

## **Version 3.51 UNIX Utilities**

You'll find the following software included with Version 3.51 of the UNIX Utilities:

**Development Set** (8 Disks)—Libraries and tools for developing software applications.

**Document Preparation** (4 Disks)—Text formatting tools.

**Enhanced Editors** (1 Disk)—Contains editors: vi, edit, ex and view; UNIX command bfs.

**Curses/Terminfo Programmer's Package** (1 Disk)—Upgrades curses library routines to System V Release 3. You **MUST** install the Development Set **PRIOR** to installing this package.

**VDI** (1 Disk)—Virtual Device Interface Library (GSS-Drivers C-Language Binding).

**Windowing Utilities (Xt/Layers)** (1 Disk)—Windowing Utilities for AT&T 5620, 615, and 620 Terminals.

All of these disks are optional. Some, however, may depend on one or more other disks being installed. These are noted above.

## AT&T UNIX PC Utilities - Version 3.5.1

This section contains important information for using the UNIX Utilities version 3.51. Please review this section before using the UNIX Utilities and keep it with your *AT&T UNIX<sup>®</sup> PC User's Manual*.

### NEW FEATURES:

#### 1. New Manual Pages

The following is a list of manual pages which were not included in the previous release. The entries for these manual pages will be found neither in the Table of Contents nor the Permuted Index of the *User's Manual*.

470(1)	478(1)	arconvert(1)
checkmm(1)	convert(1)	fx(1)
hfc_ctl(1)	hplj(1)	ismpx(1)
jterm(1)	jwin(1)	layers(1)
layers(5)	man(1)	ndx(1)
neqn(1)	nterm(5)	subj(1)
xtd(1M)	xts(1M)	xt(7)
xtt(1M)		

#### 2. 132-Character Font

A 132-Character font which is described in the file */usr/lib/wfont/PLAIN.R.E.11.A*, is now available. It is installed when you install the Asynchronous Terminal Emulator.

#### 3. arconvert, convert (1)

**convert (1)** is now included with UNIX Utilities. A shell script **arconvert (1)** provides a simpler interface for using this tool. See the manual page for more information.

**4. hfc\_ctl(1)**

Hardware flow control is now available on the serial ports of the UNIX PC. See the manual page for more information.

**5. ktune(1)**

**ktune(1)** provides a flag, **-r**, that will reset all tunable parameters to their default values. See the manual page for more information.

**6. UNIX Windowing Utilities (XT/Layers)**

UNIX System V Release 3 Windowing Utilities are provided with version 3.51. For complete information, refer to "AT&T UNIX PC Windowing Software XT/Layers - Version 3.51" in this section.

**7. Documentation Preparation Set**

The Documentation Preparation package has been extensively revised with version 3.51. For complete information, refer to the "AT&T UNIX PC Documentation Preparation Set - Version 3.51" in this section.

## TECHNICAL TIPS:

### 1. mas and mcc

If a language or application needs **mas** or **mcc**, the commands may be accessed as follows:

- a. Log in as *root*
- b. *cd* to */bin*
- c. type: **ln as mas** and **ln cc mcc**
- d. There is no guarantee that *mas* or *mcc* will work exactly as before.

### 2. Shared Library

In order to use the shared library, the user program must be linked with a special *ifile* and runtime startoff. Typically, the command line is written as:

```
ld /lib/crt0s.o /lib/shlib.ifile -o target usrprog -lx
```

where *-lx* may be any library other than *libc.a*.

The shared library is intended to contain all of the routines loaded by *-lc*, *-liam*, and *-ltermib*. With the delivery of new system software, *libc.a* is contained in the shared library. The user should note that the packaging of the software has the shared library in the Foundation Set and *libc.a* in the Development Set. If for any reason, the user believes that the contents of the shared library and *libc.a* do not correspond, such as might result from loading different releases of the software or from deliberate modifications to either of the libraries, then the shared library may be used by invoking a two step link edit process as follows:

Step 1: `ld -r -o output /lib/crt0s.o /lib/shlib.ifile object_files <Return>`

Step 2: `ld -o target output -lx -lc <Return>`

It is recommended that the user create private libraries rather than modify *libc.a* or the shared library. A user-modified *libc.a* or shared library will not be supported.

### 3. Math Library

For several of the routines in the math library, the number HUGE is returned when the result of the function is outside the floating point range. On the AT&T UNIX PC, a floating point number can have an exponent as large as 307, and when exceeded, HUGE is returned with a value of 3.4...e38.

HUGE has been defined in <math.h> as being equivalent to MAXFLOAT which in <values.h> is 3.402823...e38. HUGE should have been defined as equivalent to MAXDOUBLE which in <value.h> is 1.797693...e307.

Users requiring double precision accuracy may want to handle HUGE by defining their own error handling procedure and including it in a function named **matherr** which must be of the form described in **matherr(3M)**.

### 4. Installable Software

When creating an installable software package, do not use any special characters such as a single quote (') in the name.

### 5. Debugging C Source Code

The C compiler will catch some but not all of the bugs in C programs. Programmers may use the C program checker **lint(1)** for better debugging. **Lint** enforces the rules of C more strictly than the C compiler. See the *Programmer's Guide* for details.

### 6. Diagnostic Tests

Do not abort a diagnostic test of communications that you have initiated from a remote terminal. This may cause the host system to hang. If this happens, you must press the reset button on the rear panel of the host to reboot the system.

### 7. Hard Disk Surface Test

If on running a hard disk surface check, you see a message describing a bad-block table overflow, your hard disk probably suffered extensive damage and should be replaced. Please contact your AT&T Customer Services Support representative or your authorized AT&T Dealer for further assistance.

8. **cc(1)**

The C compiler supports flexnames. Programs that have symbol names longer than eight characters are not truncated automatically as was done with the pre-flexname compiler, which was available in Version 2.0 and 3.0.

A new option, **-T**, is provided to truncate long symbol names to eight characters. This is useful so that new object files can be linked with older (pre-flexname) files and then run under either old or new systems.

9. **cu(1)**

On single phone line systems where the Phone Manager has been installed, make sure the line is selected to DATA 1 before the **cu** command is used.

10. **cut(1)**

The **cut(1)** command returns a value of 2 whether or not execution was successful.

11. **dial(3C)**

The calling program for the **dial(3C)** function must include a statement which defines a static *termio* structure and assigns its address to the *attr* element of the CALL structure. This is necessary because a NULL value for the *attr* element in CALL cannot be passed to the **dial** function even though the manual page says that a NULL value for this pointer element may be passed. The problem is that *attr* is a pointer to a *termio* structure that has been defined as an auto variable, hence it does not get initialized. The structure should have been declared as static for the variable to default initially to zero.

On compiling the calling program, the user will see that the **dial(3C)** function references the functions *tgetent*, *tgetstr*, and *tgetnum* which are NOT in *libc.a*. In order to resolve this, the user should invoke the compiler as:

**\$cc prog.c -lc -ltermcap**

(note: **-lc** and **-ltermcap** must be in this order.)

or invoke the shared library as

```
$cc -c prog.c
```

```
$ld /lib/crt0s.o /lib/shlib.ifile prog.o
```

12. **grep(1)**

For **grep**, the **-b** option reports the block number of the match relative to 512-byte blocks. For **egrep** and **fgrep**, the **-b** option reports the block number relative to 1024-byte blocks.

13. **ld(1)**

The Link Editor, **ld** provides a new option, **-G**, that resolves symbol referencing errors between new flexnames-generated files containing long symbol names and older (pre-flexname) object files. Under Version 3.51, this option assures backwards compatibility with old files.

14. **lp(1)**

If you have opened a UNIX window from the Office, **lp -w** will corrupt the Office window. If this happens, you can use the Refresh command, **<Rfrsh>**, to restore it. If you have accessed a shell directly after logging in, the **-w** option works as documented.

15. **menu(3T)**

**menu(3T)** incorrectly states that a newline character in a menu title will cause all lines of the title except the first line to be underlined. This is not the case. A newline character in the menu title will cause the menu process to hang, and therefore should never appear.

16. **setgrp(2)**

The C system call **setgrp** (set group process id) cannot be called from processes associated with windows. Any process calling **setgrp(0)** must have *stdin*, *stdout*, and *stderr* directed to other than window devices to work properly.

17. **shform(1)**

**shform(1)** expects the form in **formname** to reside in the current directory or in the directory specified by **pathname**, not in */usr/lib/ua*

as stated in the manual page.

**18. tam(3T)**

Using the alternative form of the *wslk()* routine, which writes all of the screen labeled keys at once, you will be unable to supply short (one line) labels as documented in the **tam(3)** manual pages. There is no problem writing all of the screen labeled keys at once with long (two line) labels.

If it is your intention to write all screen labeled keys at once with short (1 line) labels, the following *wslk()* procedure will accomplish this.

- a. Supply the appropriate window descriptor for *wn*.
- b. Supply the value 0 for *kn*.
- c. Treat the variable *slongl* as you normally would treat *sshort*. They are both pointers to 80 character strings. Assign the appropriate short labels to the variable *slongl*.
- d. Assign *NULL* to both variables *slong2* and *sshort*.

This will result in all of the screen labeled keys being written at once with short (one line) labels. Actually, the screen labeled keys are being written with long (two line) labels, but the second line of the label is *NULL*.

*This page was left blank intentionally.*

## AT&T UNIX PC Windowing Software (xt/layers) - Version 3.51

AT&T UNIX PC Windowing Software (xt/layers) is a SVR3 feature included with the AT&T UNIX PC Version 3.51 software. Layers are asynchronous windows supported by the operating system in a windowing terminal. Windowing terminals include the AT&T 5620 DMD, AT&T Teletype 615, and the AT&T Teletype 620. Users should consult terminal manuals for specific usage.

The AT&T Windowing Software is packaged on an installable floppy diskette. Software installation follows the normal install procedures (see AT&T UNIX PC Owner's Manual, p. 4-55).

The AT&T Windowing Utilities consist of seven commands. A brief description of these commands and a sample test script follow.

### Windowing Utilities Commands

The Windowing Utilities Commands are:

1. **ismpx(1)** - command reporting whether standard input is connected to a multiplexed xt channel running layers.
2. **jterm(1)** - command used to reset a layer of a windowing terminal after downloading a terminal program that changes the terminal attributes of the layer.
3. **jwin(1)** - command that runs only under layers and is used to determine the size of the layer associated with the current process.
4. **layers(1)** - manages asynchronous windows on a windowing terminal, finding an unused xt channel group, and associating it with the terminal line on its standard output.
5. **xtd(1M)** - extracts and prints xt driver link structure.
6. **xts(1M)** - extracts and prints xt driver statistics.
7. **xtt(1M)** - extract and print xt driver packet traces.

Manual pages describing these functions are included with this documentation.

### Sample Test Script

This test script assumes that the attached terminal is an AT&T 5620 DMD. If you are using another "windowing" terminal, consult the terminal manual for window creation information.

From the UNIX shell prompt:

1. Type "ismpx<cr>". The message "no" should appear on the screen and the command should have a return value of "1". Use "echo \$?" to verify return value.
2. Type "jwin<cr>". The message "jwin: not mpx" should appear on the screen.
3. Type "jterm<cr>". The message "jterm: Must be invoked with stderr attached to an xt channel." should appear on the screen.
4. Type "layers -t<cr>". The screen should clear, then go to a green tone. A mouse pointer will appear on the screen. Create a window as follows:
  - Press the third button on the mouse. A menu will appear on the screen.
  - Highlight "new" and release the button.
  - Reclick the third button, and expand image to desired window size.
  - Release the button. A window should form with a shell prompt in the upper left corner of the window.
5. Now type "ismpx<cr>" once again. The message "yes" should appear on the screen, and the return value should be "0".
6. Type "jwin<cr>". The result should be two lines like the following:  
bytes: 57 37  
bits: 526 604
7. Type "jterm<cr>". The window should clear and the cursor should move back to the upper left corner.
8. Type "xts<cr>". The date should appear followed by five lines of statistics.

9. Type "xtd<cr>". Output similar to the following should appear:

```
Link data structure
line 0x402525d8
rdatap 0x4023b0be
rpkt
<more>
```

10. Type "xtt<cr>". Output similar to the following should appear:

```
Dec 4 14:06:03 Xt Packet Traces:-
0.00 Ch 2 Rcv 7: <0333> <0325>
0.00 Ch 2 Xmt 1:0 x 402 [more]
<more>
```

11. Create five more windows and toggle between the windows using the first mouse button. Concurrent output can simultaneously appear in several windows. Attempt to create a seventh window should fail by returning the user to the last current window.

### Existing Problems

The following problems have been uncovered during system testing of the AT&T UNIX PC Windowing Software (Version 1.0):

1. A <defunct> process is spawned after "layers" is invoked, and a window is created. This process is cleaned up upon exiting "layers".
2. Attempt to select UNIX PC Office objects other than the "UNIX System" on the AT&T 5620 DMD terminal will fail. The user will be logged out.
3. Attempting to execute the UNIX PC User Agent (ua) from the AT&T 5620 DMD terminal while in layers will produce a memory fault - core dump.
4. Attempting to execute the UNIX PC User Agent (ua) from the AT&T Teletype 615 terminal causes the layers screen label keys to be irretrievably removed.
5. The User Agent (Office) should **not** be invoked from within a window created by layers.



## AT&T UNIX PC Document Preparation Set - Version 3.51

### 1. Introduction

The Document Preparation Set has been significantly enhanced with this release. The purpose of these notes is to highlight the changes and provide assistance in using the software. Please refer to the manual pages provided for a full description of each command and macro. The manual page and section are indicated by the name of the command or file followed by the section in parentheses, i.e. **nroff(1)**, in the discussion below.

**Nroff** and its associated programs are a powerful and complex text formatting system. We recommend that users obtain additional documentation to assist in making full use of **nroff**. Suggested documents are:

- AT&T Documenter's Workbench™ Software Release 2.0 manuals which contain a thorough discussion of **nroff** (and **troff**), macros, and preprocessors. Both of these manuals are recommended:
  - *UNIX System V Documenter's Workbench Software Release 2.0 User's Guide* (order using AT&T Select Code 310-004).
  - *UNIX System V Documenter's Workbench Software Release 2.0 Technical Discussion and Reference Manual* (order using AT&T Select Code 310-005).
- *Document Formatting and Typesetting on the UNIX System*, by Narain Gehani, Silicon Press, Summit N.J., 1986. This book contains a thorough discussion of document formatting using **nroff** and **troff**.

### 2. What You Get With the Document Preparation Set

- **nroff(1)** – the text formatter for constant-width printers which uses the following macros:
  - **mm(5)** – macros to format letters, memoranda, reports, and books.
  - **man(5)** – macros to format UNIX manual pages (new with this release).

- **mptx(5)** – macros to format permuted indices.
- **mosdd** – macros to prepare Operations Systems Deliverable Documentation.
- **tbl(1)** – the **nroff** preprocessor used to format tables.
- **neqn(1)** – the **nroff** preprocessor used to format mathematical equations.
- Various filters to modify **nroff**'s output to work correctly with certain printers: **col(1)**, **300(1)**, **4014(1)**, **450(1)**, **470(1)**, **478(1)**, **fx(1)**, **greek(1)**, **hp(1)** and **hplj(1)**.
- **mm(1)** – shell program that invokes **nroff** with the **mm** macros and appropriate output filters.
- **man(1)** – shell program that invokes **nroff** with the **man** macros and appropriate output filters.
- **osdd** (description in **mm(1)** manual page) – shell program that invokes **nroff** with the **mm** and **mosdd** macros.
- Programs to assist in checking documents:
  - **spell(1)** – checks documents for spelling errors.
  - **checkmm(1)** – checks for **mm** macro syntax errors in documents.
  - **hyphen(1)** – finds hyphenated words, useful to proofread **nroff**'s hyphenation.
  - **diffmk(1)** – marks differences between two versions of a document.
- Programs to assist in producing indices:
  - **ndx(1)** – produces a subject-page index of a document (new for this release).
  - **subj(1)** – generates a list of subjects from a document, which can be used by **ndx** (new for this release).
  - **ptx(1)** – produces a permuted index of a document.

- And the following additional utilities:
  - **macref(1)** – produces a cross-reference listing of macro files; useful for creating or modifying macros.
  - **cw(1)** – a preprocessor for **troff** (which is not provided with the Document Preparation Set) which formats the document to be printed using a constant-width font.
  - **deroff(1)** – removes **nroff**, **tbl**, and **neqn** constructs from a document. **Deroff** is used by **spell(1)** to strip a document of all **nroff** commands and **mm** macro calls before checking for misspelled words.
  - **newform(1)** – changes the format of a text file.

### 3. Changes From Version 3.5

The following programs from the Document Preparation Set Version 3.5 are not included in this release: **checkeq(1)**, **eqn(1)**, **mmt(1)** and **mvt(1)**.

The following programs are new with Version 3.51: **470(1)**, **478(1)**, **fx(1)**, **hplj(1)**, **man(1)**, **ndx(1)** and **subj(1)**.

The functionality of **mm(1)** and **nroff(1)** has been enhanced with Version 3.51. A full description of the changes can be found in the manual pages and in "How to use **nroff**" below.

## 4. How to Use **nroff**

Following are abbreviated instructions for using this release of **nroff**.

### 4.1 The **mm** and **man** Commands

**Mm(1)** executes **nroff** using the **mm** (memorandum) macros and automatically invokes whichever filters are appropriate (including **col**) for the defined printers. **Man(1)** does the same thing, except it invokes the **man** (manual page) macros instead of the **mm** macros. These programs greatly simplify the execution of **nroff**. Examples of the **mm** command are included in the "Sample Command Lines" section below.

## 4.2 Sample Command Lines

Table 1 contains sample **mm(1)** and **nroff(1)** command lines to use with several of the supported printers. The sample commands only show formatting requests for the default print pitch (10 characters per inch), although other print pitches are available for most printers. Refer to the **mm(1)** and **nroff(1)** manual pages for the available pitch and other command-line options. Notice that output is sent to **lp(1)** in *raw* mode in these samples. It may also be redirected to **/dev/rawlp**. Please refer to the "Nroff Printer Support" section below for a full explanation of output requirements.

These samples assume that the document does not contain tables or equations; if it were to contain either or both, **tbl(1)** or **neqn(1)** should be called before **nroff**. **Tbl** should always be called before **neqn** if both are required. Here are equivalent **mm** and **nroff** command lines for a document containing both tables and equations:

```
mm -t -e option(s) file(s)|lp -dprinter_R
```

```
tbl file(s)|neqn /usr/pub/eqnchar -|nroff -Ttty_type option(s)|filter(s)|  
lp -dprinter_R
```

## 4.3 Nroff Printer Support

**Nroff** now uses ASCII printer driving tables to prepare output. These tables, found in **/usr/lib/nterm**, include the definitions for printer initialization, underlining, bold, and **nroff** special characters. Your system administrator can modify the definitions for any printer or create new tables for any printer you choose. Refer to the **nterm(5)** manual page for an explanation of the driving tables.

Driving tables are included to support the following printers which were not supported in the prior release of the Document Preparation Set: AT&T 455, 470, 475, 478, and 479, Epson FX series, Hewlett-Packard LaserJet™ (Roman 8 character set), and IBM graphics printers.

If you examine the tables for these new printers, you will notice that for all printers except the AT&T 455 and HP LaserJet, the definitions for reverse and partial line feeds, and for certain escape codes, do not match the control codes actually used by the printers. This is because **col(1)** should be used to filter **nroff**'s output for these printers, and **col** only

TABLE 1. MM and NROFF Command Lines

PRINTER	COMMAND LINE
AT&T 455	<b>mm -T455 option(s) file(s)   lp -dATT455_R</b> <b>nroff -T455 option(s) file(s)   lp -dATT455_R</b>
AT&T 470	<b>mm -T470 option(s) file(s)   lp -dATT470_R</b> <b>nroff -T470 option(s) file(s)   col -xpf   470   lp -dATT470_R</b>
AT&T 475	<b>mm -T475 option(s) file(s)   lp -dATT475_R</b> <b>nroff -T475 option(s) file(s)   col -xpf   470   lp -dATT475_R</b>
AT&T 478	<b>mm -T478 option(s) file(s)   lp -dATT478_R</b> <b>nroff -T478 option(s) file(s)   col -xpf   478   lp -dATT478_R</b>
AT&T 479	<b>mm -T479 option(s) file(s)   lp -dATT479_R</b> <b>nroff -T479 option(s) file(s)   col -xpf   478   lp -dATT479_R</b>
C. Itoh	<b>mm -T8510 option(s) file(s)   lp -dC.Itoh_R</b> <b>nroff -T8510 option(s) file(s)   col -xpf   470   lp -dC.Itoh_R</b>
Epson FX	<b>mm -Tfx option(s) file(s)   lp -dEpson_R</b> <b>nroff -Tfx option(s) file(s)   col -xpf   fx   lp -dEpson_R</b>
HP LaserJet (Roman 8)	<b>mm -Thpljr8 option(s) file(s)   lp -dHPLaserjet_R</b> <b>nroff -Thpljr8 option(s) file(s)   hplj   lp -dHPLaserjet_R</b>
IBM Graphics	<b>mm -Tibm option(s) file(s)   lp -dprinter_R</b> <b>nroff -Tibm option(s) file(s)   col -xpf   478   lp -dprinter_R</b>
Line Printer	<b>mm -Tlp option(s) file(s)   lp -dOther_R</b> <b>nroff -Tlp option(s) file(s)   col   lp -dOther_R</b>

recognizes ESC-7, ESC-8, and ESC-9 for reverse, half-reverse, and half-forward line feeds. `col` also garbles any escape command that is more than two characters long. Furthermore, `nroff` cannot handle 8-bit special characters used by some printers, and this requires using the `SO` and `SI` characters to indicate extended characters. You should therefore use the filters `470(1)` (for AT&T 470, 475, and C. Itoh 8510 printers), `478(1)` (for AT&T 478, 479, and IBM printers), and `fx(1)` (for Epson FX printers) after `col` to output the printer's correct control codes. You should also use `hplj(1)` as a filter directly after `nroff` (do not use `col`), to output correct 8-bit extended characters for the HP LaserJet. The AT&T 455 printer does not require any filtering. The manual pages for these filters contain sample UNIX `nroff` command lines.

You must send these filters' output either to `/dev/rawlp` or to `lp(1)` in *raw* mode because `lp`'s normal filtering will cause unwanted effects.

Please note that new `nroff` special characters have been defined in the driving tables for these printers and several of the previously supported printers. These definitions are included to permit printer-supported special characters to be used with `neqn(1)`. This requires the use of two versions of `/usr/pub/eqnchar`. Please refer to the `eqnchar(5)` manual page for an explanation. A full list of `nroff` special characters is provided in Appendix A. This list was printed on a typesetter using `troff`. The appearance of the characters is specific to the typesetter used and will vary on other printers. Due to character set and incremental motion limitations, most `nroff` supported printers will only print a rough approximation of many special characters, and will not print `\(cs` and `\(vs`, the control-shift and visible space indicator characters. The `\(br`, `\(rb`, `\(lf`, and `\(lc` characters have been offset to the right one space in the table so they will be fully displayed without running into the characters above and below.

## 5. Memo Headings and Proprietary Markings

The file `/usr/lib/macros/strings.mm` is external to and supplements the `mm` macro package. The file contains string definitions enabling the user to specify a company's name, logo, and proprietary information to appear

on memos written with the formal memorandum macros. Refer to Appendix B for a sample of this file as provided with the Document Preparation Set and for instructions for modifying its contents.

## 6. Sample Documents

Sample documents are provided in `/usr/lib/dwb/samples`. All require using the `mm` macro. `eqn.stats` uses `neqn` and `tbl.bridges`, `tbl.language`, `tbl.pres` and `mm.sales` use `tbl`.

## 7. Cautions and Bugs

- Do not leave any blank lines at the beginning of a document because they will cause `mm` macros (such as `ND`, `TL`, and `AU`) to fail.
- If you provide an invalid `tty_type` to the `mm` command, it will produce output formatted for the `lp` printer type.
- The 'bold' command is ignored by `neqn`.
- When a document containing equations is processed using the `mm` macros, the `.EQ` and `.EN` equation delimiters normally must occur inside a `.DS/.DE` pair as follows, or erroneous output will result:

```
.DS
.EQ [ label ]
Equation(s) input
.EN
.DE
```

The exception is that if `.EQ` and `.EN` are only used to specify the delimiters for in-line equations or to specify `neqn` defines, the `.DS` and `.DE` macros must not be used; otherwise extra blank lines will appear in the output.

The `.EQ` macro takes an argument that will be used as a label for the equation. By default, the equation will appear at the right margin in the "vertical center" of the general equation. The `Eq` register may be set to 1 to change labeling to the right margin. The equation will be centered for centered displays; otherwise, the equation will be adjusted to the opposite margin from the label.

- The HP LaserJet printer has a default page length of 60 lines. Because **nroff** uses a default page length of 66 lines, the printer driving table (`/usr/lib/nterm/tab.hp1jr8`) for the HP LaserJet includes a printer initialization string which sets the page length to 66 lines. The printer is reset to its 60 line page default after the document is printed. Be aware that the HP LaserJet cannot print on the top two and bottom two physical lines of the page. Normally this will not cause a problem because these regions lie within the top and bottom margins of an **nroff** document.
- You should send **nroff** output to printers via **lp(1)** in *raw* mode because **lp**'s filtering causes undesirable effects. Invoke **lp** as follows: **lp -dprinter\_R**, where *printer* is the name of your printer administered through the "Printer Setup" option in the Administration window. You may also send **nroff** output to `/dev/rawlp`.
- The **checkmm(1)** command will flag some business letter macros as possible errors, even though a file containing the letter macros will format properly.
- The **checkmm** command will flag as possible errors the macros that produce labeled footnotes, if they occur inside lists. A file containing such a sequence of macros will format properly, however.
- **Col** may strip escape sequences off the end of documents. For example, the ESC-N command to the AT&T 470 (produced by the "twrest" definition in the printer driving table) is sometimes removed from the end of a document. If this happens and the **-T470-12** or **-T470-17** option to **mm** or **nroff** is used, the printer will not be reset to 10 cpi at the end of printing.
- Italics do not work correctly with the **mm** macros. **Mm** attempts to use the continuous style of emphasizing on headings by default. If the "iton" and "itoff" definitions in the printer driving tables contain the printer's correct "begin italics" and "end italics" codes, the result is headings in which the words are in italics and the spaces between them are underlined. This is because **mm** assumes that **nroff** will convert all requests for italics to underlining (an obsolete assumption) and inserts underscore characters in all the spaces.

In order to prevent this error, italics have been disabled by leaving the "iton" and "itoff" definitions in the printer driving tables empty. **Nroff** sees the empty definitions and proceeds to underline text instead of italicizing it.

If you wish to use italics with your printer, enter your printer's italicize commands in the appropriate printer driving table and then invoke the **mm** or **nroff** command with the **-rUI** flag for any documents which contain italicized text. This will force emphasizing of headings to the non-continuous style (only letters and digits will be italicized while spaces will be left blank).



## APPENDIX A

### Special Characters

#### 1. Input Names for ' , ‘ , and - and for Non-ASCII Special Characters

Char	Input Name	Character Name	Char	Input Name	Character Name
”	”	close quote	fi	\(fi	fi ligature
“	“	open quote	ff	\(ff	ff ligature
'	\(fm	foot mark	fl	\(fl	fl ligature
¢	\(ct	cent sign	ffi	\(Fi	ffi ligature
—	\(em	3/4 em dash	ffl	\(Fl	ffl ligature
-	\-	current font minus	¼	\(14	one-fourth
-	\(ru	rule	½	\(12	one-half
-	\(hy	hyphen	¾	\(34	three-fourths
-	-	literal hyphen	†	\(dg	dagger
°	\(de	degree	‡	\(dd	double dagger
●	\(bu	bullet	®	\(rg	registered
□	\(sq	square	©	\(co	copyright
■	\(bx	box	™	\(tm	trademark

## 2. Non-ASCII Characters and *Minus* on the Standard Fonts

The upper-case Greek letter names followed by † are mapped into upper-case English letters in whatever font is mounted on font position one (default Times Roman). The special math plus, minus, and equals are provided to insulate the appearance of equations from the choice of standard fonts.

Char	Input Name	Character Name	Char	Input Name	Character Name
A	\(*A	Alpha†	α	\(*a	alpha
B	\(*B	Beta†	β	\(*b	beta
Γ	\(*G	Gamma	γ	\(*g	gamma
Δ	\(*D	Delta	δ	\(*d	delta
E	\(*E	Epsilon†	ε	\(*e	epsilon
Z	\(*Z	Zeta†	ζ	\(*z	zeta
H	\(*Y	Eta†	η	\(*y	eta
Θ	\(*H	Theta	θ	\(*h	theta
I	\(*I	Iota†	ι	\(*i	iota
K	\(*K	Kappa†	κ	\(*k	kappa
Λ	\(*L	Lambda	λ	\(*l	lambda
M	\(*M	Mu†	μ	\(*m	mu
N	\(*N	Nu†	ν	\(*n	nu
Ξ	\(*C	Xi	ξ	\(*c	xi
O	\(*O	Omicron†	ο	\(*o	omicron
Π	\(*P	Pi	π	\(*p	pi
P	\(*R	Rho†	ρ	\(*r	rho
T	\(*T	Tau†	τ	\(*t	tau
Υ	\(*U	Upsilon	υ	\(*u	upsilon
Φ	\(*F	Phi	φ	\(*f	phi
X	\(*X	Chi†	χ	\(*x	chi
Ψ	\(*Q	Psi	ψ	\(*q	psi
Ω	\(*W	Omega	ω	\(*w	omega
Σ	\(*S	Sigma	σ	\(*s	sigma
			ς	\(ts	terminal sigma

Char	Input Name	Character Name	Char	Input Name	Character Name
+	<code>\pl</code>	math plus	$\infty$	<code>\if</code>	infinity
-	<code>\mi</code>	math minus	$\partial$	<code>\pd</code>	partial derivative
=	<code>\eq</code>	math equals	$\nabla$	<code>\gr</code>	gradient
*	<code>\**</code>	math star	$\neg$	<code>\no</code>	not
´	<code>\aa</code>	acute accent	$\propto$	<code>\pt</code>	proportional to
˘	<code>\ga</code>	grave accent	$\emptyset$	<code>\es</code>	empty set
—	<code>\ul</code>	underrule	$\in$	<code>\mo</code>	member of
/	<code>\sl</code>	slash (match b/s)		<code>\or</code>	or
√	<code>\sr</code>	square root		<code>\br</code>	box verticalrule
∛	<code>\rn</code>	root en extender	■	<code>\rh</code>	right hand
≥	<code>\&gt;=</code>	>=	■	<code>\lh</code>	left hand
≤	<code>\&lt;=</code>	<=	○	<code>\ci</code>	circle
≡	<code>\==</code>	identically equal	{	<code>\lt</code>	left top of big curly bracket
≈	<code>\~=</code>	approx =	l	<code>\lb</code>	left bottom
~	<code>\ap</code>	approximates	}	<code>\rt</code>	right top
≠	<code>\!=</code>	not equal	}	<code>\rb</code>	right bottom
→	<code>\-&gt;</code>	right arrow	{	<code>\lk</code>	left ctr of big curly bracket
←	<code>\&lt;-</code>	left arrow	}	<code>\rk</code>	right ctr of big curly bracket
↑	<code>\ua</code>	up arrow		<code>\bv</code>	bold vertical
↓	<code>\da</code>	down arrow	└	<code>\lf</code>	left floor
×	<code>\mu</code>	multiply	┘	<code>\rf</code>	right floor
÷	<code>\di</code>	divide	┌	<code>\lc</code>	left ceiling
±	<code>\+-</code>	plus-minus	┐	<code>\rc</code>	right ceiling
∪	<code>\cu</code>	cup (union)	˘	<code>\cs</code>	control-shift indicator
∩	<code>\ca</code>	cap (intersection)	⊂	<code>\vs</code>	visible space indicator
⊆	<code>\sb</code>	subset of	⊃	<code>\sc</code>	section
⊇	<code>\sp</code>	superset of			
⊈	<code>\ib</code>	improper subset			
⊉	<code>\ip</code>	improper superset			
∫	<code>\is</code>	integral sign			

The following special character definitions are used in `/usr/pub/cteqnchar` in creating `neqn` special symbols. `/usr/pub/o.neqnchar` does not use these characters and thus can be used with older `nroff` printer definition tables that do not define the characters.

Char	Input Name	EQN Keyword	Char	Input Name	EQN Keyword
$\approx$	<code>\(-~</code>	<code>-wig</code>	$\text{\AA}$	<code>\(ag</code>	<code>angstrom</code>
$*$	<code>\(b*</code>	<code>bigstar</code>	$\vee$	<code>\(os</code>	<code>orsign</code>
$\equiv$	<code>\(=D</code>	<code>=del</code>	$\wedge$	<code>\(as</code>	<code>andsign</code>
$\nabla$	<code>\(oA</code>	<code>oppA</code>	$\parallel$	<code>\(  </code>	<code>  </code>
$\exists$	<code>\(oE</code>	<code>oppE</code>	$\hbar$	<code>\(hb</code>	<code>hbar</code>
$\perp$	<code>\(pp</code>	<code>ppd</code>	$\sphericalangle$	<code>\(an</code>	<code>ang</code>
$\text{\rlap{/}L}$	<code>\(sL</code>	<code>scrL</code>	$\sphericalangle$	<code>\(ra</code>	<code>rang</code>
$\dots$	<code>\(3d</code>	<code>3dot</code>	$\leftrightarrow$	<code>\(&lt;</code>	<code>&lt;-&gt;</code>
$\dot{=}$	<code>\(=.</code>	<code>=dot</code>	$\Leftrightarrow$	<code>\(df</code>	<code>&lt;=&gt;</code>
$\notin$	<code>\(nm</code>	<code>nomem</code>	$\langle$	<code>\(ll</code>	<code>langle</code>
$\therefore$	<code>\(tf</code>	<code>thf</code>	$\rangle$	<code>\(rl</code>	<code>rangle</code>
$\sqcup$	<code>\(ic</code>	<code>incl</code>	$\oplus$	<code>\(c+</code>	<code>ciplus</code>
$\succsim$	<code>\(&gt;~</code>	<code>&gt;wig</code>	$\otimes$	<code>\(cx</code>	<code>citimes</code>
$\precsim$	<code>\(&lt;~</code>	<code>&lt;wig</code>			

## APPENDIX B

The file `/usr/lib/macros/strings.mm` defines 28 `nroff` strings. These string definitions are used by the `mm` macro package to specify a company's name, logo, and the proprietary information to appear on memos written with the formal memorandum macros. A sample of this file is on the next page. It has been annotated with comments describing how the strings are used by the `mm` macros.

Please notice that the logo character definition has been commented out since it can only be used by `troff`. To change the company name that appears at the top of memos and on cover sheets, replace the "AT&T" with your company's name in the `".ds}Z AT&T"` string definition. You may also change any of the string definitions for the six proprietary markings used by the `".PM"` macro.

```
"
                Copyright (c) 1984 AT&T
"
                All Rights Reserved
"
    THIS IS UNPUBLISHED PROPRIETARY SOURCE CODE OF AT&T
"
    The copyright notice above does not evidence any actual
"
    or intended publication of such source code.
"#ident "UNIX PC DWB"
"
    "\tab begins comments.
"
    No comments should appear on the same line as the string
"
    definition.
"
    The following string is used by the macro MT.
"
    ]S defined as logo character
"
    .ds]S \s36(Lb\s0
"
    }Z defined as Company Name
.ds}Z AT&T
"
    The following strings are used by the macro PM:
"
    Proprietary marking "PM1" (BP and default) lines 1-4
.ds]M \fBAT&T - PROPRIETARY\fR
.ds]O Not for use or disclosure outside of
.ds]Q AT&T or its affiliated companies
.ds]R except under written agreement.
"
    Proprietary marking "PM2" (P) lines 1-4
.ds]A \fBAT&T - PROPRIETARY (RESTRICTED)\fR
.ds]F Information herein is solely for persons having a need-
.ds]G to-know as authorized by the Director of the originating
.ds]H organization.  Reproduction is prohibited.
"
    Proprietary marking "PM3" (BR) lines 1-5
.ds]I \fBAT&T - PROPRIETARY (REGISTERED)\fR
.ds]J Information herein is solely for persons having a need-
.ds]K to-know as authorized by the Director of the originating
.ds]L organization.  Reproduction is prohibited.  This document
.ds]P is subject to cover sheet restrictions.
"
    Proprietary marking "PM4" lines 1-5
.ds]U \fBAT&T - PROPRIETARY (REGISTERED)\fR
.ds]V Information herein is solely for persons having a need-
.ds]W to-know as authorized by the Director of the originating
.ds]X organization.  Reproduction is prohibited.  This document
.ds]Y is subject to cover sheet restrictions.
"
    Proprietary marking "PM5" lines 1-4
.ds]i \fBAT&T - PROPRIETARY\fR
.ds]j Not for use or disclosure outside of
.ds]k AT&T or its affiliated companies
.ds]l except under written agreement.
"
    Proprietary marking "PM6" lines 1-4
.ds]m \fBAT&T - PROPRIETARY\fR
.ds]o Not for use or disclosure outside of
.ds]p AT&T or its affiliated companies
.ds]q except under written agreement.
```





**NAME**

470 – filters *nroff*(1) output to AT&T 470 (and C. Itoh 8510) printers.

**SYNOPSIS**

**nroff** -T470 files | col -xfp | 470 | lp -dprinter\_R

**nroff** -T475 files | col -xfp | 470 | lp -dprinter\_R

**nroff** -T8510 files | col -xfp | 470 | lp -dprinter\_R

**DESCRIPTION**

Input to 470 can be either *nroff*(1) or *col*(1) output. 470's output should be sent to *raw* lp. *Col* cannot handle eight-bit special characters. 470 expects the same syntax as *col* for indicating special characters; it sets the eighth bit on any characters between the ASCII SO (\017) and SI (\016) characters, and discards SO and SI from its output. To pass SO or SI to the printer (to control double-width printing), precede each with the escape character.

470 converts LF to CR-LF and it also translates the tty-37 vertical motion codes (ESC-7, ESC-8, ESC-9) to the correct AT&T 470 vertical motion codes. 470 issues the correct commands for all reverse or forward line motion whether 1/8" or 1/6" line spacing is set. The default value is 1/6" line spacing. ESC-B sets 1/8" line spacing, and ESC-A sets 1/6" line spacing.

470 sets the printer to incremental mode before the backspace character and vertical motion codes, and re-sets logic-seeking mode afterward (unless the printer is already in incremental mode).

The following translations are performed on input characters:

LF	output 'CRLF'
SO	set 8th bit on following characters
SI	stop setting 8th bit
ESC-SO	print double width
ESC-SI	cancel double width print
ESC-7	reverse lf
ESC-8	reverse 1/2 lf
ESC-9	forward 1/2 lf

**SEE ALSO**

*col*(1), *man*(1), *mm*(1), *nroff*(1), and *nterm*(5).

**NOTES**

It is strongly advised to use *col* with the AT&T 470 & 475 and C. Itoh printers in order to prevent misalignment of characters caused by reverse carriage motion.

The AT&T 470 & 475 printers must be set to receive 8-bit characters in order for special characters to be printed correctly. Set DIP switch 6 on switch block 2 to "off" or "open" to enable 8-bit characters.

**NAME**

478 - filters *nroff*(1) output to AT&T 478 & 479 and IBM Graphics printers.

**SYNOPSIS**

```
nroff -T478 files | col -xfp | 478 | lp -dprinter_R
nroff -T479 files | col -xfp | 478 | lp -dprinter_R
nroff -Tibm files | col -xfp | 478 | lp -dprinter_R
```

**DESCRIPTION**

Input to 478 can be either *nroff*(1) or *col*(1) output. 478's output should be sent to *raw* lp. *Col* cannot handle eight-bit special characters. 478 expects the same syntax as *col* for indicating special characters; it sets the eighth bit on any characters between the ASCII SO (017) and SI (016) characters, and discards SO and SI from its output. To pass SO or SI to the printer (to control double-width or compressed printing), precede each with the escape character.

478 translates the *tyt-37* vertical motion codes (ESC-7, ESC-8, ESC-9) to the correct AT&T 478 vertical motion codes. 478 issues the correct commands for all reverse or forward line motion whether 1/8" or 1/6" line spacing is set. The default value is 1/6" line spacing. ESC-\0 sets 1/8" line spacing, and ESC-\003 sets 1/6" line spacing.

478 converts each backspace to a carriage return and padding with blanks. The printer is set to unidirectional print mode whenever a backspace is encountered, then reset to bidirectional print mode upon the next linefeed (unless the printer was set to unidirectional print mode by a previous command).

*Col* only accepts two-character escape codes; since several printer functions are controlled by longer escape sequences, 478 translates two-character escape sequence aliases to their correct forms. The following translations are performed on input characters:

BS	replace BS with CR and overprinted blanks
SO	set 8th bit on following characters
SI	stop setting 8th bit
ESC-SO	print double width
ESC-SI	print compressed
ESC-DC2	print 10 cpi
ESC-DC4	cancel double width print caused by ESC-SO
ESC-\003	set 1/6" linefeed
ESC-0	set 1/8" linefeed
ESC-7	reverse lf
ESC-8	reverse 1/2 lf
ESC-9	forward 1/2 lf
ESC-I	set draft print mode
ESC-i	set NLQ print mode
ESC-S	print superscript

<b>ESC-s</b>	print subscript
<b>ESC-U</b>	set unidirectional print mode
<b>ESC-u</b>	set bidirectional print mode
<b>ESC-W</b>	print double width
<b>ESC-w</b>	cancel double width print caused by ESC-W
<b>ESC--</b>	begin underscore
<b>ESC-_</b>	end underscore

**SEE ALSO**

*col*(1), *man*(1), *mm*(1), *nroff*(1), *nterm*(5).

**NOTES**

It is strongly advised to use *col* with the AT&T 478 & 479 printers in order to prevent misalignment of characters caused by reverse carriage motion. *Col* must be used with IBM printers which are not capable of reverse line motion.

**NAME**

arconvert – archive conversion tool

**SYNOPSIS**

**arconvert** [**archive**]

**DESCRIPTION**

*Arconvert* uses the *convert(1)* utility to transform UNIX System Release 5.0 archive files to the UNIX System Release 5.2 archive format.

If no *archive* arguments are passed to *arconvert*, it will transform all files found in */lib* and */usr/lib* (and recursively under any of their subdirectories) whose name ends with the ".a" suffix, and whose type, as reported by the *file(1)* utility, is "5.0 archive".

If an argument is passed to *arconvert*, that file alone will be transformed if its type, as reported by the *file* utility, is "5.0 archive".

Any file which is converted is first saved under its pathname, with the suffix "-old" appended to it.

**FILES**

*/tmp/convlibs*, */tmp/archives*

**SEE ALSO**

*ar(1)*, *a.out(4)*, *ar(4)*, *find(1)*, *convert(1)*.

**NAME**

*brk*, *sbrk* – change data segment space allocation

**SYNOPSIS**

```
int brk (endds)
char *endds;

char *sbrk (incr)
int incr;
```

**DESCRIPTION**

*Brk* and *sbrk* are used to change dynamically the amount of space allocated for the calling process's data segment; see *exec(2)*. The change is made by resetting the process's break value and allocating the appropriate amount of space. The break value is the address of the first location beyond the end of the data segment. The amount of allocated space increases as the break value increases.

*Brk* sets the break value to *endds* and changes the allocated space accordingly.

*Sbrk* adds *incr* bytes to the break value and changes the allocated space accordingly. *Incr* can be negative, in which case the amount of allocated space is decreased. *Sbrk* clears only the page actually allocated, starting at a page boundary.

*Brk* and *sbrk* will fail without making any change in the allocated space if one or more of the following are true:

Such a change would result in more space being allocated than is allowed by a system-imposed maximum (see *ulimit(2)*).  
[ENOMEM]

Such a change would result in the break value being greater than the start address of any attached shared memory segment (see *shmop(2)*).

**RETURN VALUE**

Upon successful completion, *brk* returns a value of 0 and *sbrk* returns the old break value. Otherwise, a value of -1 is returned and *errno* is set to indicate the error.

**SEE ALSO**

*exec(2)*.

**NAME**

checkmm – checks documents formatted with the **mm** macros

**SYNOPSIS**

**checkmm** [ files ]

**DESCRIPTION**

*checkmm* stands for “check memorandum macros.” Use *checkmm* to check for syntax errors in files that have been prepared for the *nroff*(1) or *mm*(1) commands. For example, *checkmm* checks that you have a **.DE** (display end macro) corresponding to every **.DS** (display start macro).

The output for *checkmm* is the number of lines checked, and a list of macros that are unfinished because of missing macros. If you do not include a file name on the command line, *checkmm* takes input from standard input.

**SEE ALSO**

*mm*(1), *neqn*(1), *nroff*(1), *tbl*(1), and *mm*(5).

**DIAGNOSTICS**

“checkmm Cannot open *file(s)*” if *file(s)* is unreadable. The remaining output of the program is diagnostic of the source file.

**NAME**

*col* – filter reverse line-feeds

**SYNOPSIS**

*col* [ *-bfx* ]

**DESCRIPTION**

*col* reads from the standard input and writes onto the standard output. It performs the line overlays implied by reverse line feeds (ASCII code ESC-7), and by forward and reverse half-line-feeds (ESC-9 and ESC-8). *col* is particularly useful for filtering multicolumn output made with the *.rt* command of *nroff* and output resulting from use of the *tbl(1)* preprocessor.

If the *-b* option is given, *col* assumes that the output device in use is not capable of backspacing. In this case, if two or more characters are to appear in the same place, only the last one read will be output.

Although *col* accepts half-line motions in its input, it normally does not emit them on output. Instead, text that would appear between lines is moved to the next lower full-line boundary. This treatment can be suppressed by the *-f* (fine) option; in this case, the output from *col* may contain forward half-line-feeds (ESC-9), but will still never contain either kind of reverse line motion.

Unless the *-x* option is given, *col* will convert white space to tabs on output wherever possible to shorten printing time.

The ASCII control characters SO (\017) and SI (\016) are assumed by *col* to start and end text in an alternate character set. The character set to which each input character belongs is remembered, and on output SI and SO characters are generated as appropriate to ensure that each character is printed in the correct character set.

On input, the only control characters accepted are space, backspace, tab, return, new-line, SI, SO, VT (\013), and ESC followed by 7, 8, or 9. The VT character is an alternate form of full reverse line-feed, included for compatibility with some earlier programs of this type. All other non-printing characters are ignored.

Normally, *col* will ignore any escape sequences unknown to it that are found in its input; the *-p* option may be used to cause *col* to output these sequences as regular characters, subject to overprinting from reverse line motions. The use of this option is highly discouraged unless the user is fully aware of the textual position of the escape sequences.

**SEE ALSO**

*man(1)*, *mm(1)*, *nroff(1)*, and *tbl(1)*.

**NOTES**

The input format accepted by *col* matches the output produced by *nroff* with either the *-T37* or *-Tlp* options. If your output device is not supported by *nroff(1)*, you may use *-T37* (and the *-f* option of *col*) if the ultimate disposition of the output of *col* will be a device that can interpret half-line motions, and *-Tlp* otherwise.

**BUGS**

Cannot back up more than 128 lines.

Allows at most 800 characters, including backspaces, on a line.

Local vertical motions that would result in backing up over the first line

of the document are ignored. As a result, the first line must not have any superscripts.

*Col* will sometimes strip escape character sequences from the end of documents, which may render printer reset commands ineffective.

**NAME**

`convert` – convert object and archive files to common formats

**SYNOPSIS**

`convert` [-5] infile outfile

**DESCRIPTION**

*Convert* transforms input *infile* to output *outfile*. *Infile* must be different from *outfile*. The -5 option causes *convert* to work exactly as it did for UNIX system release 5.0. *Infile* may be any one of the following:

- 1) a pre-UNIX system 5.0 object file or link-edited (a.out) module (only with the -5 option),
- 2) a pre-UNIX system 5.0 archive of object files or link edited (a.out) modules (only with the -5 option), or
- 3) a UNIX system 5.0 archive file (without the -5 option).

*Convert* will transform *infile* to one of the following (respectively):

- 1) an equivalent UNIX system 5.0 object file or link edited (a.out) module (with the -5 option),
- 2) an equivalent UNIX system 5.0 archive of equivalent object files or link edited (a.out) modules (with the -5 option), and
- 3) an equivalent UNIX system 5.0 release 2.0 portable archive containing unaltered members (without the -5 option).

All other types of input to the *convert* command will be passed unmodified from the input file to the output file (along with appropriate warning messages). When transforming archive files with the -5 option, the *convert*(1) command will inform the user that the archive symbol table has been deleted. To generate an archive symbol table, this archive file must be transformed again by *convert* without the -5 option to create a UNIX system 5.0 archive file. Then the archive symbol table may be created by executing the *ar*(1) command with the *ts* option. If a UNIX system 5.0 archive with an archive symbol table is being transformed, the archive symbol table will automatically be converted.

**FILES**

/tmp/conv\*

**SEE ALSO**

*ar*(1), *a.out*(4), *ar*(4).

**NAME**

*diffmk* – mark differences between files

**SYNOPSIS**

**diffmk** [ - - ] file1 file2 file3

**DESCRIPTION**

*diffmk* compares two versions of a file and creates a third file that includes “change mark” requests (.mc) for *nroff*(1) or *troff*. *file1* and *file2* are the old and new versions of the file. *diffmk* generates *file3*, which contains the lines of *file2* plus inserted formatter “change mark” requests. When *file3* is formatted, changed or inserted text is shown by | at the right margin of each line. The position of deleted text is shown by a single asterisk: \*.

If anyone is so inclined, *diffmk* can be used to produce listings of C (or other) programs with changes marked. A typical command line for such use is

```
diffmk old.c new.c tmp; nroff macs tmp | pr
```

where the file *macs* contains

```
.pl 1
.ll 77
.nf
.eo
.ec `
```

The *.ll* request might specify a different line length, depending on the nature of the program being printed. The *.eo* and *.ec* requests are probably needed only for C programs.

**SEE ALSO**

*man*(1), *mm*(1), and *nroff*(1).

Where *diffmk* encounters – it uses the standard input.

**BUGS**

Aesthetic considerations may dictate manual adjustment of some output. File differences involving only formatting requests may produce undesirable output, i.e., replacing *.sp* by *.sp 2* will produce a “change mark” on the preceding or following line of output.

**NAME**

eqnchar – special character definitions for eqn and neqn

**SYNOPSIS**

```
eqn /usr/pub/eqnchar [ options ] [ - - ] [ files ] | troff [ options ]
neqn /usr/pub/eqnchar [ options ] [ - - ] [ files ] | nroff [ options ]
neqn /usr/pub/o.neqnchar [ options ] [ - - ] [ files ] | nroff [
options ]
```

**DESCRIPTION**

*/usr/pub/eqnchar* contains the following *troff* and *nroff*(1) character definitions that are not ordinarily available on a phototypesetter or printer. These definitions are primarily intended for use with *eqn* and *neqn*(1). *Eqnchar* contains definitions for the following characters:

ciplus	⊕			square	□
citimes	⊗	langle	<	circle	○
wig	~	rangle	>	blot	■
-wig	≡	hbar	ℏ	bullet	•
>wig	≧	ppd	⊥	prop	∞
<wig	≦	<->	↔	empty	∅
=wig	≐	<=>	↔	member	∈
star	*	<	⋈	nomem	∉
bigstar	*	>	⋈	cup	∪
=dot	≐	ang	∠	cap	∩
orsign	∨	rang	┌	incl	⊆
andsign	∧	3dot	⋮	subset	⊂
=del	≐	thf	∴	supset	⊃
oppA	∇	quarter	¼	!subset	⊄
oppE	∃	3quarter	¾	!supset	⊇
angstrom	Å	degree	°	scrL	ℓ
==<	≦	==>	≧		

*/usr/pub/eqnchar* is normally linked to */usr/pub/cateqnchar*.

*/usr/pub/cateqnchar* is a fairly "device-independent" version of *eqnchar* and should produce output that looks reasonable on most devices supported by *troff* and *nroff*. */usr/pub/o.neqnchar* must be used with *neqn* for the following older printers because the *nterm*(5) tables for these printers do not contain several special character definitions used by */usr/pub/cateqnchar*:

- Anderson Jacobson 832 terminal
- DASI-300
- DASI-300s
- DASI-450
- DTC-382
- GE Terminet 300 terminal
- Hewlett-Packard 2631
- Trendata 4000a

**FILES**

**/usr/pub/eqnchar**  
**/usr/pub/cateqnchar**  
**/usr/pub/o.neqnchar**

**SEE ALSO**

**mm(1), neqn(1), and nroff(1).**

**NAME**

*fx* – filters *nroff*(1) output to Epson FX-series printers.

**SYNOPSIS**

**nroff -Tfx files | col -xfp | fx | lp -dprinter\_R**

**DESCRIPTION**

Input to *fx* can be either *nroff* or *col*(1) output. *fx*'s output should be sent to *raw* lp.

*fx* translates the tty-37 vertical motion codes (**ESC-7**, **ESC-8**, **ESC-9**) to the correct Epson FX vertical motion codes. *fx* issues the correct commands for all reverse or forward line motion whether 1/8" or 1/6" line spacing is set. The default value is 1/6" line spacing. **ESC-0** sets 1/8" line spacing, and **ESC-3** sets 1/6" line spacing.

*Col* only accepts two-character escape codes; since several printer functions are controlled by longer escape sequences, *fx* translates two-character escape sequence aliases to their correct forms, permitting use of *col* with *fx*.

The following translations are performed on input characters:

<b>ESC-DC2</b>	cancel compressed print
<b>ESC-DC4</b>	cancel expanded print
<b>ESC-\003</b>	set 1/6" linefeed
<b>ESC-0</b>	set 1/8" linefeed
<b>ESC-7</b>	reverse lf
<b>ESC-8</b>	reverse 1/2 lf
<b>ESC-9</b>	forward 1/2 lf
<b>ESC-R</b>	select international character set 1 (French)
<b>ESC-r</b>	select international character set 0 (U.S.)
<b>ESC-S</b>	print superscript
<b>ESC-s</b>	print subscript
<b>ESC-U</b>	set unidirectional print mode
<b>ESC-u</b>	set bidirectional print mode
<b>ESC-W</b>	print double width
<b>ESC-w</b>	cancel double width print
<b>ESC-X</b>	set draft print mode
<b>ESC-x</b>	set NLQ print mode
<b>ESC--</b>	begin underscore
<b>ESC-_</b>	end underscore

**SEE ALSO**

*col*(1), *man*(1), *mm*(1), *nroff*(1), and *nterm*(5).

**NOTES**

It is strongly advised to use *col* with Epson FX printers in order to prevent misalignment of characters caused by reverse carriage motion.

**NAME**

*hfc\_ctl* - configure hardware flow control

**SYNOPSIS**

*/etc/hfc\_ctl* +*port*

*/etc/hfc\_ctl* -*port*

**DESCRIPTION**

*hfc\_ctl* sets the hardware flow control (HFC) on or off for a specified serial port.

The options are:

+*port* turn HFC on for *port*

-*port* turn HFC off for *port*

where *port* is the full pathname of a serial device file, i.e., */dev/tty00x*.

**EXAMPLES**

To turn HFC on for the built-in serial port

```
/etc/hfc_ctl +/dev/tty000
```

To turn HFC off for the expansion serial port 1

```
/etc/hfc_ctl -/dev/tty001
```

You can also create a daemons file in the */etc/daemons* directory that configures HFC each time the system boots. For example, to automatically set HFC on at boot-time for */dev/tty000*, put the line

```
/etc/hfc_ctl +/dev/tty000
```

in a file, say, */etc/daemons/tty000.init*, with execution permission.

**NAME**

*hplj* – filters *nroff*(1) output to Hewlett-Packard LaserJet printers.

**SYNOPSIS**

**nroff -Thpljr8 files | hplj | lp -dprinter\_R**

**DESCRIPTION**

*hplj* accepts *nroff* output as its input, and *hplj*'s output should be sent to *raw* lp. *hplj* uses the same syntax as *col*(1) for indicating special characters; it sets the eighth bit on any characters between the ASCII **SO** (\017) and **SI** (\016) characters, and discards **SO** and **SI** from its output.

The following translations are performed on input characters:

**SO**            set 8th bit on following characters

**SI**            stop setting 8th bit

**SEE ALSO**

*col*(1), *man*(1), *mm*(1), *nroff*(1), and *nterm*(5).

**NAME**

hyphen – find hyphenated words

**SYNOPSIS**

**hyphen** [ files ]

**DESCRIPTION**

*hyphen* finds all the hyphenated words ending lines in *files* and prints them on the standard output. If no arguments are given or if *hyphen* encounters `-`, it uses the standard input. Thus, *hyphen* may be used as a filter.

**EXAMPLE**

You would use the following command-line to proofread *nroff*'s hyphenation in *files*:

```
mm mm_options files | hyphen
```

**SEE ALSO**

man(1), mm(1), and nroff(1).

**BUGS**

*hyphen* can't cope with hyphenated italic (or underlined words); it frequently will either miss them altogether or mishandle them. *hyphen* occasionally gets confused but with no ill effects other than spurious extra output.

**NAME**

ismpx – return windowing terminal state

**SYNOPSIS**

ismpx [-s]

**DESCRIPTION**

The *ismpx* command reports whether its standard input is connected to a multiplexed *xt(7)* channel; i.e., whether it's running under *layers(1)* or not. It is useful for shell scripts that download programs to a windowing terminal or depend on screen size.

*ismpx* prints **yes** and returns **0** if invoked under *layers(1)*, and prints **no** and returns **1** otherwise.

**-s**        Do not print anything; just return the proper exit status.

**EXIT STATUS**

Returns **0** if invoked under *layers(1)*, **1** if not.

**SEE ALSO**

*layers(1)*, *jwin(1)*.  
*xt(7)* in the *System Administrator's Reference Manual*.

**EXAMPLE**

```
if ismpx -s
then
    jwin
fi
```

**NAME**

*jterm* – reset layer of windowing terminal

**SYNOPSIS**

*jterm*

**DESCRIPTION**

The *jterm* command is used to reset a layer of a windowing terminal after downloading a terminal program that changes the terminal attributes of the layer. It is useful only under *layers(1)*. In practice, it is most commonly used to restart the default terminal emulator after using an alternate one provided with a terminal-specific application package. For example, on the AT&T Teletype 5620 DMD terminal, after executing the *hp2621(1)* command in a layer, issuing the *jterm* command will restart the default terminal emulator in that layer.

**EXIT STATUS**

Returns **0** upon successful completion, **1** otherwise.

**NOTE**

The layer that is reset is the one attached to standard error; that is, the window you are in when you type the *jterm* command.

**SEE ALSO**

*layers(1)*.

**NAME**

*jwin* – print size of layer

**SYNOPSIS**

*jwin*

**DESCRIPTION**

*jwin* runs only under *layers(1)* and is used to determine the size of the layer associated with the current process. It prints the width and the height of the layer in bytes (number of characters across and number of lines, respectively). For bit-mapped terminals only, it also prints the width and height of the layer in bits.

**EXIT STATUS**

Returns **0** on successful completion, **1** otherwise.

**DIAGNOSTICS**

If *layers(1)* has not been invoked, an error message is printed:

```
jwin: not mpx
```

**NOTE**

The layer whose size is printed is the one attached to standard input; that is, the window you are in when you type the *jwin* command.

**SEE ALSO**

*layers(1)*.

**EXAMPLE**

```
jwin
bytes:  86 25
bits:   780 406
```

**NAME**

ktune – kernel tunable parameters

**SYNOPSIS**

```
ktune [options] [files]
ktune [ -d ] [ [ -r ] [ nbuf=n ] [ inode=n ] [ nfile=n ]
[ nproc=n ] [ ntext=n ] [ nclist=n ] [ npbuf=n ]
[ ncall=n ] [ nttyhog=n ] [ kern=filename]
```

**DESCRIPTION**

*Ktune* provides a way to change values of the following parameters which reside in the file 'filename' specified in the argument 'kern=filename'. If the argument 'kern=filename' is absent, the program modifies /unix.

The table below summarizes the parameters that can be set using *ktune*.

	Parameter	Minimum Value	Default
	nbuf	25	100
	ninode	80	400
	nfile	80	300
	nproc	30	100
	ntext	24	75
	nclist	32	150
	npbuf	4	16
	ncall	16	32
	nttyhog	0	1024
<b>nbuf</b>	Number of system buffers available. These buffers are used mostly by block device drivers for file system operations. Range: 25 up to system capacity.		
<b>ninode</b>	Number of memory-resident inodes that can be allocated at any time. The inode is the focus of all file activity in UNIX. There is a unique inode allocated for each open file, each current directory, each mounted-on file, text file, and the root. Range: 80 up to system capacity.		
<b>nfile</b>	Total number of files that can be opened on the system at any time. One file structure is allocated for each open/creat/pipe call. Note that while <b>nfile</b> controls the <i>total</i> number of files that can be open at any given time, another parameter, <b>nopen</b> , sets the number of files that can be open at any given time by any <i>single</i> process. <b>Nopen</b> is not tunable, and is currently set to 80. Range: 80 up to system capacity.		
<b>nproc</b>	Number of processes that can exist at any time. One process structure is allocated per active process, and it contains all the data about the process. Range: 30 up to system capacity.		
<b>ntext</b>	Number of text structures allocated in the kernel. One text structure is allocated per pure procedure on swap devices. Range: 24 up to system capacity.		

- nclist**      Number of clist buffers available. These buffers are used mostly by character device drivers for terminal I/O operations.  
Range: 32 up to system capacity.
- npbuf**      Number of buffer headers available in the raw I/O pool of headers.  
Range: 4 up to system capacity.
- ncall**      Number of callouts allowed in the kernel. When a process must be sure that it is awakened after a specific period of time, it calls the kernel timeout routine with a specified amount of time. The timeout routine places an entry in the callout table. **ncall** specifies the number of entries in the callout table.  
Range: 16 up to system capacity.
- nttyhog**    Maximum number of characters outstanding in the tty buffer for a given port before the system will flush that port's queue. If this value is set to 0, the system will no longer check for the maximum characters outstanding in the buffers. The **ttyhog** option keeps one port from using all the clist buffers, ensuring that each port has enough buffer space.  
Range: 0 up to 1024
- kern=**      If an argument 'kern=filename' is present, the program modifies file 'filename' instead of `/unix`.

*ktune* commands that list only some of the parameters cause only those parameters to change. An argument consisting of a dash (-) is taken to be a file name corresponding to the standard input. The *options* may appear in any order but must appear before the *files*.

The **-d** flag lists each parameter and the value which the kernel is currently using. Note that this might be different than the setting on the actual file on disk. Each parameter appears on a separate line, with the value preceded by a keyword (i.e., **ninode=200**). Input lists that list only some of the parameters cause only those parameters to change. This option displays the actual settings in use for the running kernel (not the settings stored on the disk). These settings may be lower than the disk settings due to small memory size.

The **-r** option resets parameters to their default values as specified in the `/etc/master` file.

There is a table called 'tuhi' which resides in the kernel. Tuning is accomplished by changing the parameters in this table on the disk, and requires the user to reboot.

All input parameters are checked against a set of minimum parameters. Any input with an error on any parameter results in no changes to any parameters. Input lists containing a value that violates these minimums result in no changes, and an error return.

The kernel boot routine is modified to provide for sanity checking on boot up to insure that enough memory is present for the values specified, and that the kernel virtual memory addressing limits are not violated. If the memory found is too small for the values in 'tuhi', the values in core (not on the disk) are adjusted downward until the resulting kernel runs on

the system being booted. If after ten refinements the values in 'tuh' are still too large, the default tuning is used.

**NAME**

*layers* – layer multiplexor for windowing terminals

**SYNOPSIS**

**layers** [-s] [-t] [-d] [-p] [-f *file*] [*layersys-prgm*]

**DESCRIPTION**

*layers* manages asynchronous windows (see *layers(5)*) on a windowing terminal. Upon invocation, *layers* finds an unused *xt(7)* channel group and associates it with the terminal line on its standard output. It then waits for commands from the terminal.

Command-line options:

- s Reports protocol statistics on standard error at the end of the session after you exit from *layers*. The statistics may be printed during a session by invoking the program *xst(1M)*.
- t Turns on *xt(7)* driver packet tracing, and produces a trace dump on standard error at the end of the session after you exit from *layers*. The trace dump may be printed during a session by invoking the program *xst(1M)*.
- d If a firmware patch has been downloaded, prints out the sizes of the text, data, and bss portions of the firmware patch on standard error.
- p If a firmware patch has been downloaded, prints the downloading protocol statistics and a trace on standard error.
- f *file* Starts *layers* with an initial configuration specified by *file*. Each line of the file represents a layer to be created, and has the following format:

```
origin_x  origin_y  corner_x  corner_y
command_list
```

The coordinates specify the size and position of the layer on the screen in the terminal's coordinate system. If all four are 0, the user must define the layer interactively. *command\_list*, a list of one or more commands, must be provided. It is executed in the newlayer using the user's shell (by executing: \$SHELL -i -c "*command\_list*"). This means that the last command should invoke a shell, such as */bin/sh*. (If the last command is not a shell, then, when the last command has completed, the layer will not be functional.)

*layersys-prgm*

A file containing a firmware patch that the *layers* command downloads to the terminal before layers are created and *command\_list* is executed.

Each layer is in most ways functionally identical to a separate terminal. Characters typed on the keyboard are sent to the standard input of the UNIX system process attached to the current layer (called the host process), and characters written on the standard output by the host process appear in that layer. When a layer is created, a separate shell is established and bound to the layer. If the environment variable SHELL is set, the user will get that shell; otherwise, */bin/sh* will be used. In order to enable communications with other users via *write(1)*, *layers* invokes the

command *relogin*(1M) when the first layer is created. *relogin*(1M) will reassign that layer as the user's logged-in terminal. An alternative layer can be designated by using *relogin*(1M) directly. *layers* will restore the original assignment on termination.

Layers are created, deleted, reshaped, and otherwise manipulated in a terminal-dependent manner. For instance, the AT&T Teletype 5620 DMD terminal provides a mouse-activated pop-up menu of layer operations. The method of ending a *layers* session is also defined by the terminal.

If a user wishes to take advantage of a terminal-specific application software package, the environment variable **DMD** should be set to the pathname of the directory where the package was installed. Otherwise **DMD** should not be set.

#### EXAMPLE

```
layers -f startup
```

where **startup** contains

```
8 8 700 200 date ; pwd ; exec $SHELL
8 300 780 850 exec $SHELL
```

#### NOTES

The *xt*(7) driver supports an alternate data transmission scheme known as ENCODING MODE. This mode makes *layers* operation possible even over data links which intercept control characters or do not transmit 8-bit characters. ENCODING MODE is selected either by setting a configuration option on your windowing terminal or by setting the environment variable **DMDLOAD** to the value *hex* before running *layers*:

```
export DMDLOAD; DMDLOAD=hex
```

If, after executing *layers -f file*, the terminal does not respond in one or more of the layers, often the last command in the *command-list* for that layer did not invoke a shell.

#### WARNING

To access this version of *layers*, make sure */usr/bin* appears before any other directory, such as *\$DMD/bin*, you have in your path that contains a *layers* program. (For information about defining the shell environmental variable **PATH** in your *profile*, see *profile*(4)) Otherwise, if there is a terminal-dependent version of *layers*, you may get it instead of the correct one.

When invoking *layers* with the **-s**, **-t**, **-d**, or **-p** options, it is best to redirect standard error to another file to save the statistics and tracing output (e.g., *layers -s 2>stats*); otherwise all or some of the output may be lost.

#### FILES

```
/dev/xt??[0-7]
/usr/lib/layersys/lsys.8;7;3
$DMD/lib/layersys/lsys.8;?;?
```

**SEE ALSO**

sh(1), write(1).

layers(5), libwindows(3X) in the *Programmer's Reference Manual*.

relogin(1M), xt(7), xts(1M), xtt(1M), wtinit(1M) in the *System Administrator's Reference Manual*.

**NAME**

layers – protocol used between host and windowing terminal under *layers*(1)

**SYNOPSIS**

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

**DESCRIPTION**

*layers* are asynchronous windows supported by the operating system in a windowing terminal. Communication between the UNIX system processes and terminal processes under *layers*(1) occurs via multiplexed channels managed by the respective operating systems using a protocol as specified in *xproto*(5).

The contents of packets transferring data between a UNIX system process and a layer are asymmetric. Data sent from the UNIX system to a particular terminal process is undifferentiated and it is up to the terminal process to interpret the contents of packets.

Control information for terminal processes is sent via channel 0. Process 0 in the windowing terminal performs the designated functions on behalf of the process connected to the designated channel. These packets take the form:

```
command, channel
```

except for *timeout* and *jagent* information which take the form

```
command, data...
```

The commands are the bottom eight bits extracted from the following *ioctl*(2) codes:

- JBOOT** Prepare to load a new terminal program into the designated layer.
- JTERM** Kill the downloaded layer program, and restore the default window program.
- JTIMO** Set the timeout parameters for the protocol. The data consist of two bytes: the value of the receive timeout in seconds, and the value of the transmit timeout in seconds.
- JTIMOM** Set the timeout parameters for the protocol. The data consist of four bytes in two groups: the value of the receive timeout in milliseconds (the low eight bits followed by the high eight bits) and the value of the transmit timeout (in the same format).
- JZOMBOOT** Like JBOOT, but do not execute the program after loading.
- JAGENT** Send a source byte string to the terminal agent routine and wait for a reply byte string to be returned.

The data are from a *bagent* structure (see *jagent*(5)) and consist of a one-byte size field followed by a two-byte agent command code and parameters. Two-byte integers transmitted as part of an agent command are sent with the high-order byte first. The response from the terminal is generally

identical to the command packet, with the two command bytes replaced by the return code: 0 for success, -1 for failure. Note that the routines in the *libwindows(3X)* library all send parameters in an *agentrect* structure. The agent command codes and their parameters are as follows:

<b>A_NEWLAYER</b>	followed by a two-byte channel number and a rectangle structure (four two-byte coordinates).
<b>A_CURRENT</b>	followed by a two-byte channel number.
<b>A_DELETE</b>	followed by a two-byte channel number.
<b>A_TOP</b>	followed by a two-byte channel number.
<b>A_BOTTOM</b>	followed by a two-byte channel number.
<b>A_MOVE</b>	followed by a two-byte channel number and a point to move to (two two-byte coordinates).
<b>A_RESHAPE</b>	followed by a two-byte channel number and the new rectangle (four two-byte coordinates).
<b>A_NEW</b>	followed by a two-byte channel number and a rectangle structure (four two-byte coordinates).
<b>A_EXIT</b>	no parameters needed.
<b>A_ROMVERSION</b>	no parameters needed. The response packet contains the size byte, two-byte return code, two unused bytes, and the parameter part of the terminal id string (e.g., "8;7;3").

Packets from the windowing terminal to the UNIX system all take the following form:

command, data...

The single-byte commands are as follows:

<b>C_SENDCHAR</b>	Send the next byte to the UNIX system process.
<b>C_NEW</b>	Create a new UNIX system process group for this layer. Remember the window size parameters for this layer. The data for this command is in the form described by the <i>jwinsize</i> structure. The size of the window is specified by two 2-byte integers, sent low byte first.
<b>C_UNBLK</b>	Unblock transmission to this layer. There is no data for this command.
<b>C_DELETE</b>	Delete the UNIX system process group attached to this layer. There is no data for

	this command.
<b>C_EXIT</b>	Exit. Kill all UNIX system process groups associated with this terminal and terminate the session. There is no data for this command.
<b>C_DEFUNCT</b>	Layer program has died, send a terminate signal to the UNIX system process groups associated with this terminal. There is no data for this command.
<b>C_SEDNCHARS</b>	The rest of the data are characters to be passed to the UNIX system process.
<b>C_RESHAPE</b>	The layer has been reshaped. Change the window size parameters for this layer. The data takes the same form as for the <b>C_NEW</b> command.

**SEE ALSO**

libwindows(3X), jagent(5), xproto(5).  
layers(1) in the *User's Reference Manual*.  
xt(7) in the *System Administrator's Reference Manual*.

**NAME**

*macref* – produces cross-reference listing of macro files

**SYNOPSIS**

*macref* [-t] [-s] [-n] [ - - ] *files*

**DESCRIPTION**

*macref* reads the named *files* (which are assumed to be *nroff*(1) or *troff* input) and produces a cross-reference listing of the symbols in the input.

A -t on the command line causes a macro table of contents to be printed. The option -s causes symbol-use statistics to be printed. The option -n causes one line to be printed for each reference to a symbol. The options may be grouped behind one -. You may use "--" to delimit the end of options. *macref* does not accept - as standard input.

The default output is a list of the symbols found in the input, each accompanied by a list of all references to that symbol. *macref* lists the symbols alphabetically in the leftmost column, with the references following to the right. Each reference is given in the form:

[ [(*NMname*)] *Mname*-] *type* *lnum* [#]

where the fields have the following meanings:

*Mname* the name of the macro within which the reference occurs. This field is missing if the reference occurs outside a macro. Any names listed in the *NMname* part are macros within which *Mname* is defined.

*type* the type associated, by context, with this occurrence of the symbol. The types may be:

r	request
m	macro
d	diversion
s	string
n	number register
p	parameter (e.g., \ \$x is a parameter reference to x. Note that parameters are never modified, and that the only valid parameter symbol names are 1, 2, ... 9).

*lnum* the line number on which the reference occurred.

# this reference modifies the value of the symbol.

Generated names are listed under the artificial symbol name “ ~sym”.

**SEE ALSO**

*man*(1), *mm*(1), *mm*(5), *man*(5), and *nroff*(1),

**NAME**

**man** – prints documents formatted with the man macros

**SYNOPSIS**

**man** [ options ] [ files ]

**DESCRIPTION**

Use *man* to format UNIX® manual pages using *nroff*(1) and the *man*(5) macro package. *man* has options to specify preprocessing by *tbl*(1) and/or *neqn*(1), and postprocessing by various terminal-oriented output filters. The proper pipelines and the required arguments and flags for *nroff* and *man* are generated, depending on the options that you select.

*Options* for *man* are given below. Any other arguments or flags (e.g., *-rC3*) are passed to *nroff* as appropriate. You may use such options in any order, but you must put them before the *files* arguments. If you do not specify arguments, *man* prints a list of its options.

*-Ttty\_type*

specifies the type of output terminal.

Here is a list of recognized values for *tty\_type*:

<b>2631</b>	Hewlett-Packard 2631 printer in regular mode
<b>2631-c</b>	Hewlett-Packard 2631 printer in compressed mode
<b>2631-e</b>	Hewlett-Packard 2631 printer in expanded mode
<b>300</b>	DASI-300 printer
<b>300-12</b>	DASI-300 terminal set to 12-pitch (12 characters per inch)
<b>300s</b>	DASI-300s printer (300S is a synonym)
<b>300s-12</b>	DASI-300s printer set to 12-pitch (12 characters per inch) (300S-12 is a synonym)
<b>37</b>	Teletype Model 37 terminal (default)
<b>382</b>	DTC-382
<b>4000a</b>	Trendata 4000a terminal (4000A is a synonym)
<b>450</b>	DASI-450 (Diablo Hyterm) printer
<b>450-12</b>	DASI-450 terminal set to 12-pitch (12 characters per inch)
<b>455</b>	AT&T 455 printer
<b>455-12</b>	AT&T 455 printer set to 12-pitch (12 characters per inch)
<b>455-15</b>	AT&T 455 printer set to 15-pitch (15 characters per inch)
<b>470</b>	AT&T 470 printer
<b>470-12</b>	AT&T 470 printer set to 12-pitch (12 characters per inch)
<b>470-17</b>	AT&T 470 printer set to 17-pitch (17 characters per inch)
<b>475</b>	AT&T 475 printer
<b>475-12</b>	AT&T 475 printer set to 12-pitch (12 characters per inch)
<b>475-17</b>	AT&T 475 printer set to 17-pitch (17 characters per inch)
<b>478</b>	AT&T 478 printer

<b>478-12</b>	AT&T 478 printer set to 12-pitch (12 characters per inch)
<b>478-18</b>	AT&T 478 printer set to 18-pitch (18 characters per inch)
<b>479</b>	AT&T 479 printer
<b>479-12</b>	AT&T 479 printer set to 12-pitch (12 characters per inch)
<b>479-18</b>	AT&T 479 printer set to 18-pitch (18 characters per inch)
<b>832</b>	Anderson Jacobson 832 terminal
<b>8510</b>	C.ITOH printer
<b>8510-12</b>	C.ITOH printer set to 12-pitch (12 characters per inch)
<b>8510-17</b>	C.ITOH printer set to 17-pitch (17 characters per inch)
<b>fx</b>	Epson FX-series printer
<b>fx-12</b>	Epson FX-series printer set to 12-pitch (12 characters per inch)
<b>fx-17</b>	Epson FX-series printer set to 17-pitch (17 characters per inch)
<b>hpljr8</b>	Hewlett-Packard LaserJet printer with Roman 8 cartridge
<b>ibm</b>	IBM graphics printer
<b>ibm-17</b>	IBM graphics printer set to 17-pitch (17 characters per inch)
<b>tn300</b>	GE Terminet 300 terminal
<b>X</b>	printers equipped with TX print train
<b>lp</b>	generic name for printers that can underline and tab. <b>lp</b> is the default device for <i>man</i> . All text sent to <b>lp</b> requiring reverse linefeeds, such as those having tables, must be processed with <i>col(1)</i> , invoked with <i>man's</i> <b>-c</b> option. If you do not use this option, <i>man</i> uses the value of the shell variable <b>\$TERM</b> from the environment (see <i>profile(4)</i> and <i>environ(5)</i> ) as the value of <i>tty_type</i> if <b>\$TERM</b> is set; otherwise, <i>man</i> uses <b>450</b> as the value of <i>tty_type</i> . If you specify several terminal types, the last one takes precedence.

(Check with your system administrator for a list of locally supported devices.)

- 12** indicates that the document is to be produced in 12-pitch. You may use this option when **\$TERM** is set to one of **300**, **300s**, and **450**. (You must manually set the pitch switch on the DASI 300 and 300s terminals to **12** if you use this option.)
- c** causes *man* to invoke *col(1)*; note that *col* is invoked automatically by *man* unless *tty\_type* is one of **300**, **300s**, **450**, **455**, **37**, **4000a**, **382**, **hpljr8**, and **X**.
- e** causes *man* to invoke *neqn*; also causes *neqn* to read the */usr/pub/eqnchar* file (see *eqnchar(5)*).
- t** causes *man* to invoke *tbl(1)*.

- E invokes the `-e` option of *nroff*.
- D sends a copy of the UNIX command line that *man* produces to invoke *nroff* to the standard error for debugging purposes.

As an example, assume that the shell variable `$TERM` is set in the environment to `450`. The two command lines below are then equivalent:

```
man -t -rC3 -12 files
tbl files | nroff -man -T450-12 -h -rC3
```

*man* reads the standard input when you specify `-` instead of any *files*. (Mentioning other files together with `-` leads to undesired results.) This option allows you to use *man* as a filter, for example:

```
cat files | man -
```

## HINTS

1. *man* invokes the appropriate *nroff* output filter for *tty\_type* `470`, `475`, `478`, `479`, `fx`, `ibm`, and `hpljr8`.
2. *man* invokes *nroff* with the `-h` flag unless *tty\_type* is `455` or `hpljr8`. With this flag, *nroff* assumes that the terminal has tabs set every 8 character positions.
3. Use the `-olist` option of *nroff* to specify ranges of pages to be output. Note, however, that if you invoke *man* with one or more of the `-e`, `-t`, and `-` options, together with the `-olist` option of *nroff*, you may cause a harmless "broken pipe" diagnostic if you do not specify the last page of the document in *list*.
4. If you use the `-s` option of *nroff* (to stop between pages of output), use line-feed (rather than return or new-line) to restart the output. The `-s` option of *nroff* does not work with the `-c` option of *man*, or if *man* automatically invokes *col* (see `-c` option above).
5. If you lie to *man* about the kind of terminal its output will be printed on, you will get (often subtle) garbage; however, if you are redirecting output into a file, use the `-T37` option, and then use the appropriate terminal filter when you actually print that file.

## FILES

`/usr/pub/terminals` list of supported terminals

## DIAGNOSTICS

"man: no input file"  
if none of the arguments is a readable file and you do not use *man* as a filter.

## SEE ALSO

`checkmm(1)`, `col(1)`, `env(1)`, `greek(1)`, `mm(1)`, `mm(5)`, `neqn(1)`, `nroff(1)`, `profile(4)`, `tbl(1)`, and `term(5)`;

**NAME**

**man** – macros for formatting manual page entries

**SYNOPSIS**

**nroff** *-man* files

**troff** *-man* [ *-rs1* ] files

**DESCRIPTION**

These *nroff*(1)/*troff* macros are used to lay out the format of UNIX® manual pages. The default page size is 8.5"×11", with a 6.5"×10" text area; the *-rs1* option reduces these dimensions to 6"×9" and 4.75"×8.375", respectively; this option also reduces the default type size from 10-point to 9-point, and the vertical line spacing from 12-point to 10-point. The *-rv2* option may be used to set certain parameters to values appropriate for certain Versatec printers: it sets the line length to 82 characters, the page length to 84 lines, and it inhibits underlining.

Any *text* argument below may be one to six "words." Double quotes ("" ) may be used to include blanks in a "word." If *text* is empty, the special treatment is applied to the next line that contains text to be printed. For example, *.I* may be used to italicize a whole line, or *.SM* followed by *.B* to make small bold text. By default, hyphenation is turned off for *nroff*, but remains on for *troff*.

Type font and size are reset to default values before each paragraph and after processing font- and size-setting macros, for example, *.I*, *.RB*, *.SM*. Tab stops are neither used nor set by any macro except *.DT* and *.TH*.

Default units for indents *in* are ens. When *in* is omitted, the previous indent is used. This remembered indent is set to its default value (7.2 ens in *troff*, 5 ens in *nroff*) by *.TH*, *.P*, and *.RS*, and restored by *.RE*.

**.TH** *t s c n* Set the title and entry heading; *t* is the title, *s* is the section number, *c* is extra commentary, e.g., "local", *n* is new manual name. Invokes *.DT* (see below).

**.SH** *text* Place subhead *text*, for example, SYNOPSIS, here.

**.SS** *text* Place sub-subhead *text*, for example, Options, here.

**.B** *text* Make *text* bold.

**.I** *text* Make *text* italic.

**.SM** *text* Make *text* 1 point smaller than default point size.

**.RI** *a b* Concatenate roman *a* with italic *b*, and alternate these two fonts for up to six arguments. Similar macros alternate between any two of roman, italic, and bold:

**.IR .RB .BR .IB .BI**

**.PP** Begin a paragraph with normal font, point size, and indent. *.PP* is a synonym for the *mm*(5) macro *.P*.

**.HP** *in* Begin paragraph with hanging indent.

**.TP** *in* Begin indented paragraph with hanging tag. The next line that contains text to be printed is taken as the tag. If the tag does not fit, it is printed on a separate line.

**.IP** *t in* Same as *.TP in* with tag *t*; often used to get an indented paragraph without a tag.

**.RS** *in* Increase relative indent (initially zero). Indent all output an extra *in* units from the current left margin.

**.RE** *k* Return to the *k*th relative indent level (initially, *k*=1; *k*=0 is equivalent to *k*=1); if *k* is omitted, return to the most recent lower indent level.

- .PM** *m* Produces proprietary markings which depend on value of *m*.  
*m* of "BP", "PM1", "PM5", or "PM6" produces this message:  
**AT&T - PROPRIETARY**  
 Not for use or disclosure outside of  
 AT&T or its affiliated companies  
 except under written agreement.
- m* of "P" or "PM2" produces this message:  
**AT&T - PROPRIETARY (RESTRICTED)**  
 Information herein is solely for persons having a need-  
 to-know as authorized by the Director of the originating  
 organization. Reproduction is prohibited.
- m* of "BR", "PM3", or "PM4" produces this message:  
**AT&T - PROPRIETARY (REGISTERED)**  
 Information herein is solely for persons having a need-  
 to-know as authorized by the Director of the originating  
 organization. Reproduction is prohibited. This document  
 is subject to cover sheet restrictions.
- .DT** Restore default tab settings (every 7.2 ens in *troff*, 5 ens in  
*nroff*).
- .PD** *v* Set the interparagraph distance to *v* vertical spaces. If *v* is  
 omitted, set the interparagraph distance to the default value  
 (0.4*v* in *troff*, 1*v* in *nroff*).

The following *strings* are defined:

- \\*R** ® in *troff*, "Registered" symbol (the "\(rg" special character)  
 in *nroff*.
- \\*S** Change to default type size.
- \\*(Tm** Trademark indicator.

The following *number registers* are given default values by **.TH**:

- IN** Left margin indent relative to subheads (default is 7.2 ens in  
*troff*, 5 ens in *nroff*).
- LL** Line length including **IN**.
- PD** Current interparagraph distance.

## EXAMPLES

The *man* macros are provided to process manual pages already on-line at a given location and to enable users to make their own manual pages. The preceding section demonstrated the usage of the macros themselves; the following section provides examples of command lines typically used to process the completed files.

*man* macros are designed to run with either *nroff* or *troff*. The first command line will process *files* using only macros and *nroff* requests:

```
nroff -Tlp -man files | lp
```

*Files* is piped to the local line printer, **lp**.

The next command line will process *files* containing tables as well as macros and *nroff* requests:

```
tbl files | nroff -Tlp -man | col | lp
```

Notice that before it is sent to the line printer, the output is first filtered through *col(1)*, to process the reverse line feeds used by *tbl(1)*.

The final example is a command line that processes an unusual manual page, one using *pic* and *grap*. If the manual pages created with *man* are intended for an on-line facility, components requiring *troff*, such as *grap* or *pic*, should be avoided since the average installation of terminals will not be able to process typeset documents.

`grap files | pic | tbl | troff -Taps -man | typesetter`

*grap* precedes *pic* because it is a preprocessor to *pic*; the reverse order will not format correctly. *Files* contains one or more tables, requiring *tbl*, but *col* is no longer necessary because typeset documents do not use reverse line feeds with which to make tables. The `-T` option for specifying the output device (Terminal type) takes the argument `aps` here, readying the document for processing on the APS-5 phototypesetter.

#### CAVEATS

Special macros, strings, and number registers exist, internal to *man*, in addition to those mentioned above. Except for names predefined by *nroff* and number registers `d`, `m`, and `y`, all such internal names are of the form `XA`, where `X` is one of `)`, `]`, and `}`, and `A` stands for any alphanumeric character. User defined macros should avoid these naming conventions.

The macro package itself uses only the roman font (so that one can replace, for example, the bold font by the constant-width font (CW). Of course, if the input text of an entry contains requests for other fonts (for example, `.I`, `.RB`, `\II`), the corresponding fonts must be mounted.

#### FILES

`/usr/lib/tmac/tmac.an`  
`/usr/lib/macros/an`

#### SEE ALSO

`man` (1), `neqn` (1), `nroff` (1), `tbl` (1), and `tc` (1),

#### BUGS

If the argument to `.TH` contains *any* blanks and is *not* enclosed by double quotes ("`"`), there will be strange irregular dots on the output.

**NAME**

**mm** – prints documents formatted with the mm macros

**SYNOPSIS**

**mm** [ options ] [ files ]

**osdd** [ options ] [ files ]

**DESCRIPTION**

Use *mm* to format documents using *nroff*(1) and the *mm*(5) text-formatting macro package. *mm* has options to specify preprocessing by *tbl*(1) and/or *neqn*(1), and postprocessing by various terminal-oriented output filters. The proper pipelines and the required arguments and flags for *nroff* and *mm* are generated, depending on the options that you select.

*Osdd* is equivalent to the command *mm -mosd*.

*Options* for *mm* are given below. Any other arguments or flags (e.g., *-rC3*) are passed to *nroff* as appropriate. You may use such options in any order, but you must put them before the *files* arguments. If you do not specify arguments, *mm* prints a list of its options.

*-Ttty\_type*

specifies the type of output terminal.

Here is a list of recognized values for *tty\_type*:

<b>2631</b>	Hewlett-Packard 2631 printer in regular mode
<b>2631-c</b>	Hewlett-Packard 2631 printer in compressed mode
<b>2631-e</b>	Hewlett-Packard 2631 printer in expanded mode
<b>300</b>	DASI-300 printer
<b>300-12</b>	DASI-300 terminal set to 12-pitch (12 characters per inch)
<b>300s</b>	DASI-300s printer (300S is a synonym)
<b>300s-12</b>	DASI-300s printer set to 12-pitch (12 characters per inch) (300S-12 is a synonym)
<b>37</b>	Teletype Model 37 terminal (default)
<b>382</b>	DTC-382
<b>4000a</b>	Trendata 4000a terminal (4000A is a synonym)
<b>450</b>	DASI-450 (Diablo Hyterm) printer
<b>450-12</b>	DASI-450 terminal set to 12-pitch (12 characters per inch)
<b>455</b>	AT&T 455 printer
<b>455-12</b>	AT&T 455 printer set to 12-pitch (12 characters per inch)
<b>455-15</b>	AT&T 455 printer set to 15-pitch (15 characters per inch)
<b>470</b>	AT&T 470 printer
<b>470-12</b>	AT&T 470 printer set to 12-pitch (12 characters per inch)
<b>470-17</b>	AT&T 470 printer set to 17-pitch (17 characters per inch)
<b>475</b>	AT&T 475 printer
<b>475-12</b>	AT&T 475 printer set to 12-pitch (12 characters per inch)

<b>475-17</b>	AT&T 475 printer set to 17-pitch (17 characters per inch)
<b>478</b>	AT&T 478 printer
<b>478-12</b>	AT&T 478 printer set to 12-pitch (12 characters per inch)
<b>478-18</b>	AT&T 478 printer set to 18-pitch (18 characters per inch)
<b>479</b>	AT&T 479 printer
<b>479-12</b>	AT&T 479 printer set to 12-pitch (12 characters per inch)
<b>479-18</b>	AT&T 479 printer set to 18-pitch (18 characters per inch)
<b>832</b>	Anderson Jacobson 832 terminal
<b>8510</b>	C.ITOH printer
<b>8510-12</b>	C.ITOH printer set to 12-pitch (12 characters per inch)
<b>8510-17</b>	C.ITOH printer set to 17-pitch (17 characters per inch)
<b>fx</b>	Epson FX-series printer
<b>fx-12</b>	Epson FX-series printer set to 12-pitch (12 characters per inch)
<b>fx-17</b>	Epson FX-series printer set to 17-pitch (17 characters per inch)
<b>hpljr8</b>	Hewlett-Packard LaserJet printer with Roman 8 cartridge
<b>ibm</b>	IBM graphics printer
<b>ibm-17</b>	IBM graphics printer set to 17-pitch (17 characters per inch)
<b>tn300</b>	GE Terminet 300 terminal
<b>X</b>	printers equipped with TX print train
<b>lp</b>	generic name for printers that can underline and tab. <b>lp</b> is the default device for <i>mm</i> . All text sent to <b>lp</b> requiring reverse linefeeds, such as those having tables, must be processed with <i>col(1)</i> , invoked with <i>mm's-c</i> option. If you do not use this option, <i>mm</i> uses the value of the shell variable <b>\$TERM</b> from the environment (see <i>profile(4)</i> and <i>environ(5)</i> ) as the value of <i>tty_type</i> if <b>\$TERM</b> is set; otherwise, <i>mm</i> uses <b>450</b> as the value of <i>tty_type</i> . If you specify several terminal types, the last one takes precedence.

(Check with your system administrator for a list of locally supported devices.)

- 12** indicates that the document is to be produced in 12-pitch. You may use this option when **\$TERM** is set to one of **300**, **300s**, and **450**. (You must manually set the pitch switch on the DASI 300 and 300s terminals to **12** if you use this option.)
- c** causes *mm* to invoke *col(1)*; note that *col* is invoked automatically by *mm* unless *tty\_type* is one of **300**, **300s**, **450**, **455**, **37**, **4000a**, **382**, **hpljr8**, and **X**.

- e** causes *mm* to invoke *neqn*; also causes *neqn* to read the */usr/pub/eqnchar* file (see *eqnchar(5)*).
- t** causes *mm* to invoke *tbl*.
- E** invokes the **-e** option of *nroff*.
- D** sends a copy of the UNIX command line that *mm* produces to invoke *nroff* to the standard error for debugging purposes.

As an example, assume that the shell variable `$TERM` is set in the environment to `450`. The two command lines below are then equivalent:

```
mm -t -rC3 -12 files
tbl files | nroff -mm -T450-12 -h -rC3
```

*mm* reads the standard input when you specify **-** instead of any *files*. (Mentioning other files together with **-** leads to undesired results.) This option allows you to use *mm* as a filter, for example:

```
cat files | mm -
```

## HINTS

1. *mm* invokes the appropriate *nroff* output filter for *tty\_type* **470**, **475**, **478**, **479**, **fx**, **ibm**, and **hpljr8**.
2. *mm* invokes *nroff* with the **-h** flag unless *tty\_type* is **455** or **hpljr8**. With this flag, *nroff* assumes that the terminal has tabs set every 8 character positions.
3. Use the **-olist** option of *nroff* to specify ranges of pages to be output. Note, however, that if you invoke *mm* with one or more of the **-e**, **-t**, and **-** options, *together* with the **-olist** option of *nroff*, you may cause a harmless "broken pipe" diagnostic if you do not specify the last page of the document in *list*.
4. If you use the **-s** option of *nroff* (to stop between pages of output), use line-feed (rather than return or new-line) to restart the output. The **-s** option of *nroff* does not work with the **-c** option of *mm*, or if *mm* automatically invokes *col* (see **-c** option above).
5. If you lie to *mm* about the kind of terminal its output will be printed on, you will get (often subtle) garbage; however, if you are redirecting output into a file, use the **-T37** option, and then use the appropriate terminal filter when you actually print that file.

## FILES

*/usr/pub/terminals* list of supported terminals

## DIAGNOSTICS

"mm: no input file"  
if none of the arguments is a readable file and you do not use *mm* as a filter.

## SEE ALSO

*checkmm(1)*, *col(1)*, *env(1)*, *greek(1)*, *man(1)*, *neqn(1)*, *nroff(1)*, *tbl(1)*, *profile(4)*, *mm(5)*, and *term(5)*.

**NAME**

**mm** – the mm macro package for formatting documents

**SYNOPSIS**

```
mm [ options ] [ files ]  
nroff [ options ] -mm [ files ]  
troff [ options ] -mm [ files ]
```

**DESCRIPTION**

This package provides a formatting capability for a very wide variety of documents. The manner in which you type and edit a document is essentially independent of whether the document is to be eventually formatted at a terminal or is to be phototypeset.

**FILES**

```
/usr/lib/tmac/tmac.m      pointer to the mm package  
/usr/lib/macros/mm[nt]  the mm package
```

**SEE ALSO**

**mm(1)** and **nroff(1)**.  
*MM—Memorandum Macros* by D.~W. Smith and J.~R. Mashey.  
*Typing Documents with MM* by D.~W. Smith and E.~M. Piskorik.

**NAME**

**mptx** – the macro package for formatting a permuted index

**SYNOPSIS**

**nroff** **-mptx** [ options ] files | printer

**troff** **-mptx** [ options ] files | typesetter

**DESCRIPTION**

This package provides a definition for the **.xx** macro used for formatting a permuted index as produced by *ptx*(1). This package does not provide any other formatting capabilities such as headers and footers. If these or other capabilities are required, the *mptx* macro package may be used in conjunction with the *mm*(5) macro package. In this case, the **-mptx** option must be invoked after the **-mm** call. For example,

**nroff -mm -mptx** files | printer

or

**mmt -mptx** files | typesetter

**FILES**

**/usr/lib/tmac/tmac.ptx** pointer to the macro package

**/usr/lib/macros/ptx** macro package

**SEE ALSO**

*mm*(1), *mm*(5), *nroff*(1), and *ptx*(1).

**NAME**

**ndx** – create a subject-page index for a document

**SYNOPSIS**

**ndx** [ subfile ] "*formatter command line*"

**DESCRIPTION**

*ndx*, given a list of subjects (*subfile*), searches a specified document and writes a subject-page index to the standard output.

*subfile* is the list of subjects to be included in the index. Each subject must begin on a new line and have the following format:

```
word1 [word2 ...] [, wordk ...]
```

For example,

```
printed circuit boards
arrays
arrays, dynamic storage
Smith, W. P.
printed circuit boards, channel-oriented multi-layer
Aranoff
University of Illinois
PL/1
```

The subject must start in column 1.

The syntax for the formatter command line is

```
formatter [ options ] files
```

It is the command that will be used to create the final form of the document. The following are examples of valid formatter command lines:

```
mm -Tlp files
nroff -mm -Tlp -rW60 files
troff -rB2 -Taps -r01.5i files
```

For more information on the formatter command line, see, *mm*(1), and *nroff*(1).

The document must include formatting commands for *mm*, *nroff*, or *troff*. The formatter command line tells *ndx* whether *troff*, *nroff*, *mm* or *mmt* would be used to produce the final version of the document.

**troff** or **mmt**

specifies *troff* as the formatting program.

**nroff** or **mm**

specifies *nroff* as the formatting program

The *options* are those that would be given to the *troff*, *nroff*, *mm* or *mmt* command in printing the final form of the document, and are necessary to determine the correct page numbers for subjects as they are located in the document. *ndx* does not actually cause the final version of the document to be printed. The author must create the document separately. The indexer, of course, should not be used until the document is complete and no further

changes are expected.

#### EXAMPLES

The command

```
ndx subjfile "nroff -mm -rW70 files" > indexfile
```

would produce a subject-page index for the document *files* and take its subjects from the list, *subjfile*. The page numbers would correspond to the document produced by

```
nroff -mm -rW70 files
```

The command

```
ndx subjfile "mm -rW60 -rN2 -rO0 ch1 ch2 ch3" > indexfile
```

would produce a subject-page index for the documents *ch1*, *ch2*, and *ch3*. The page numbers would correspond to the documents produced by

```
mm -rW60 -rN2 -rO0 ch1 ch2 ch3
```

The command

```
ndx subjfile "troff -rB2 -rW5i -rO1.5i -mm files" > indexfile
```

would produce a subject-page index for the document *files*, and the page numbers would correspond to the document produced by

```
troff -rB2 -rW5i -rO1.5i -mm files
```

#### SEE ALSO

*mm*(1), *nroff*(1), and *subj*(1).

**NAME**

neqn – format mathematical text for nroff

**SYNOPSIS**

neqn [ -dxy ] [ -pn ] [ -sn ] [ -fn ] [ files ]

**DESCRIPTION**

*Neqn* is an *nroff*(1) preprocessor for typing mathematical text on typewriter-like terminals. Usage is almost always:

```
neqn files | nroff
```

or equivalent.

If no files are specified (or if `-` is specified as the last argument), these programs read the standard input. A line beginning with `.EQ` marks the start of an equation; the end of an equation is marked by a line beginning with `.EN`. Neither of these lines is altered, so they may be defined in macro packages to get centering, numbering, etc. It is also possible to designate two characters as *delimiters*; subsequent text between delimiters is then treated as *neqn* input. Delimiters may be set to characters *x* and *y* with the command-line argument `-dxy` or (more commonly) with `delim xy` between `.EQ` and `.EN`. The left and right delimiters may be the same character; the dollar sign is often used as such a delimiter. Delimiters are turned off by `delim off`. All text that is neither between delimiters nor between `.EQ` and `.EN` is passed through untouched.

Tokens within *neqn* are separated by spaces, tabs, new-lines, braces, double quotes, tildes, and circumflexes. Braces `{ }` are used for grouping; generally speaking, anywhere a single character such as *x* could appear, a complicated construction enclosed in braces may be used instead. Tilde (`~`) represents a full space in the output, circumflex (`^`) half as much.

Subscripts and superscripts are produced with the keywords `sub` and `sup`. Thus `x sub j` makes  $x_j$ , `a sub k sup 2` produces  $a_k^2$ , while  $e^{x^2+y^2}$  is made with `e sup {x sup 2 + y sup 2}`. Fractions are made with `over`: `a over b` yields  $\frac{a}{b}$ ; `sqrt` makes square roots: `1 over sqrt {ax sup 2+bx+c}` results

in  $\frac{1}{\sqrt{ax^2+bx+c}}$ .

The keywords `from` and `to` introduce lower and upper limits:  $\lim_{n \rightarrow \infty} \sum_0^n x_i$

is made with `lim from {n -> inf} sum from 0 to n x sub i`. Left and right brackets, braces, etc., of the right height are made with `left` and `right`:

`left [ x sup 2 + y sup 2 over alpha right ] ~ ~ 1` produces

$\left[ x^2 + \frac{y^2}{\alpha} \right] = 1$ . Legal characters after `left` and `right` are braces, brackets,

bars, `c` and `f` for ceiling and floor, and `"` for nothing at all (useful for a right-side-only bracket). A *left thing* need not have a matching *right thing*.

Vertical piles of things are made with `pile`, `lpile`, `cpile`, and `rpile`:

`pile {a above b above c}` produces  $\begin{matrix} a \\ b \\ c \end{matrix}$ . Piles may have arbitrary numbers

of elements; `lpile` left-justifies, `pile` and `cpile` center (but with different

vertical spacing), and **rpile** right justifies. Matrices are made with **matrix**:

*matrix { lcol { x sub i above y sub 2 } ccol { 1 above 2 } }*

produces 
$$\begin{matrix} x_i & 1 \\ y_2 & 2 \end{matrix}$$
. In addition, there is **rcol** for a right-justified column.

Diacritical marks are made with **dot**, **dotdot**, **hat**, **tilde**, **bar**, **vec**, **dyad**, and **under**: *x dot = f(t) bar* is  $\bar{x} = \overline{f(t)}$ , *y dotdot bar ~ = ~ n under* is  $\bar{y} = \underline{n}$ , and *x vec ~ = ~ y dyad* is  $\vec{x} = \overleftarrow{y}$ .

Point sizes and fonts can be changed with **size n** or **size ±n**, **roman**, **italic**, **bold**, and **font n**. Point sizes and fonts can be changed globally in a document by **gsize n** and **gfont n**, or by the command-line arguments **-sn** and **-fn**.

Normally, subscripts and superscripts are reduced by 3 points from the previous size; this may be changed by the command-line argument **-pn**.

Successive display arguments can be lined up. Place **mark** before the desired lineup point in the first equation; place **lineup** at the place that is to line up vertically in subsequent equations.

Shorthands may be defined or existing keywords redefined with **define**:

*define thing % replacement %*

defines a new token called *thing* that will be replaced by *replacement* whenever it appears thereafter. The % may be any character that does not occur in *replacement*.

Keywords such as **sum** ( $\Sigma$ ), **int** ( $\int$ ), **inf** ( $\infty$ ), and shorthands such as **>=** ( $\geq$ ), **!=** ( $\neq$ ), and **->** ( $\rightarrow$ ) are recognized. Greek letters are spelled out in the desired case, as in **alpha** ( $\alpha$ ), or **GAMMA** ( $\Gamma$ ). Mathematical words such as **sin**, **cos**, and **log** are made Roman automatically. *Troff*(1) four-character escapes such as **\(dd** ( $\ddagger$ ) and **\(bs** ( $\text{\textcircled{#}}$ ) may be used anywhere. Strings enclosed in double quotes ("...") are passed through untouched; this permits keywords to be entered as text, and can be used to communicate with *nroff* when all else fails. Full details are given in the manual cited below.

#### SEE ALSO

*Typesetting Mathematics—User's Guide* by B. W. Kernighan and L. L. Cherry.

**cw**(1), **man**(1), **mm**(1), **nroff**(1), **tbl**(1), **eqnchar**(5), **mm**(5), and **man**(5).

#### BUGS

To embolden digits, parentheses, etc., it is necessary to quote them, as in **bold "12.3"**.

**NAME**

`nroff` – text formatting language

**SYNOPSIS**

`nroff` [ options ] [ files ] | printer

**DESCRIPTION**

`nroff` formats text contained in *files* (standard input by default) for printing on typewriter-like devices and line printers.

An argument consisting of a minus (-) is taken to be a file name corresponding to the standard input. The *options*, which may appear in any order, but must appear before the *files*, are,

- `-olist` Print only pages whose page numbers appear in the *list* of numbers and ranges, separated by commas. A range *N* – *M* means pages *N* through *M*; an initial *-N* means from the beginning to page *N*; and a final *N-* means from *N* to the end. (See **BUGS** below.)
- `-nN` Number first generated page *N*.
- `-sN` Stop every *N* pages. `nroff` will halt *after* every *N* pages (default *N*=1) to allow paper loading or changing, and will resume upon receipt of a line-feed or new-line (new-lines do not work in pipelines, e.g., with `mm(1)`). This option does not work if the output of `nroff` is piped through `col(1)`. When `nroff` halts between pages, an ASCII BEL is sent to the terminal.
- `-raN` Set register *a* (which must have a one-character name) to *N*.
- `-i` Read standard input after *files* are exhausted.
- `-q` Invoke the simultaneous input-output mode of the `.rd` request.
- `-z` Print only messages generated by `.tm` (terminal message) requests.
- `-mname` Prepend to the input *files* the macro file `/usr/lib/tmac/tmac.name`.
- `-Ttty_type` Prepare output for specified terminal. Known *tty\_types* are

<b>2631</b>	Hewlett-Packard 2631 printer in regular mode
<b>2631-c</b>	Hewlett-Packard 2631 printer in compressed mode
<b>2631-e</b>	Hewlett-Packard 2631 printer in expanded mode
<b>300</b>	DASI-300 printer
<b>300-12</b>	DASI-300 terminal set to 12-pitch (12 characters per inch)
<b>300s</b>	DASI-300s printer (300S is a synonym)
<b>300s-12</b>	DASI-300s printer set to 12-pitch (12 characters per inch) (300S-12 is a synonym)
<b>37</b>	Teletype Model 37 terminal (default)
<b>382</b>	DTC-382
<b>4000a</b>	Trendata 4000a terminal (4000A is a synonym)
<b>450</b>	DASI-450 (Diablo Hyterm) printer
<b>450-12</b>	DASI-450 terminal set to 12-pitch (12 characters per inch)
<b>455</b>	AT&T 455 printer

455-12	AT&T 455 printer set to 12-pitch (12 characters per inch)
455-15	AT&T 455 printer set to 15-pitch (15 characters per inch)
470	AT&T 470 printer
470-12	AT&T 470 printer set to 12-pitch (12 characters per inch)
470-17	AT&T 470 printer set to 17-pitch (17 characters per inch)
475	AT&T 475 printer
475-12	AT&T 475 printer set to 12-pitch (12 characters per inch)
475-17	AT&T 475 printer set to 17-pitch (17 characters per inch)
478	AT&T 478 printer
478-12	AT&T 478 printer set to 12-pitch (12 characters per inch)
478-18	AT&T 478 printer set to 18-pitch (18 characters per inch)
479	AT&T 479 printer
479-12	AT&T 479 printer set to 12-pitch (12 characters per inch)
479-18	AT&T 479 printer set to 18-pitch (18 characters per inch)
832	Anderson Jacobson 832 terminal
8510	C.ITOH printer
8510-12	C.ITOH printer set to 12-pitch (12 characters per inch)
8510-17	C.ITOH printer set to 17-pitch (17 characters per inch)
fx	Epson FX-series printer
fx-12	Epson FX-series printer set to 12-pitch (12 characters per inch)
fx-17	Epson FX-series printer set to 17-pitch (17 characters per inch)
hpljr8	Hewlett-Packard LaserJet printer with Roman 8 cartridge
ibm	IBM graphics printer
ibm-17	IBM graphics printer set to 17-pitch (17 characters per inch)
lp	generic name for printers that can underline and tab. (All text sent to <i>lp</i> requiring reverse line feeds, such as those having tables, must be processed with <i>col(1)</i> ).
tn300	GE Terminet 300 terminal
X	printers equipped with TX print train
-e	Produce equally-spaced words in adjusted lines, using the full resolution of the particular terminal.
-h	Use output tabs during horizontal spacing to speed output and reduce output character count. Tab settings are assumed to be every 8 nominal character widths.
-un	Set the emboldening factor (number of character over-strikes) for the third font position (bold) to <i>n</i> , or to zero if <i>n</i> is missing.

**FILES**

**/usr/lib/tmac/tmac.\*** pointers to standard macro files  
**/usr/lib/macros/\*** standard macro files  
**/usr/lib/nterm/\*** terminal driving tables for *nroff*  
**/usr/pub/terminals** list of supported terminals

**SEE ALSO**

*col*(1), *greek*(1), *man*(1), *man*(5), *mm*(1), *mm*(5), *mptx*(5), *neqn*(1), *nterm*(5), and *tbl*(1).

**BUGS**

*nroff* believes in Eastern Standard Time; as a result, depending on the time of the year and on your local time zone, the date that *nroff* generates may be off by one day from your idea of what the date is.

When *nroff* is used with the *-olist* option inside a pipeline (e.g., with one or more of *neqn*(1), and *tbl*(1)), it may cause a harmless "broken pipe" diagnostic if the last page of the document is not specified in *list*.

**NAME**

nterm – terminal driving tables for nroff

**DESCRIPTION**

*nroff*(1) uses driving tables to customize its output for various types of output devices, such as printing terminals, special word-processing terminals (such as Diablo, Qume, or NEC Spinwriter mechanisms), or special output filter programs. These driving tables are written as ASCII files, and are installed in `/usr/lib/nterm/tab.name`, where *name* is the name for that terminal type as given in *term*(5).

The first line of a driving table should contain the name of the terminal: simply a string with no embedded white space. ‘‘White space’’ means any combination of spaces, tabs and new-lines. The next part of the driver table is structured as follows:

```

bset [integer] (not supported in all versions of nroff)
breset [integer] (not supported in all versions of nroff)
Hor [integer]
Vert [integer]
Newline [integer]
Char [integer]
Em [integer]
Halfline [integer]
Adj [integer]
twinit [character string]
twrest [character string]
twnl [character string]
hhr [character string]
hlf [character string]
flr [character string]
bdon [character string]
bdoff [character string]
iton [character string]
itoff [character string]
ploton [character string]
plotoff [character string]
up [character string]
down [character string]
right [character string]
left [character string]

```

The meanings of these fields are as follows:

<b>bset</b>	bits to set in the <code>c_oflag</code> field of the <code>termio</code> structure before output.
<b>breset</b>	bits to reset in the <code>c_oflag</code> field of the <code>termio</code> structure before output.
<b>Hor</b>	horizontal resolution in units of 1/240 of an inch.
<b>Vert</b>	vertical resolution in units of 1/240 of an inch.
<b>Newline</b>	space moved by a newline (linefeed) character in units of 1/240 of an inch.
<b>Char</b>	quantum of character sizes, in units of 1/240 of an inch. (i.e., a character is a multiple of Char units wide)

<b>Em</b>	size of an em in units of 1/240 of an inch.
<b>Halfline</b>	space moved by a half-linefeed (or half-reverse-linefeed) character in units in 1/240 of an inch.
<b>Adj</b>	quantum of white space, in 1/240 of an inch. (i.e., white spaces are a multiple of Adj units wide) Note: if this is less than the size of the space character, <i>nroff</i> will output fractional spaces using plot mode. Also, if the <i>-e</i> switch to <i>nroff</i> is used, Adj is set equal to Hor by <i>nroff</i> .
<b>twinit</b>	sequence of characters used to initialize the terminal in a mode suitable for <i>nroff</i> .
<b>twrest</b>	sequence of characters used to restore the terminal to normal mode.
<b>twnl</b>	sequence of characters used to move down one line.
<b>h1r</b>	sequence of characters used to move up one-half line.
<b>h1f</b>	sequence of characters used to move down one-half line.
<b>flr</b>	sequence of characters used to move up one line.
<b>bdon</b>	sequence of characters used to turn on hardware boldface mode, if any.
<b>bdoff</b>	sequence of characters used to turn off hardware boldface mode, if any.
<b>iton</b>	sequence of characters used to turn on hardware italics mode, if any.
<b>itoff</b>	sequence of characters used to turn off hardware italics mode, if any.
<b>ploton</b>	sequence of characters used to turn on hardware plot mode (for Diablo type mechanisms), if any.
<b>plotoff</b>	sequence of characters used to turn off hardware plot mode (for Diablo type mechanisms), if any.
<b>up</b>	sequence of characters used to move up one resolution unit (Vert) in plot mode, if any.
<b>down</b>	sequence of characters used to move down one resolution unit (Vert) in plot mode, if any.
<b>right</b>	sequence of characters used to move right one resolution unit (Hor) in plot mode, if any.
<b>left</b>	sequence of characters used to move left one resolution unit (Hor) in plot mode, if any.

This part of the driving table is fixed format, and you cannot change the order of entries. You should put entries on separate lines, and these lines should contain exactly two fields (no comments allowed) separated by white space. For example,

```

bset 0
breset 0
Hor 24

```

and so on.

Follow this first part of the driving table with a line containing the word “charset,” and then specify a table of special characters that you want to include. That is, specify all the non-ASCII characters that *nroff*(1) knows by two character names, such as \hy. If *nroff* does not find the word “charset” where it expects to, it will abort with an error message.

Each definition in the part after “charset” occupies one line, and has the following format:

chname width output

where “chname” is the (two letter) name of the special character, “width” is its width in ems, and “output” is the string of characters and escape sequences to send to the terminal to produce the special character.

If any field in the “charset” part of the driving table does not pertain to the output device, you may give that particular sequence as a null string, or leave out the entry. Special characters that do not have a definition in this file are ignored on output by *nroff*(1).

You may put the “charset” definitions in any order, so it is possible to speed up *nroff* by putting the most used characters first. For example,

```
charset
em 1 -
hy 1 -
\ - 1 -
bu 1 +\bo
```

and so on.

The best way to create a terminal table for a new device is to take an existing terminal table and edit it to suit your needs. Once you create such a file, put it in the directory `/usr/lib/nterm`, and give it the name `tab.xyz` where `xyz` is the name of the terminal and the name that you pass *nroff* via the `-T` option (for example, `nroff -Txyz`).

#### FILES

`/usr/lib/nterm/tab.name` terminal files

#### SEE ALSO

*nroff*(1)

**NAME**

*ptx* – make permuted index

**SYNOPSIS**

*ptx* [ options ] [ input [output] ]

**DESCRIPTION**

*ptx* generates the file *output* that can be processed with a text formatter (*nroff*(1) or *troff*) to produce a permuted index of file *input*. Standard input (–) and standard output are default. It has three phases: the first does the permutation, generating one line for each keyword in an input line. The keyword is rotated to the front. The permuted file is then sorted. Finally, the sorted lines are rotated, so the keyword comes at the middle of each line. *ptx* output is in the following form:

```
.xx "" "before keyword" "keyword" "after keyword"
```

where *.xx* is assumed to be an *nroff* or *troff* macro provided by the user or provided by *ptx*. The *–mptx* macro package provides the *.xx* macro definition. The *before keyword* and *after keyword* fields incorporate as much of the line as will fit around the *keyword* when it is printed. The first field and last field, at least one of which is always the empty string, are wrapped-around to fit in the unused space at the opposite end of the line.

The following *options* can be applied:

- f* Fold upper- and lower-case letters for sorting.
- t* Prepare the output for the phototypesetter.
- w n* Use the next argument, *n*, as the length of the output line. The default line length is 72 characters for *nroff* and 100 for *troff*.
- g n* Use the next argument, *n*, as the number of characters that *ptx* will reserve in its calculations for each gap among the four parts of the line as finally printed. The default gap is 3.
- o only* Use as keywords any words given in the *only* file.
- i ignore*  
Do not use as keywords any words given the *ignore* file. If the *–i* and *–o* options are missing, use */usr/lib/eign* as the *ignore* file.
- b break*  
Use the characters in the *break* file to separate words. Tab, new-line, and space characters are *always* used as break characters.
- r* Take any leading non-blank characters of each input line to be a reference identifier (as to a page or chapter), separate from the text of the line. Attach that identifier as a 5th field on each output line.

**FILES**

```
/bin/sort
/usr/lib/eign
/usr/lib/tmac/tmac.ptx
```

**SEE ALSO**

mm(1), nroff(2), mm(5), and mptx(5).

**BUGS**

Line length counts do not account for overstriking or proportional spacing. Lines that contain tildes (~) are botched because *ptx* uses that character internally. *ptx* does not discard non-alphanumeric characters.

**NAME**

subj – generate a list of subjects from a document

**SYNOPSIS**

subj files

**DESCRIPTION**

*subj* searches *files* for subjects that might be appropriate in a subject-page index and prints the list of subjects on the standard output. The document should contain formatting commands (from *nroff*(1), *troff*, and *mm*(1) among others) to make the best use of *subj*.

**SEE ALSO**

man(1), mm(1), ndx(1), and nroff(1).

**WARNINGS**

*subj* selects sequences of capitalized words as subjects except the first word in each sentence. Thus, if a sentence begins with a proper noun, the capitalization rule will not select this word as a subject. On the other hand, since each sentence is expected to begin on a newline, the first word of a sentence that begins in the middle of a line may be erroneously selected.

The output of *subj* may not be appropriate for your needs and should be edited accordingly.

**BUGS**

*subj* also selects as subjects modifier-noun sequences from the abstract, headings, and topic sentences (the first sentence in each paragraph), and occasionally a word is incorrectly categorized as a noun or adjective.

**NAME**

`tbl` – prepare tables for `nroff` or `troff`

**SYNOPSIS**

`tbl [ -T X ] [ -- ] [ files ] [ - ]`

**DESCRIPTION**

`tbl` is a preprocessor that prepares tables for `nroff(1)` or `troff`. `tbl` assumes that lines between the `.TS` and `.TE` command lines describe tables, thus they are re-formatted; lines outside these command lines are copied to the standard output. (`tbl` does not alter the `.TS` and `.TE` command lines.

Follow `.TS` with global options. The available global options are:

<b>center</b>	centers the table (default is left-adjust);
<b>expand</b>	makes the table as wide as the current line length;
<b>box</b>	encloses the table in a box;
<b>doublebox</b>	encloses the table in a double box;
<b>allbox</b>	encloses each item of the table in a box;
<b>tab (x)</b>	uses the character <code>x</code> instead of a tab to separate items in a line of input data.
<b>linesize (n)</b>	sets line or rules (e.g., from <code>box</code> ) in <code>n</code> -point type;

End the global options, if any, with a semi-colon (;).

After global options come lines describing the format of each line of the table. Each such format line describes one line of the table itself, except that the last format line (which you must end with a period) describes all remaining lines of the table. A single key letter describes each column of each line of the table. You may follow this key letter with specifiers that determine the font and point size of the corresponding item, that indicate where vertical bars are to appear between columns, and that determine column width, inter-column spacing, etc. The available key-letters are:

<b>c</b>	centers item within the column;
<b>r</b>	right-adjusts item within the column;
<b>l</b>	left-adjusts item within the column;
<b>n</b>	numerically adjusts item in the column: units positions of numbers are aligned vertically;
<b>s</b>	spans previous item on the left into this column;
<b>a</b>	centers longest line in this column and then left-adjust all other lines in this column with respect to that centered line;
<b>^</b>	spans down previous entry in this column;
<b>_</b>	replaces this entry with a horizontal line;
<b>=</b>	replaces this entry with a double horizontal line.

The characters **B** and **I** stand for the bold and italic fonts, respectively; the character **|** indicates a vertical line between columns.

The format lines are followed by lines containing the actual data for the table, followed finally by `.TE`. Within such data lines, data items are normally separated by tab characters.

If a data line consists of only `_` or `=`, a single or double line, respectively, is drawn across the table at that point; if a *single item* in a data line consists of only `_` or `=`, then that item is replaced by a single or double line. Some printers do not have the vertical resolution to produce double lines.

Full details of all these and other features of *tbl* are given in the tutorial and technical discussion cited below.

The `-TX` option forces *tbl* to use only full vertical line motions, making the output more suitable for devices that cannot generate partial vertical line motions (for example, line printers).

If you give no *files* as arguments, or if you specify `-` as *files*, *tbl* reads the standard input, so it may be used as a filter. When you use it with *neqn(1)*, put *tbl* first to minimize the volume of data passed through pipes.

#### EXAMPLE

If we let `→` represent a tab (which should be typed as a genuine tab), then the input:

```
.TS
center box ;
cB s s
cI | cI s
  ^ | c c
  l | n n .
Household Population

Town→Households
→Number→Size
=
Bedminster→789→3.26
Bernards Twp.→3087→3.74
Bernardsville→2018→3.30
Bound Brook→3425→3.04
Bridgewater→7897→3.81
Far Hills→240→3.19
.TE
```

yields:

Household Population		
Town	Households	
	Number	Size
Bedminster	789	3.26
Bernards Twp.	3087	3.74
Bernardsville	2018	3.30
Bound Brook	3425	3.04
Bridgewater	7897	3.81
Far Hills	240	3.19

#### SEE ALSO

*man(1)*, *mm(1)*, *neqn(1)*, *nroff(1)*, *man(5)*, and *mm(5)*,

#### BUGS

See BUGS under *nroff(1)*.

**NAME**

*xt* – multiplexed tty driver for AT&T windowing terminals

**DESCRIPTION**

The *xt* driver provides virtual *tty(7)* circuits multiplexed onto real *tty(7)* lines. It interposes its own channel multiplexing protocol as a line discipline between the real device driver and the standard *tty(7)* line disciplines.

Virtual *tty(7)* circuits are named by character-special files of the form */dev/xt???*. Filenames end in three digits, where the first two represent the channel group and the last represents the virtual *tty(7)* number (0-7) of the channel group. Allocation of a new channel group is done dynamically by attempting to open a name ending in 0 with the `O_EXCL` flag set. After a successful open, the *tty(7)* file onto which the channels are to be multiplexed should be passed to *xt* via the `XTIOCLINK ioctl(2)` request. Afterwards, all the channels in the group will behave as normal *tty(7)* files, with data passed in packets via the real *tty(7)* line.

The *xt* driver implements the protocol described in *xproto(5)* and in *layers(5)*. Packets are formatted as described in *xproto(5)*, while the contents of packets conform to the description in *layers(5)*.

There are three groups of `ioctl(2)` requests recognized by *xt*. The first group contains all the normal *tty ioctl(2)* requests described in *termio(7)*, with the addition of the following:

<b>TIOEXCL</b>	Set exclusive use mode; no further opens are permitted until the file has been closed.
<b>TIOCNXCL</b>	Reset exclusive use mode; further opens are once again permitted.

The second group of `ioctl(2)` requests concerns control of the windowing terminal, and is described in the header file `<sys/jioctl.h>`. The requests are as follows:

<b>JTYPE, JMPX</b>	Both return the value <code>JMPX</code> . These are used to identify a terminal device as an <i>xt</i> channel.
<b>JBOOT, JTERM</b>	Both generate an appropriate command packet to the windowing terminal affecting the layer associated with the file descriptor argument to <code>ioctl(2)</code> . They may return the error code <code>EIO</code> if the system <i>clist</i> is empty.
<b>JTIMO, JTIMOM</b>	<b>JTIMO</b> specifies the timeouts in seconds, and <b>JTIMOM</b> in milliseconds. Invalid except on channel 0. They may return the error code <code>EIO</code> if the system <i>clist</i> is empty.
<b>JWINSIZE</b>	Requires the address of a <i>jwinsize</i> structure as an argument. The window sizes of the layer associated with the file descriptor argument to <code>ioctl(2)</code> are copied to the structure.
<b>JZOMBOOT</b>	Generate a command packet to the windowing terminal to enter download mode on the channel associated with the file descriptor argument to <code>ioctl(2)</code> , like <b>JBOOT</b> ; but when the download is finished, make the layer a zombie (ready for debugging). It

may return the error code EIO if the system *clist* is empty.

**JAGENT**

Send the supplied data as a command packet to invoke a windowing terminal agent routine, and return the terminal's response to the calling process. Invalid except on the file descriptor for channel 0. See *jagent(5)*. It may return the error code EIO if the system *clist* is empty.

The third group of *ioctl(2)* requests concerns the configuration of *xt*, and is described in the header file `<sys/xt.h>`. The requests are as follows:

**XTIOCTYPE**

Returns the value XTIOCTYPE.

**XTIOCLINK**

Requires an argument that is a structure, *xtioclm*, containing a file descriptor for the file to be multiplexed and the maximum number of channels allowed. Invalid except on channel 0. This request may return one of the following errors:

- EINVAL** *nchans* has an illegal value.
- ENOTTY** *fd* does not describe a real *tty(7)* device.
- ENXIO** *linesw* is not configured with *xt*.
- EBUSY** An XTIOCLINK request has already been issued for the channel group.
- ENOMEM** There is no system memory available for allocating to the *tty(7)* structures.
- EIO** The JTIMOM packet described above could not be delivered.

layers(1) in the *User's Reference Manual*.

**NAME**

`xtd` – extract and print xt driver link structure

**SYNOPSIS**

`xtd [-f] [-n ...]`

**DESCRIPTION**

The `xtd` command is a debugging tool for the `xt(7)` driver. It performs an `XTIOCDATA ioctl(2)` call on its standard input file to extract the *Link* data structure for the attached group of channels. This call will fail if data extraction has not been configured in the driver or the standard input is not attached to an `xt(7)` channel. The data are printed one item per line on the standard output. The output should probably be formatted via `pr -3`.

The optional flags affect output as follows:

- `-n`      *n* is a number in the range 0 to 7. Channel *n* is included in the list of channels to be printed. The default prints all channels, whereas the occurrence of one or more channel numbers implies a subset.
- `-f`      Causes a “formfeed” character to be put out at the end of the output, for the benefit of page-display programs.

**EXIT STATUS**

Returns **0** upon successful completion, **1** otherwise.

**SEE ALSO**

`xts(1M)`, `xtd(1M)`, `ioctl(2)`, `xtproto(5)` in the *Programmer's Reference Manual*.

`xt(7)` in the *System Administrator's Reference Manual*.

`pr(1)` in the *User's Reference Manual*.

**NAME**

`xts` – extract and print `xt` driver statistics

**SYNOPSIS**

`xts` [-f]

**DESCRIPTION**

The `xts` command is a debugging tool for the `xt(7)` driver. It performs an `XTIOCSTATS ioctl(2)` call on its standard input file to extract the accumulated statistics for the attached group of channels. This call will fail if statistics have not been configured in the driver or the standard input is not attached to an `xt(7)` channel. The statistics are printed one item per line on the standard output.

-f Causes a “formfeed” character to be put out at the end of the output, for the benefit of page-display programs.

**EXIT STATUS**

Returns **0** upon successful completion, **1** otherwise.

**SEE ALSO**

`xt(7)`, `xtt(1M)`, `ioctl(2)`, `xtproto(5)` in the *Programmer's Reference Manual*.

`xt(7)` in the *System Administrator's Reference Manual*.

**NAME**

`xtt` – extract and print `xt` driver packet traces

**SYNOPSIS**

`xtt` [-f] [-o]

**DESCRIPTION**

The `xtt` command is a debugging tool for the `xt(7)` driver. It performs an `XTIOCTRACE ioctl(2)` call on its standard input file to turn on tracing and extract the circular packet trace buffer for the attached group of channels. This call will fail if tracing has not been configured in the driver, or the standard input is not attached to an `xt(7)` channel. The packets are printed on the standard output.

The optional flags are:

- f Causes a “formfeed” character to be put out at the end of the output, for the benefit of page-display programs.
- o Turns off further driver tracing.

**EXIT STATUS**

Returns **0** upon successful completion, **1** otherwise.

**NOTE**

If driver tracing has not been turned on for the terminal session by invoking `layers(1)` with the `-t` option, `xtt` will not generate any output the first time it is executed.

**SEE ALSO**

`xtd(1M)`, `xts(1M)`, `ioctl(2)`, `layers(5)` in the *Programmer's Reference Manual*.

`xt(7)` in the *System Administrator's Reference Manual*.

`layers(1)` in the *User's Reference Manual*.

**NAME**

locking – exclusive access to regions of a file

**SYNOPSIS**

```
int locking (filedes, mode, size);
int filedes, mode;
long size;
```

**DESCRIPTION**

*Locking* places or removes a lock on a region of a file. The calling process has exclusive access to regions it has locked: another process that attempts to read or write any byte in the locked region sleeps until the lock is removed. *Filedes* specifies the file to be locked or unlocked; *filedes* is a file descriptor returned by an *open*, *create*, *pipe*, or *dup* system call. *Mode* specifies the action: 0 for lock removal; 1 for blocking lock; 2 for checking lock. The region affected begins at the current file offset associated with *filedes* and is *size* bytes long. If *size* is zero, the region affected ends at the end of the file.

A blocking lock differs from a checking lock only when a process tries to lock bytes already locked by another process. A process locked out of a blocking lock goes to sleep until the specified region is free. A process locked out of a checking lock immediately gets an error return from *locking*.

Locking imposes no structure on a UNIX file. A process can arbitrarily lock any unlocked byte and unlock any locked byte. However, creating a large number of noncontiguous locked regions can fill up the lock table and make further locks impossible.

A process is said to be deadlocked if it is sleeping until an unlocking which cannot occur until after the deadlocked process itself does some unlocking. *Locking*, *read*, and *write* deliberately fail if proceeding would cause the calling process to be deadlocked.

Special files and pipes can be locked, but no input/output is blocked.

**SEE ALSO**

*creat(2)*, *close(2)*, *dup(2)*, *open(2)*, *read(2)*, *write(2)*.

**RETURN VALUE**

A return value of *-1* indicates an error, with the error value in *errno*. An unsuccessful checking lock sets *errno* to *EACCES*. A system call aborted by an incipient deadlock or by a lock table overflow sets *errno* to *EDEADLOCK*.

**WARNING**

Do not apply any standard input/output library function to a locked file.

## Important Information for Users of the UNIX PC UNIX Utilities Package

---

This update package contains additional information for use with the UNIX Utilities package. Please review this information and keep it with your installation documentation.

- 1 If you have previously installed Encryption (part of the System Software) on your UNIX PC and you are installing the Development Set (Disks 1-8) and/or the Enhanced Editors Disk (Disk 11), you need to remove Encryption from your UNIX PC and reinstall it after you have installed the Development Set and/or the Enhanced Editors.
  - 2 For EXPERT USERS ONLY: Since ksh is distributed as part of the UNIX Utilities, users who have invoked ksh as their default shell via the /etc/passwd file may be unable to log in after upgrading to Version 3.5 To avoid this situation, remove the reference to ksh from the passwd file for the **install** login (allowing it to default to /bin/sh), install the UNIX Utilities for the **install** login and then replace the reference to ksh in the passwd file for the **install** login.
-