/dev/rdsk0/c0d0s0`.

FILES

`/etc/fsck`
`/etc/ufs/fsck`
`/etc/svfs/fsck`
`/etc/%fsck`

`/etc/fstab`	Contains default list of file systems to check

SEE ALSO

`clri`(1M), `fsirand`(1M), `mkfs`(1M), `ncheck`(1M), `newfs`(1M), `typefs`(1M), `fstab`(4), `fs`(4).

"Checking the A/UX File System: `fsck`," in *A/UX Local System Administration.*

DIAGNOSTICS

The diagnostics produced by `fsck` are intended to be self-explanatory.

BUGS

Inode numbers for `.` and `..` in each directory should be checked for validity.

NAME

fsdb — debug the file system

SYNOPSIS

/etc/fsdb [-T4.2] [-?] [-o] [-p*string*] [-w] *special*

/etc/fsdb [-T5.2] *special* [-]

DESCRIPTION

fsdb can be used to patch up a damaged file system after a crash. It has conversions to translate block and inumbers into their corresponding disk addresses. Also included are mnemonic offsets to access different parts of an inode. These greatly simplify the process of correcting control-block entries or descending the file-system tree.

Since fsdb reads the disk raw, it is able to circumvent normal file-system security. Extreme caution is advised in determining its availability on the system. Suggested permissions are 600 and owned by bin.

fsdb has different formats depending on the type of file system you are debugging. fsdb can be used for either a Berkeley 4.2 file system (UFS) or a System V file system (SVFS).

fsdb contains several error-checking routines to verify inode and block addresses. These routines can be disabled, if necessary, by invoking fsdb with the -o option for a UFS file system or by using the - option for a SVFS file system. The o command works for both file systems.

fsdb reads a block at a time and therefore works with raw as well as block I/O. A buffer management routine is used to retain commonly used blocks of data in order to reduce the number of read system calls. All assignment operations result in an immediate write-through of the corresponding block. Note that in order to modify any portion of the disk for a UFS file system, fsdb must be invoked with the -w option.

Wherever possible, syntax similar to adb syntax was adopted to promote the use of fsdb through familiarity.

fsdb considers numbers in UFS as hexadecimal by default and considers numbers in SVFS as decimal by default. However, the user has control over how data is to be displayed or accepted. The base command displays or sets the input/output base. Once set, all input defaults to this base, and all output is shown in this base. The base can be overriden temporarily for input by preceding hex-

NAME

fsstat — report file-system state

SYNOPSIS

fsstat [-T*file-system-type*] *file-system*

DESCRIPTION

fsstat reports the state of the specified *file-system*. If the system was brought down cleanly or if *file-system* was successfully repaired by fsck(1M), the state should be OK.

FLAG OPTIONS

The following flag option is interpreted by fsstat:

-T*file-system-type*
: Indicate the file-system type, such as 4.2 (UFS) or 5.2 (SVFS). If this option is not used, fsstat attempts to determine the type.

FILES

/etc/fsstat
/etc/fs/*/fsstat

SEE ALSO

fsck(1M), fstyp(3), fs(4).

NAME

ftpd — Internet File Transfer Protocol server

SYNOPSIS

/usr/etc/in.ftpd [-d] [-l] [-t*timeout*]

DESCRIPTION

ftpd is the DARPA Internet File Transfer Prototocol server process. The server uses the TCP protocol and listens at the port specified in the ftp service specification; see services(4N).

If the -d flag option is specified, debugging information is written to the standard output.

If the -l flag option is specified, each ftp session is logged on the standard error output.

The ftp server will timeout an inactive session after 15 minutes. If the -t flag option is specified, the inactivity timeout period will be set to *timeout*.

The ftp server currently supports the following ftp requests; case is not distinguished.

Request	Description
ABOR	abort previous command
ACCT	specify account (ignored)
ALLO	allocate storage (vacuously)
APPE	append to a file
CDUP	change to parent of current working directory
CWD	change working directory
DELE	delete a file
HELP	give help information
LIST	give list files in a directory ("ls -lg")
MKD	make a directory
MODE	specify data transfer *mode*
NLST	give name list of files in directory ("ls")
NOOP	do nothing
PASS	specify password
PASV	prepare for server-to-server transfer
PORT	specify data connection port
PWD	print the current working directory
QUIT	terminate session
RETR	retrieve a file
RMD	remove a directory

RNFR	specify rename-from file name
RNTO	specify rename-to file name
STOR	store a file
STOU	store a file with a unique name
STRU	specify data transfer *structure*
TYPE	specify data transfer *type*
USER	specify user name
XCUP	change to parent of current working directory
XCWD	change working directory
XMKD	make a directory
XPWD	print the current working directory
XRMD	remove a directory

The remaining `ftp` requests specified in Internet RFC 959 are recognized, but not implemented.

The `ftp` server will abort an active file transfer only when the ABOR command is preceded by a Telnet "Interrupt Process" (IP) signal and a Telnet "Synch" signal in the command Telnet stream, as described in Internet RFC 959.

`ftpd` interprets file names according to the "globbing" conventions used by `csh`(1). This allows users to utilize the metacharacters "`*?[]{}~`".

`ftpd` authenticates users according to three rules.

1) The user name must be in the password data base, `/etc/passwd`, and not have a null password. In this case, a password must be provided by the client before any file operations may be performed. `ftpd` also examines the shell field of the user's entry in `/etc/passwd`. If the shell field is `/bin/sh`, `/bin/csh`, or empty, the login is allowed. If the shell field contains the name of any other shell, such as `/bin/ksh`, the file `/etc/shells` must contain the specified name, or the authentification will fail.
2) The user name must not appear in the file `/etc/ftpusers`.
3) If the user name is "`anonymous`" or "`ftp`", an anonymous `ftp` account must be present in the password file (user "`ftp`"). In this case the user is allowed to log in by specifying any password (by convention this is given as the client host's name).

In the last case, ftpd takes special measures to restrict the client's access privileges. The server performs a chroot(2) command to the home directory of the "ftp" user. In order that system security is not breached, it is recommended that the "ftp" subtree be constructed with care; the following rules are recommended.

~ftp)
Make the home directory owned by "ftp" and unwritable by anyone.

~ftp/bin)
Make this directory owned by the superuser and unwritable by anyone. The program ls(1) must be present to support the list commands. This program should have mode 111.

~ftp/etc)
Make this directory owned by the superuser and unwritable by anyone. The files passwd(4) and group(4) must be present for the ls command to work properly. These files should be mode 444.

~ftp/pub)
Make this directory mode 777 and owned by "ftp". Users should then place files which are to be accessible via the anonymous account in this directory.

SEE ALSO

ftp(1N).

BUGS

The anonymous account is inherently dangerous and should be avoided when possible.

The server must run as the superuser to create sockets with privileged port numbers. It maintains an effective user ID of the logged in user, reverting to the superuser only when binding addresses to sockets. The possible security holes have been extensively scrutinized, but are possibly incomplete.

NAME

kconfig — tune kernel parameters for work-load optimization

SYNOPSIS

/etc/kconfig [-a [-v] [-V]] [-n*namelist*]

DESCRIPTION

kconfig manipulates an A/UX® kernel code file for changing system parameters.

> *Note:* It is not recommended that this utility be used unless you know exactly what you are doing. Incorrect use can cause system failures.

kconfig can be used to either list or change the system parameters listed later in the section "System Parameters."

Note that kconfig does not change parameters of the running kernel, just the image on the disk. You must then run shutdown(1M) and reboot(1M) for the changes to be effective.

FLAG OPTIONS

kconfig interprets the following flag options:

-a	List the current values of the parameters in the kernel object file *namelist*.
-v	Use with the -a flag option to produce verbose (commented) output.
-V	Use with the -a flag option to print the current version of the kernel object file *namelist*.
-n *namelist*	Specify which kernel object file is being modified (the default is /unix).

If -a is used, kconfig displays the parameters and exits. If -a is not used, standard input is read for a list of changes. You can specify one change per input line of the form

PARAM = *value*
PARAM += *value*
PARAM -= *value*

where *PARAM* is one of the parameter names (listed later in the section "System Parameters") and the *value* is either a decimal constant or a hexadecimal constant preceded by 0x.

If the = form is used, the parameter is given the value specified. If the -= form is used, the parameter is decreased by the value given. If the += form is used, the parameter is increased by the value. If the value is not within a system-defined maximum and minimum range for the parameter, an error occurs and the kernel is not changed.

SYSTEM PARAMETERS

The following system parameters are recognized by `kconfig`:

Note: Not all of these parameters will necessarily be supported in future releases of the operating system.

`NBUF` Specify the number of disk I/O buffers to allocate. These form a data cache for information read or written to file systems. Each buffer consists of `SBUFSIZE` data areas and about three dozen bytes of header information. Increasing the number of buffers improves the "cache hit ratio" on the buffer pool, but at the expense of available memory for processes. The number of system buffers normally ranges from 100 to 1500. One hundred buffers should be used for systems with 2 megabytes of total memory. Probably not more than 750 buffers should be used on systems with 4 MB of memory. Increasing the number of buffers reduces the memory available for applications and may cause more paging to occur. Systems with a single user might conceivably use more buffers than systems that typically have numerous memory-intensive applications running. The default value of `NBUF` is 0. When the default value is used, the system dynamically calculates the number of buffers as 10 percent of free memory at boot time.

`SBUFSIZE` Determine the size of system I/O logical block size on disk devices. The size is configurable. Most System V file systems (SVFS) use a 1 kilobyte (KB) block size. The `SBUFSIZE` parameter determines the size of in-core buffers allocated for the

buffer cache. The number must be an even multiple of 512 bytes and should be large enough to accept the largest block size of all active file systems. When the system buffer size is increased, the total number of buffers should be decreased, assuming the same amount of memory is used for the buffer cache.

NPBUF — Specify how many physical input/output buffer headers to allocate. One header is needed for each simultaneous read or write of a "character-special" disk or tape device, or for each concurrent swap I/O. The default value is 20.

NFILE — Determines the size of the system file-table pool. Each entry represents an open file in use by some process. When no space is available in the file table, the message "`file: table is full`" is printed on the system console. The size is generally between 100 (the default) and 400. `NFILE` is often equal to the `NINODE` parameter.

NINODE — Set the size of the system inode table. Each table entry represents an in-core inode being used for an open file, an open working directory, or a mount point. For systems using a network file system (NFS) to access remote file systems, only locally open files consume inodes. Normally the `NINODE` parameter is greater than or equal to the `NFILE` parameter. Generally the range is from 100 (the default) to 400 inodes. When all inodes in the system inode table are used, the message "`inode: table is full`" is printed.

NSPTMAP — Allocate table entries that are used to map the system page table entries. The default value is 75. If the message "`sptreserve: No kernel virtual space`" is printed, the system has exhausted the map.

NCALL
: Specify the size of the timeout table. Each entry may be used by device-driver software to arrange for a function to be called at a later time. The default value is 50. If many add-on drives, such as for NuBus™ peripherals, are added to the system, this value might require an increase. If the timeout table is exhausted, the message "`timeout table overflow`" is printed, and the system halts execution.

NMOUNT
: Specify the size of the SVFS mount table. The parameter does not affect the number of remotely mounted network file systems allowed. If numerous disk devices containing many file-system partitions are present, the mount table may need to be increased. The default size is 10.

NFLOCK
: Specify the number of system-wide `locking`(2) file locks. Each area of a locked file requires one of these table entries. If the table is exhausted, the error EDEADLOCK is returned to the application that made the lock request. The default number of locks is 200.

NREGION
: Defines the number of memory regions available to all processes in the system. A typical process has a memory region for data, a memory region for stack, and a memory region for program text, but this region would be shared between all processes executing the application. If the region table is exhausted, the message "`Region table overflow`" is printed on the system console. The default value is 200.

NPROC
: Specify the total number of processes in the system. In general, each executing command, application, or system daemon is a process. Each user of the system, or each active layer or window generally uses between 2 and 8 simultaneous processes. When no processes are available, the mes-

sage "proc: table is full" is printed at the system console. By default, NPROC is 50.

NCLIST — Specify the number of system command lists (CLIST). A CLIST is a memory area used by driver software for terminals, built-in modems, or serial printer connections. Five to 10 CLISTs are required by each active terminal. When no CLISTs are available, incoming characters are lost. Each CLIST requires 64 bytes of data, and a 12-byte header. The console and some optional serial cards use the streams interface and do not require CLISTs. The default number of CLISTs is 200.

MAXUP — Determines the maximum number of concurrent processes for each user ID. The superuser is exempt from this restriction. This limit is based on user ID, not on the login terminal. If ten people are logged in using a single user ID, the limit could be reached quickly. Normally NPROC is at least 10 percent larger than MAXUP. The default value of MAXUP is currently 25.

VHNDFRAC — The virtual memory system depends on the activity of the paging daemon vhand to free memory by paging unused memory to the swap-disk device. The algorithm writes out to disk pages that have not been used for some time. If the system is not active, there is plenty of free memory and no work for vhand. VHNDFRAC and other tuning parameters allow the adventurous system administrator to fine-tune the performance.

The vhand fraction is used to determine the initial value of the system variable VHANDL. If free memory falls below VHANDL, the paging daemon, vhand, is awakened to begin aging and monitoring the resident set of virtual memory pages. At system startup time, VHANDL is set to

```
vhandl=MAX(maxmem/VHANDFRAC,GETPGSHI)
```

where maxmem is the available free memory at startup time and GETPGSHI is the free-memory high-water mark, described later. Normally VHANDFRAC is set to 16.

MAXPMEM — Specify the maximum physical memory to use. If this is set at 0, the system uses all available physical memory. It is recommended this value be 0, unless you are testing alternate memory configurations.

NMBUFS — Allocate a number of buffers for networking. Each buffer requires 256 bytes, of which 240 are available for data. As few as 100 mbufs may be used for basic networking. When NFS is used on a system, the number should be increased. As a guide, each NFS daemon may transfer 8 KB of data. Allocating (8192**n_daemons*)/240 provides a starting point in the calculation. The command `netstat -m` may be used to determine the number of mbufs in use. If the message "`m_expand returning 0`" is seen frequently or if the system halts after displaying the message "`panic: out of mbufs`", the number of mbufs should be increased. By default, 500 mbufs are allocated.

NPTY — Determines the total number of possible pseudo tty devices (that is, /dev/pty*). This default number, which is also the maximum potential number of devices, is 16. Special files must still be created in the /dev directory for ptys to be used. If more than the allocated number of ptys are created, the error code "`No such device or address`" is returned by the open(2) of any of the unallocated device files.

MAXCORE — Set the space available for use by the memory allocation of the kernel from its internal heap. Most data structures used to

access remote files via NFS are allocated from this pool, as is space used by generic disk devices and the system's name-lookup code. If the message "panic: kmem_alloc" appears and the system halts, the value of MAXCORE should be increased. The default value is 128 KB.

MAXHEADER — Limit the number of allocations possible from the kernel's internal heap. If the message "panic: getfreehdr" is seen, this allocation should be increased. The default value MAXHEADER is 2048.

NSTREAM — Determine the maximum number of stream heads possible in the system. The console, AppleTalk®, and the shell layering by the console are examples of streams. The number of streams required might range from 8 to 48. The default number of stream allocated is 32.

NQUEUE — Define the maximum number of stream queues. Each stream head, driver, and module pushed onto a stream creates two queues. Typically this parameter is set to 8 times the number of stream heads. The default number of queues is 256.

The allocation of stream blocks determines the availability of buffer space used by stream devices. The optimal allocation depends on the types of devices present in the system. It is expected that the installation scripts for devices using the streams mechanism will include the necessary commands to increase the number of blocks allocated. If too few blocks of the size required by a driver are present, the system may lose input characters.

NBLK4096 — Allocate a number of 4 KB stream blocks. The default is 0.

NBLK2048 — Allocate a number of 2 KB stream blocks. The default is 20.

NBLK1024 — Allocate a number of 1 KB stream blocks. The default is 12.

NBLK512	Allocate a number of 512-byte stream blocks. The default is 8.
NBLK256	Allocate a number of 256-byte stream blocks. The default is 16.
NBLK128	Allocate a number of 128-byte stream blocks. The default is 64.
NBLK64	Allocate a number of 64-byte stream blocks. The default is 256.
NBLK16	Allocate a number of 16-byte stream blocks. The default is 128.
NBLK4	Allocate a number of 4-byte stream blocks. The default is 512. Keyboard and tty input uses this resource.
SLICE	Specify the maximum time slice available to a process before it is considered for rescheduling. At the end of the time slice, the active process is suspended, and the system searches for a process with a higher CPU priority. If no higher-priority process exists, the previous process is given another slice. The default time slice is 60. The units are "ticks," and there are 60 ticks in a second.
GETPGSLOW	Specifies the get-pages low limit which is the free-memory low-water mark for the vhand daemon. When this number is reached, vhand becomes active and starts stealing pages from active processes. The default value is 20. Increase the value to make the daemon more active. The value must be greater than 0 and less than GETPGSHI.
GETPGSHI	Specify the get-pages high limit which is the free-memory high-water mark for the vhand daemon. When this number is exceeded, the system stops stealing pages from active processes. The default value is set to 30. GETPGSHI should be greater than GETPGSLOW and less than about one-fourth of the total available memory.

GETPGSMSK — Specify, when used by vhand which pages to steal. The default value of GETPGSMSK (the get-pages mask) is 0x408. It may be modified, but most changes are more educational than useful. Values in this parameter correspond to bits in the page-table entries. Each masked bit in the page-table entry must be 0 in order for the page to be taken by vhand. The current setting is

```
PG_NDREF | PG_REF
```

(defined in /usr/include/sys/page.h). By including the modified bit (PG_M), vhand would not steal any pages with the modified bit on.

VHANDR — Specify in seconds the maximum rate at which vhand may run if free memory is less than VHANDL, as explained earlier for VHNDFRAC. The default value is 5. Increase the value to make the daemon less active. The value should be between 1 and 10.

MAXSC — Specify the maximum number of pages that may be written to the swap device in a single operation. The default value is 64. Increasing this number increases the I/O overhead spent in swapping, but decreasing the value may reduce the amount of free memory available when a page fault occurs.

MAXFC — Specify the maximum number of pages that may be placed on the free list at one time. The default value is 100. Increasing the number may allow for faster handling of page faults when a process needs more memory, but it may also reduce the working set of applications so that page faults occur more frequently.

MAXUMEM — Determine the maximum user virtual address space in pages. This number may range from about 30 to 0x20000. The default value is 0x40000, which eliminates all checking. Small values of MAXUMEM may make software that is normally taken for granted unusable.

FLCKREC — Specify the number of `flock`(2) lock structures. When this size is exceeded, the error ENOSPC is returned to the requesting program. The default value is 200.

FLCKFIL — Specify the number of `flock` inode structures. When this size is exceeded, the error EMFILE is returned to the requesting program. The default value is 50.

CDLIMIT — Set the process `ulimit` on file size. Only this number of 512-byte blocks may be written to any file by any process owned by any user. The default value is 16 million blocks (0x1000000).

CMASK — Determine the system wide default file-creation mask. Generally, the value of CMASK is overridden by the `umask` directive of one's chosen shell. The default value is 0.

ROOTDEV — Determine the disk device containing the root file system. This must be a device number as used internally by the kernel. The device major number is in the upper byte, and the minor number is in the lower byte. If the value is 0xffff, then the value passed from the A/UX StartupShell (see `StartupShell`(8)) booter is used.

SWAPDEV — Specify the swap disk device. The specification is the same format as ROOTDEV. If set to 0xffff, the value passed from the A/UX StartupShell booter is used.

PIPEDEV — Determine the disk device for temporary pipe-file space. The specification is the same format as ROOTDEV. If set to 0xffff, the value passed from the A/UX StartupShell booter is used.

DUMPDEV — Currently unused by the system.

SWAPLO — Specify the starting disk address of the swap area to determine the number of 512-byte blocks to skip at the beginning of the swap partition. This would be done if these blocks were to be used for some other purpose; however, this result may also be achieved by repartitioning the

disk. The default value is 0.

SWAPCNT — Specify the size of the swap area, which is a number of 512-byte blocks. The system warns you if it is running short of swap space. If SWAPCNT is 0, the size of the swap area is set to the size of the swap partition of the root device. To adjust swap space, both kconfig and the dp(1M) (disk partitioning) utility must be used. The default value is 0.

MINARMEM — Set the minimum number of pages of physical ("resident") memory reserved for user text and data segments in order to avoid deadlock. The default value is 10, and values might range from 10 to 40.

MINASMEM — Define the minimum number of pages of system (swap + resident) memory reserved for system purposes and therefore unavailable for the text and data segments of user processes. The default value is 10. Normally MINASMEM is greater than MINARMEM.

FILES

/etc/kconfig
/unix

SEE ALSO

dp(1M), reboot(1M), shutdown(1M), swap(1M), flock(2), locking(2), open(2), pty(7), StartupShell(8).

Building A/UX Device Drivers, A/UX Local System Administration, A/UX Network System Administration, A/UX Network Applications Programming.

NAME

keyset — set console keyboard mapping

SYNOPSIS

/etc/keyset [-k *keyboard*] [-c *country*]

DESCRIPTION

keyset sets the current console keyboard mapping to the type of keyboard given by *keyboard* and *country*. Currently, three different keyboard types are supported by the console driver. They are ADB extended, ADB ISO, and ADB standard. They are given to keyset as one of the following arguments: adbext, adbiso, or adbstd. If this parameter is not specified, keyset reads the file /dev/kmem to find out what type of keyboard was in use when the system was booted.

The different *country* types currently supported are Australia, Britian, Denmark, Dutch, Finland, Flemish, FrCanada, France, Germany, Greece1, Greece2, Iceland, Italy, Norway, Spain, Sweden, SwissFrench, SwissGerman, Turkey, USA, and Yugoslavia. If this parameter is not specified, keyset reads the default information stored in the Macintosh system file /mac/sys/System Folder/System for the *country* parameter.

The Macintosh environment must be shut down prior to running keyset.

EXAMPLES

To set the keyboard to ADB standard in the United States, use the command: /etc/keyset -k adbstd -c USA

FILES

/mac/sys/System Folder/System

DIAGNOSTICS

The exit status is 0 if everything went OK. A status of 1 indicates a usage error or an error while setting the new keyboard map.

`slots`
Enable calling the Slot Library.

`toolbox`
A/UX user interface device driver.

EXAMPLES

To prepare a kernel for TCP/IP services with NFS and AppleTalk networking services, type

```
/etc/newconfig nfs appletalk
```

To prepare a kernel that supports basic TCP/IP services, AppleTalk networking, the Macintosh sound chip, and Apple Tape Backup 40SC device, type

```
/etc/newconfig bnet appletalk snd tc
```

To prepare a kernel from which all networking services are removed, type

```
/etc/newconfig nonet noappletalk
```

FILES

`/etc/newunix`	A shell script called by `newconfig`
`/etc/autoconfig`	An executable utility called by `newconfig`
`/etc/startup`	A shell script created by `autoconfig`

SEE ALSO

`newunix`(1M), `autoconfig`(1M).

NAME

newfs — construct a new UFS file system

SYNOPSIS

/etc/newfs [-v] [*options*] *device-file type*

DESCRIPTION

newfs constructs a Berkeley 4.2 file system (UFS) on *device-file*, which is the device-file on which the new file system is to be created. The argument *type* indicates the disk type; this *type* is used to find the appropriate disk name entry in /etc/disktab. The newfs command consults the disk label for disk partition information and /etc/disktab for disk architecture information, calculates the appropriate parameters to use in calling mkfs, and then builds the file system by invoking mkfs.

If the -v option is supplied, newfs prints out its actions, including the parameters passed to mkfs.

newfs uses fsirand as a security precaution.

Options that may be used to override default parameters passed to mkfs are:

-s *size*
: Specify the size of the file system in sectors. If this option is not present, the size information from the disk partition map will be used. See dpme(4).

-b *block-size*
: Specify the block size of the file system in bytes. The default value is 4096.

-f *frag-size*
: Specify the fragment size of the file system in bytes. The default value is 1024.

-t *tracks-per-cylinder*
: Specify the number of tracks per cylinder, which is equivalent to the number of heads on the disk drive. If this option is not present, the information from /etc/disktab is used.

-c *cylinders-per-group*
: Specify the number of cylinders per cylinder group in a file system. The default value is 16.

-m *free-space*
: Specify the percentage of space reserved from use by normal

users. This value is known as the free-space threshold for the file system. The default value is 10%. This value can be changed later using `tunefs`(1M).

`-r` *revolutions-per-minute*
Specify the speed of the disk in revolutions per minute (usually 3600).

`-i` *number of bytes per inode*
Specify the density of inodes in the file system. The default is to create an inode for each 2048 bytes of data space. If few inodes are desired, a larger *bytes-per-inode* should be specified. If many inodes are desired, a smaller *bytes-per-inode* should be specified.

FILES

`/etc/newfs`	
`/etc/fs/ufs/mkfs`	Actually builds the file system.

SEE ALSO

`dp`(1M), `fsck`(1M), `fsirand`(1M), `tunefs`(1M), `dpme`(4), `disktab`(4), `ufs`(4), `gd`(7).

NAME

newunix — prepare for new kernel configuration

SYNOPSIS

/etc/newunix [[no]*module*] ...

DESCRIPTION

newunix is typically called by newconfig, but can also be used directly. However called, it begins the process of configuring a new kernel by installing (or removing) the appropriate scripts and driver object files needed by autoconfig. When you invoke it directly, you should run autoconfig afterwards to complete the kernel-configuration process. When you invoke it indirectly by running newconfig, autoconfig is called automatically, making newconfig easier to use.

The configuration of the new kernel is controlled by the arguments of *module*. Multiple invocations of newunix can be used to accumulate the new kernel configuration or to remove previously established modules through the argument format of no*module* (a couple of exceptions are noted in the following).

Possible values of *module* are:

appletalk	Provide AppleTalk® support.
bnet	Use basic networking. To turn off basic networking, use nonet rather than the expected nobnet.
nfs	Use the network file system (NFS). To turn off NFS support, use nonet rather than the expected nonfs.
slip	Provide support for the Serial Line/Internet Protocol (SL/IP).
tc	Provide support for the Apple® Tape Backup 40SC device.
toolbox	Use the A/UX® Toolbox.

EXAMPLES

To prepare the system for an NFS kernel, enter

```
/etc/newunix nfs
```

To prepare the system for a kernel that supports the tape controller, enter

```
/etc/newunix tc
```

See sadc(1M)

NAME

sadc, sa1, sa2 — system activity report package

SYNOPSIS

/usr/lib/sa/sadc [*t n*] [*file*]

/usr/lib/sa/sa1 [*t n*]

/usr/lib/sa/sa2 [-u] [-b] [-y] [-c] [-w] [-a] [-q] [-v] [-m] [-A] [-s*time*] [-e*time*] [-i*sec*]

DESCRIPTION

System activity data can be accessed at the special request of a user (see sar(1)) or automatically, on a routine basis as described here. The operating system contains a number of counters that are incremented as various system actions occur. These include counters for CPU utilization, buffer usage, disk and tape I/O activity, TTY device activity, switching and system-call, file-access, queue activity, and counters for interprocess communications.

sadc and the shell procedures sa1 and sa2 are used to sample, save, and process this data.

sadc, the data collector, samples system data *n* times every *t* seconds and writes in binary format to *file* or to standard output. If *t* and *n* are omitted, a special record is written. This facility is used at system boot time to mark the time at which the counters restart from zero. The /etc/rc entry

```
su adm -c "/usr/lib/sa/sadc /usr/adm/sa/da`date +%d`"
```

writes the special record to the daily data file to mark the system restart.

The shell script sa1, a variant of sadc, is used to collect and store data in binary file /usr/adm/sa/sa*dd* where *dd* is the current day. The arguments *t* and *n* cause records to be written *n* times at an interval of *t* seconds, or only once, if omitted. The entries in crontab (see cron(1M))

```
0 * * * 0,6 /usr/lib/sa/sa1
0 8-17 * * 1-5 /usr/lib/sa/sa1 1200 3
0 18-7 * * 1-5 /usr/lib/sa/sa1
```

will produce records every 20 minutes during working hours and hourly otherwise. The shell script sa2, a variant of sar(1), writes a daily report in the file /usr/adm/sa/sar*dd*. The flag options are explained in sar(1). The crontab entry

```
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 3600 -A
```

will compile a single report at 6:05 P.M. of each working day giving an hourly summary of all activity in the interval 8:00 A.M. to 6:01 P.M.

The structure of the binary daily data file is

```
struct sa {
    struct sysinfo si; /* defined in
                          /usr/include/sys/sysinfo.h   */
    int    szinode;    /* current size of inode table  */
    int    szfile;     /* current size of file table   */
    int    sztext;     /* current size of text table   */
    int    szproc;     /* current size of proc table   */
    int    mszinode;   /* maximum size of inode table  */
    int    mszfile;    /* maximum size of file table   */
    int    msztext;    /* maximum size of text table   */
    int    mszproc;    /* maximum size of proc table   */
    long   inodeovf;   /* cumulative overflows of
                          inode table since boot       */
    long   fileovf;    /* cumulative overflows of
                          file table since boot        */
    long   textovf;    /* cumulative overflows of
                          text table since boot        */
    long   procovf;    /* cumulative overflows of
                          proc table since boot        */
    time_t ts;         /* time stamp                   */
    long   devio[NDEVS][4]; /* device unit information */

#define IO_OPS  0  /* number of I/O requests since boot */
#define IO_BCNT 1  /* number of blocks transferred
                      since boot                       */
#define IO_ACT  2  /* cumulative time in ticks when
                      drive is active                  */
#define IO_RESP 3  /* cumulative I/O response time
                      in ticks since boot              */
};
```

FILES

/usr/lib/sa/sadc	
/usr/lib/sa/sa1	
/usr/lib/sa/sa2	
/usr/adm/sa/sa*dd*	daily data file
/usr/adm/sa/sar*dd*	daily report file
/tmp/sa.adrfl	address file

SEE ALSO

sag(1G), sar(1), timex(1), cron(1M).
"System Activity Package" in *A/UX Local System Administration*.

NAME

shutdown — close down the system at a given time

SYNOPSIS

/etc/shutdown [–hknry] [–g*interval*] [–i*initstate*] [*timeout* [*warning-message* ...]]

DESCRIPTION

shutdown provides an automated shutdown procedure which a superuser can use to notify users nicely when the system is shutting down, saving them from system administrators, hackers, and gurus, who would otherwise not bother with such niceties.

timeout is the time at which shutdown brings the system down and may be the word now (indicating an immediate shutdown) or specify a future time in one of two formats: *number* and *hour:min*. The first form brings the system down in *number* minutes and the second brings the system down at the time of day indicated (as a 24–hour clock). If *timeout* is not specified, shutdown prompts the user for a value.

At intervals which get closer together as apocalypse approaches, the *warning messages* are displayed at the terminals of all users on the system. The *interval* may be set with the –g option. If no message is specified on the command line, shutdown prompts the user for one.

If –k is specified, no further actions are taken. (If it isn't obvious, –k is to make people *think* the system is going down!)

When the *timeout* has expired, shutdown asks for final confirmation. Then a signal is sent to init(1M) to bring the system down to single-user state. All other user processes are then terminated. Subsequently, if –r or –h was used, then shutdown emulates reboot(1M), powerdown(1M), respectively. Alternatively, the final system state may be specified using the –i option. The *initstates* 6 and 0 are synonyms for the options –r and –h, respectively. The default *initstate* is 1.

The –n option prevents the normal sync(2) before stopping. If this option is used with the options –h or –r, the state of the file systems is not synchronized; this causes the file systems to be checked when the system is rebooted.

The –y option instructs shutdown to use a default *warning-message* and a default *timeout* of two minutes, if they are not supplied on the command line. The –y option is implied when the

options –h, –k or –r are used.

FILES

/etc/shutdown

SEE ALSO

reboot(1M), powerdown(1M), login(1), reboot(2).

BUGS

Only allows you to kill the system between now and 23:59 if you use the absolute time for shutdown.

NAME

slattach — attach serial lines as network interfaces

SYNOPSIS

/etc/slattach *ttyname* [*baudrate*]

DESCRIPTION

slattach is used to assign a tty line to a network interface. The *ttyname* parameter is a string of the form tty*XX* or /dev/tty*XX*. The optional *baudrate* parameter is used to set the speed of the connection. If the *baudrate* parameter is not specified, the default of 9600 is used.

After executing slattach, run ifconfig(1M) to define the network source and destination addresses. To specify the network source and destination addresses, use

ifconfig *interface-name address dest_address* up

The *interface-name* parameter is the name shown by netstat(1); either sl0 or sl1 under A/UX. *address* is the address of the local end of the slip point-to-point line. *dest_address* is the address of the remote end (the slip-serverhost) of the slip point-to-point line.

Only the superuser may attach a network interface.

To detach the interface, kill the slattach process, then use

ifconfig *interface-name* down

to quit the slip connection. *interface-name* is the name that is shown by netstat(1).

EXAMPLES

```
/etc/slattach ttyh8
ifconfig sl0 daisy-slip paris-slip

/etc/slattach /dev/tty01 1200
ifconfig sl1 daisy-slip paris-slip
```

DIAGNOSTICS

slattach produces messages indicating that the specified interface does not exist, the requested address is unknown, or the user is not privileged and tried to alter the configuration of an interface.

NAME

sprayd — spray server

SYNOPSIS

/usr/etc/rpc.sprayd

DESCRIPTION

sprayd is a server which returns information for spray(1). The sprayd daemon is normally invoked by inetd(1M).

FILES

/usr/etc/rpc.sprayd

SEE ALSO

spray(1M).

NAME

startmac, startmac24 — create a Macintosh environment

SYNOPSIS

startmac [-s *sysfolder*] [-f *findername*] [-o *option*[=*value*]] [-m *memsize*] [-P *patchfile*] [-S *systemfile*]

startmac24 [-u *user*] [-s *sysfolder*] [-f *findername*] [-o *option*[=*value*]] [-m *memsize*] [-P *patchfile*] [-S *systemfile*] *StartupApplication*

DESCRIPTION

startmac is used to create and initialize the Macintosh environment under A/UX®. Various options and environment variables can be used to change the default actions. startmac24 is similar to startmac except that it creates a 24-bit-only environment, thus allowing applications that are not 32-bit clean to run.

startmac will rarely be invoked directly by the user. It may be useful in those cases where the user wishes to have a console emulator login environment and still be able to run Macintosh applications and tools. Even then, the user will probably prefer to invoke mac32 or mac24, which will in turn invoke startmac or startmac24.

Wherever startmac is referenced below, startmac24 may be substituted.

FLAG OPTIONS

startmac and startmac24 accept the following options. Where environment variables may be used to set defaults, those environment variables will be overridden by the appropriate command line option.

-u *user* Specifies an alternate system folder. The preferred system folder will be taken from the indicated user's home directory, instead of invoking the user.

-s *sysfolder* Specifies an alternate system folder. The system folder will be the specified directory.

-f *findername* Specifies an alternate application to take the place of the Finder™. The name of the application is *findername* instead of Finder. The Finder or the application to run in place of the Finder is always found in the system folder. Multifinder

will automatically launch the Finder whenever no other applications exist and at initial startup.

`-o` *option* Specify one of the debugging options indicated below. *option* is a string of the form *option*[=*value*]. In each case, value should be either 0 or 1, although any nonzero value turns the feature on.

`warn`	Warning messages are printed to the invoking terminal. This is best used from a serial port or other nonconsole terminal.
`traps`	Each trap invocation generates a warning. The warning is not actually printed unless `warn` is also set.
`verbose`	More warnings than you could possibly want to see are generated. Once again, `warn` must be set for them to actually be printed.
`ram`	Copy the ROM into RAM. This can be used to facilitate debugging as breakpoints are easier to set.

`-m` *memsize* Sets the size of the virtual Macintosh the environment will be run in. *memsize* is a scaled value of the form *number scale*. *scale* is allowed to be

`m`, `M` *number* is in megabytes
`k`, `K` *number* is in kilobytes

`-P` *patchfile* Sets the name of the patch file to be used. Normally, the appropriate patch file name will be constructed by concatenating the string "Patch." and the hexadecimal ROM version number. Currently, the valid patch files are `Patch.0178` (for Macintosh II, Macintosh IIx, and Macintosh SE/30 computers) and `Patch.067C` (for Macintosh IIci and Macintosh IIfx computers). The patch file is always found in the system folder.

-S *systemfile* Sets the name of the system file to be used. Normally, this is "System." The system file is always found in the system folder.

ENVIRONMENT

The following environment variables may be used to affect the defaults of `startmac`. They are especially useful in that if they are set in the user's `.login` or `.profile` initialization script, they will affect the Macintosh environment created when the user logs in to a Macintosh environment. If both an environment variable and the appropriate command line option are specified, the command line option always takes precedence.

`TBRAM` Same as -o *ram* debug option. Copies the ROM into RAM to facilitate debugging.

`TBTRAP` Same as -o *trap* debug option. Prints a message for each trap invocation.

`TBWARN` Same as -o *warn* debug option. Causes warning messages to be printed.

`TBSYSTEM` Same as -s option. Changes the default system folder to be used.

`TBPATCHES` Same as -P option. Changes the default patch file to be used.

`TBMEMORY` Same as -m option. Changes the virtual memory size to be used.

SYSTEM FOLDER

Many files important to the Macintosh environment are found within the system folder. It is, therefore, important to know where `startmac` finds the system folder. `startmac` follows the following logic looking for a system folder:

1. If the -s option is specified, looks in the specified directory.
2. Checks for a directory named "System Folder" located in the user's home directory. The user's home directory is normally in the environment variable HOME, unless the -u option is specified, in which case it will be that user's home directory. If neither is set, the home directory will be the current user's home directory.
3. For `startmac24` only, checks the directory named "System Folder24" located in the user's home directory.

4. Checks the /mac/sys/SystemFolder.

FILES

/mac/bin/startmac
/mac/bin/startmac24
{*systemfolder*}/Patch.{*RomID*}
{*systemfolder*}/Finder
{*systemfolder*}/MultiFinder
{*systemfolder*}/System

SEE ALSO

mac32(1), mac24(1).

BUGS

Only the most common and useful options are documented here.

NAME

StartMonitor — display a progress bar during the A/UX® boot sequence

SYNOPSIS

StartMonitor

DESCRIPTION

StartMonitor displays a Macintosh® dialog box with a progress bar during the latter stages of the A/UX boot process (after A/UX startup has passed control to the kernel and the kernel has launched init. macsysinit, which is the first entry in /etc/inittab, invokes startmac with StartMonitor as the "Finder™" application (using the -f flag of startmac).

StartMonitor receives messages via a System V message queue from other processes involved in booting the system. These messages indicate the total number of boot phases, the current boot phase, what percentage of that phase has finished, the ID of messages to be displayed, and the strings to substitute for parameters in the message strings. The message strings are stored in the string list in the StartMonitor resource file (/mac/sys/Startup System Folder/%StartMonitor). The message ID is its position in the string list.

StartMonitor exits when it receives a quit message.

Shell scripts involved in booting invoke startmsg with appropriate arguments to send messages. fsck sends messages directly to StartMonitor. After the root file system has been checked, applications that need to send messages to StartMonitor could invoke startmsg to do it for them. If many messages need to be sent, the application could create a pipe to startmsg and write startmsg argument strings into the pipe.

FILES

/mac/sys/Startup System Folder/StartMonitor
/mac/sys/Startup System Folder/%StartMonitor
/usr/include/sys/startmsg.h

SEE ALSO

macsysinit(1M), startmsg(1M).

NAME

syslogd — log systems messages

SYNOPSIS

syslogd [-f*configfile*] [-m*markinterval*] [-d]

DESCRIPTION

syslogd reads and logs messages into a set of files described by the configuration file /etc/syslog.conf. Each message is one line. A message can contain a priority code, marked by a number in angle braces at the beginning of the line. Priorities are defined in <sys/syslog.h>. syslogd reads from the domain socket /dev/log, from an Internet domain socket specified in /etc/services.

syslogd configures when it starts up and whenever it receives a hangup signal. Lines in the configuration file have a *selector* to determine the message priorities to which the line applies and an *action*. The *action* field are separated from the selector by one or more tabs.

Selectors are semicolon separated lists of priority specifiers. Each priority has a *facility* describing the part of the system that generated the message, a dot, and a *level* indicating the severity of the message. Symbolic names may be used. An asterisk selects all facilities. All messages of the specified level or higher (greater severity) are selected. More than one facility may be selected using commas to separate them. For example:

```
*.emerg;mail,daemon.crit
```

selects all facilities at the emerg level and the mail and daemon facilities at the crit level.

Known facilities and levels recognized by syslogd are those listed in syslog(3) without the leading LOG_. The additional facility mark has a message at priority LOG_INFO sent to it every 20 minutes (this may be changed with the -m flag). The mark facility is not enabled by a facility field containing an asterisk. The level none may be used to disable a particular facility. For example,

```
*.debug;mail.none
```

sends all messages *except* mail messages to the selected file.

The second part of each line describes where the message is to be logged if this line is selected. There are four forms:

- A filename (beginning with a leading slash). The file will be opened in append mode.
- A hostname preceeded by an at sign ("@"). Selected messages are forwarded to the `syslogd` on the named host.
- A comma-separated list of users. Selected messages are written to those users if they are logged in.
- An asterisk. Selected messages are written to all logged-in users.

Blank lines and lines beginning with '#' are ignored.

For example, the configuration file:

```
*.notice;mail.info /usr/spool/adm/syslog
*.crit      /usr/adm/critical
*.emerg     *
*.alert     eric,kridle
*.alert;auth.warning    ralph
```

logs all notice (or higher) level messages and all mail system messages except debug messages into the file `/usr/spool/adm/syslog`, and all critical messages into `/usr/adm/critical`. All users will be informed of any emergency messages, the users `eric` and `kridle` will be informed of any alert messages, and the user `ralph` will be informed of any alert message, or any warning message (or higher) from the authorization system.

The flags are:

`-f` Specify an alternate configuration file.

`-m` Select the number of minutes between mark messages.

`-d` Turn on debugging.

`syslogd` creates the file `/etc/syslog.pid`, if possible, containing a single line with its process id. This can be used to kill or reconfigure `syslogd`.

To bring `syslogd` down, it should be sent a terminate signal (e.g. `kill` `` `cat /etc/syslog.pid` ``).

FILES

/etc/syslogd	
/etc/syslog.conf	the configuration file
/etc/syslog.pid	the process id
/dev/log	Name of the domain datagram log socket

SEE ALSO

syslog(3)

BUGS

syslogd should log kernel messages, but does not.

NAME

tzic — time zone compiler

SYNOPSIS

tzic [-v] [-d *directory*] [-l *localtime*] [-p *posixrules*] [-L *leapsecondfilename*] [-s] [*filename...*]

DESCRIPTION

tzic reads text from the file(s) named on the command line and creates the time-conversion information files specified in this input. If a *filename* is -, the standard input is read.

tzic interprets the following flag options:

-v Complain if a year that appears in a data file is outside the range of years representable by time(2) values.

-d *directory*
Create time conversion information files in the named directory rather than in the standard directory named below.

-p *timezone*
Use the rules of the given time zone when handling POSIX-format time zone environment variables. tzic acts as if the file contained a link line of the form

Link timezone posixrules

-l *localtime*
Use the given time zone as local time. tzic acts as if the file contained a link line of the form

Link timezone localtime

-L *leapsecondfilename*
Read leap-second information from the file with the given name. If this option is not used, no leap-second information appears in output files.

-s Limit time values stored in ouput files to values that are the same whether they are taken to be signed or unsigned. You can use this option to generate files compatible with the System V Interface Definition (SVID).

Input lines are made up of fields. Fields are separated from one another by any number of space characters. Leading and trailing spaces on input lines are ignored. An unquoted number sign character (#) in the input introduces a comment which extends to the end of the line the number sign character appears on. Space char-

acters and number sign characters may be enclosed in double quotes (") if they're to be used as part of a field. Any line that is blank after comment stripping is ignored. Nonblank lines are expected to be of one of three types: rule lines, zone lines, and link lines.

A rule line has the form

`Rule` *NAME FROM TO TYPE IN ON AT SAVE LETTER/S*

An example is:

```
Rule USA 1969 1973 - Apr lastSun 2:00 1:00 D
```

The fields that make up a rule line are:

NAME The (arbitrary) name of the set of rules this rule is part of.

FROM The first year in which the rule applies. The word `minimum` (or an abbreviation) means the minimum year with a representable time value. The word `maximum` (or an abbreviation) means the maximum year with a representable time value.

TO The final year in which the rule applies. In addition to `minimum` and `maximum` (as above), the word `only` (or an abbreviation) may be used to repeat the value of the *FROM* field.

TYPE The type of year in which the rule applies. If *TYPE* is –, then the rule applies in all years between *FROM* and *TO* inclusive. If *TYPE* is `uspres`, the rule applies in United States presidential election years. If *TYPE* is `nonpres`, the rule applies in years other than U.S. presidential election years. If *TYPE* is something else, then `tzic` executes the command

`yearistype` *year type*

to check the type of a year. An exit status of 0 is taken to mean that the year is of the given type; an exit status of 1 is taken to mean that the year is not of the given type.

IN The month in which the rule takes effect. Month names may be abbreviated.

ON The day on which the rule takes effect. Recognized forms include:

`5`	The fifth of the month
`lastSun`	The last Sunday in the month
`lastMon`	The last Monday in the month
`Sun>=8`	The first Sunday on or after the 8th
`Sun<=25`	The last Sunday on or before the 25th

Names of days of the week may be abbreviated or spelled out in full. Note that there must be no spaces within the *ON* field.

AT The time of day at which the rule takes effect. Recognized forms include:

`2`	Time in hours
`2:00`	Time in hours and minutes
`15:00`	24-hour time format (times after noon)
`1:28:14`	Time in hours, minutes, and seconds

Any of these forms may be followed by the letter `w` if the given time is local wall-clock time or `s` if the given time is local standard time. In the absence of `w` or `s`, wall-clock time is assumed.

SAVE The amount of time to be added to local standard time when the rule is in effect. This field has the same format as the *AT* field (although, the `w` and `s` suffixes are not used).

LETTER/S The "variable part" (for example, the `S` or `D` in `EST` or `EDT`) of time-zone abbreviations to be used when this rule is in effect. If this field is `-`, the variable part is null.

A zone line has the form

`Zone` *NAME GMTOFF RULES/SAVE FORMAT [UNTIL]*

An example is:

```
Zone Australia/South-west 9:30 Aus CST 1987 Mar 15 2:00
```

The fields that make up a zone line are:

NAME The name of the time zone. This is the name used in creating the time conversion information file

for the zone.

GMTOFF The amount of time to add to Greenwich mean time (GMT) to get standard time in this zone. This field has the same format as the *AT* and *SAVE* fields of rule lines. The field must begin with a – (minus sign) if time must be subtracted from GMT.

RULES/SAVE The name of the rule(s) that apply in the time zone or, alternately, an amount of time to add to local standard time. If this field is –, then standard time always applies in the time zone.

FORMAT The format for time-zone abbreviations in this time zone. The pair of characters `%s` is used to show where the "variable part" of the time-zone abbreviation goes.

UNTIL The time at which the GMT offset or the rule(s) change for a location. It is specified as a year, a month, a day, and a time of day. If this is specified, the time-zone information is generated from the given GMT offset and rule change until the time specified.

The next line must be a "continuation" line. This line has the same form as a zone line except that the string `Zone` and the name are omitted because the continuation line places information starting at the time specified as the *UNTIL* field in the previous line in the file used by the previous line. Continuation lines may contain an *UNTIL* field, just as zone lines do, to indicate that the next line is a further continuation.

A link line has the form

```
Link LINK-FROM LINK-TO
```

An example:

```
Link    US/Eastern    EST5EDT
```

LINK-FROM
The same as the *NAME* field in some zone line.

LINK-TO
An alternate name for the same *NAME* field as above in that zone line.

Except for continuation lines, lines may appear in any order in the input.

Lines in the file that describe leap seconds have the following form:

```
Leap YEAR MONTH DAY HH:MM:SS CORR R/S
```

An example is:

```
Leap 1974 Dec 31 23:59 + S
```

The `YEAR`, `MONTH`, `DAY`, and `HH:MM:SS` fields tell when the leap second happened. The `CORR` field should be + if a second was added or - if a second was skipped. The `R/S` field should be (an abbreviation of) Stationary if the leap second time given by the other fields should be interpreted as GMT or (an abbreviation of) Rolling if the leap second time given by the other fields should be intpreted as local wall clock time.

NOTES

For areas with more than two types of local time, you may need to use local standard time in the *AT* field of the earliest transition time's rule to ensure that the earliest transition time recorded in the compiled file is correct.

FILES

`/etc/tzic`	
`/etc/zoneinfo`	Standard directory used for created files

SEE ALSO

`tzdump`(1M), `ctime`(3), `tzfile`(4).

See mount(1M)

Sometimes a device file references a port rather than an actual device. For example, a printer connected through the printer port can be referenced as `/dev/tty1`, as described in `serial`(7). In this case, the device file references the first serial port. Conventionally, you can use the first Macintosh serial port to attach a serial printer. It is also possible to attach a modem or an A/UX user terminal to this port.

Major and minor device numbers

Programs that support devices, such as hard disks and tape drives, are known as device drivers. One device driver usually controls all the instances of one type of device. The exception is the serial device driver. It provides low-level support for a variety devices that are capable of communicating over a serial communications port. For example, a serial device driver helps control user terminals, modems, serial printers, and similar serial input/output devices.

To allow many devices of the same type to be controlled by the same device driver, each can be assigned a unique minor number using `mknod`(1M). This number is passed to the device driver for interpretation. By convention, the minor number may appear as the last part of the name of the device file for particular classes of devices. For example, `/dev/tty0` refers to the first serial port, or the port with minor device number 0. Note that this is a naming convention only, and is achieved by using `mknod`. In reality, device selection is realized through the minor number exclusively. The `-l` option of `ls` will show what major and minor device numbers have been assigned to a device file of a particular name, helping you verify whether a device file was named reasonably well.

The minor number is often used by a device driver as an inidication of the intended operational modes. For example, the device files for cartridge tape drives that contain an `n` suffix select a non-rewinding mode of tape operation through a minor number that is correctly interpreted by the device driver.

NAME

appletalk — general AppleTalk interface

DESCRIPTION

AppleTalk for A/UX provides high-performance EtherTalk and low-cost LocalTalk connectivity for A/UX systems. With an EtherTalk interface card, TCP/IP network services can be used concurrently with EtherTalk network services. A/UX support for LocalTalk has been extended so that the standard Macintosh serial ports can be used. This enables AppleTalk printing in any A/UX hardware configuration, either from Macintosh applications or the command line.

AppleTalk Phase 2 is incorporated as well. Its benefits include extended addressing, reduced network traffic, and improved routing.

A/UX applications that make calls to the old AppleTalk interface may not work with AppleTalk for A/UX. If you have any A/UX applications that make calls to the A/UX Toolbox, you must relink them to the Toolbox in order for those applications to function properly.

SEE ALSO

appletalk(1M), newunix(1M), close(2), fcntl(2), open(2), read(2), select(2), write(2), atp(3N), ddp(3N), lap(3N), nbp(3N), pap(3N), rtmp(3N), zip(3N), appletalkrc(4), arp(5), fcntl(5), forwarder(7), termio(7);
"AppleTalk Programming Guide" in
A/UX Network Applications Programming;
"Installing and Administering AppleTalk," in
A/UX Network System Administration;
Inside AppleTalk.

NAME

read_disk — retrieve information from floppy disks during the installation of A/UX

SYNOPSIS

read_disk

DESCRIPTION

read_disk helps retrieve information from floppy disks during the installation of A/UX. This command is only available in the A/UX Startup environment; it does not have an A/UX counterpart.

The command changes your current directory to the root directory and then ejects the floppy disk (if any) that is in the drive. A dialog box is then presented to tell the user to insert the next installation disk. read_disk then extracts the information from the floppy based on the format of the disk. read_disk uses one of the following three commands:

```
cpio -idm
cpio -idumc
tar xf /dev/dsk/c8d0s0
```

SEE ALSO

cpio(1), tar(1).

THE APPLE PUBLISHING SYSTEM

This Apple manual was written, edited, and composed on a desktop publishing system using Apple Macintosh® computers and Microsoft Word software. Proof pages were created on Apple LaserWriter® printers. Final pages were created on the Linotronic L300 imagesetter. Line art was created using Adobe Illustrator. POSTSCRIPT®, the page-description language for the LaserWriter, was developed by Adobe Systems Incorporated.

Text type and display type are Apple's corporate font, a condensed version of ITC Garamond. Bullets are ITC Zapf Dingbats®. Some elements, such as program listings, are set in Apple Courier.

Writer: Kristi Fredrickson

Developmental Editor: Loralee Windsor

Illustrator: Sandee Karr

Production Supervisor: Josephine Manuele

Formatter: Teresa Lujan

Special thanks to Vicki Brown for her elegant engineering, to Li Greiner for his artistic expertise, to Dave Payne and Eryk Vershen for their technical expertise, and to Shawn Morningstar and Tony Francis for making sure that things actually work as written

Acknowledgments to Chris Wozniak for her expert guidance, and to Sharon Everson and Lori Kaplan for their exceptional work in developing previous versions of the *A/UX Installation Guide*

030-0754-A