AT&T

UNIX® System V/386

Release 3.2

UTILITIES RELEASE NOTES

UNIX® System V/386 Release 3.2 UTILITIES RELEASE NOTES

PRENTICE HALL

AT&T

UNIX® System V/386 Release 3.2

Utilities Release Notes

Prentice Hall, Englewood Cliffs, New Jersey 07632

Library of Congress Catalog Card Number: 88-62534

Editorial/production supervision: Karen Skrable Fortgang
Manufacturing buyer: Mary Ann Gloriande

Published by Prentice-Hall, Inc.
A Division of Simon & Schuster
Englewood Cliffs, New Jersey 07632

NOTICE
The information in this document is subject to change without notice. AT&T assumes no responsibility for any errors that may appear in this document.

The publisher offers discounts on this book when ordered in bulk quantities. For more information, write or call:

Special Sales
Prentice-Hall, Inc.
College Technical and Reference Division
Englewood Cliffs, NJ 07632
(201) 592-2498

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-944943-4

Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall Hispanoamericana, S.A., *Mexico*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Simon & Schuster Asia Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

CONTENTS

AT&T UNIX System V/386 Release 3.2 Release Notes

AT&T UNIX System V/386 Release 3.2 Software Development Release Notes

AT&T UNIX SYSTEM V/386
RELEASE 3.2
RELEASE NOTES

Preface

AT&T UNIX System V/386 Release 3.2 successfully merges the functionality of the AT&T UNIX System V/386 and *Microsoft* XENIX System V/386 operating systems into a single UNIX operating system for the *Intel* 80386 based computer. It provides an environment capable of running current application executables developed for earlier releases of AT&T UNIX System V/386 as well as *Microsoft* XENIX System V/386. The product provides support for application executables developed for AT&T UNIX System V/286 Release 2 and *Microsoft* XENIX System V/286.

These *Release Notes* describe how AT&T UNIX System V/386 Release 3.2 compares to both AT&T UNIX System V/386 and *Microsoft* XENIX System V/386, focusing on new or modified features and functionality. In addition, these *Release Notes* contain a list of known software problems and workarounds.

Consult the *Product Overview* for a complete description of the functionality and components (software and documentation) of AT&T UNIX System V/386 Release 3.2.

Notational Conventions

The following notational conventions are used throughout these *Release Notes*:

bold — User input, such as commands, options to commands, and names of directories and files, appear in **bold**.

italic — Names of variables to which values must be assigned (such as *filename*) appear in *italic*.

`constant width` — UNIX System output, such as prompt signs and responses to commands, appear in `constant width`.

<> — Input that does not appear on the screen when typed, such as passwords, keys used as commands, or <RETURN> and other special keys, appear between angle brackets.

<^*char*> — Control characters are shown between angle brackets because they do not appear on the screen when typed. The circumflex (^) represents the control key (usually labeled CTRL). To type a control character, hold down the control key while you type the character specified by *char*. For example, the notation <^**D**> means to hold down the control key while pressing the d key; the letter d will not appear on the screen.

[ ] — Command options and arguments that are optional, such as **[-msCj]**, are enclosed in square brackets.

| — The vertical bar separates optional arguments from which you may choose one. For example, when a command line has the format

command [*arg1* | *arg2*]

you may use either *arg1* or *arg2* when you issue *command*.

...	An ellipsis after an argument means that more than one argument may be used on a single command line.

command(number)	A command name followed by a number in parentheses refers to the part of a UNIX System reference manual that documents that command. (There are two reference manuals: the *User's/System Administrator's Reference Manual* and the *Programmer's Reference Manual.*) For example, the notation **cat**(1) refers to the page in Section 1 of the *User's/System Administrator's Reference Manual* that documents the **cat** command.

In sample commands, the dollar sign ($) is used as the shell command prompt. This is not true for all systems. Whichever symbol your system uses, keep in mind that prompts are produced by the system. Although a prompt is sometimes shown at the beginning of a command line as it would appear on your screen, you are not meant to type it. In addition, some examples may use the default superuser prompt, the pound sign (#). As with the system prompt, you are not meant to type the superuser prompt.

Foundation Set Software Packages

The Foundation Set is the fundamental UNIX System software product supplied with your system. The Foundation Set provides you with the UNIX operating system kernel and a basic set of utilities. The Foundation Set consists of the following separately installable packages:

- Base System Package
- Editing Package
- Remote Terminal Package
- Security Administration Package
- 2 Kilobyte File System Utility Package
- Network Support Utility Package
- Remote File Sharing Package
- XENIX File System Utility Package

The Base System Package is the minimal required UNIX System. The other Foundation Set packages are optional, and you do not need to install them if you do not require the utilities they provide.

Features of AT&T UNIX System V/386 Release 3.2

AT&T UNIX System V/386 Release 3.2 provides the following new features. Consult the *Product Overview* for an overview of all the features.

XENIX System V Compatibility

AT&T UNIX System V/386 Release 3.2 provides full binary and source code compatibility with applications developed for *Microsoft* XENIX System V/386, *Microsoft* XENIX System V/286, AT&T UNIX System V/386, and AT&T UNIX System V/286. The following list describes the level of XENIX System V support:

- Source code written for *Microsoft* XENIX System V/386 programs and applications can be compiled and linked on AT&T UNIX System V/386 without having to modify the source code.
- Binary applications developed for *Microsoft* XENIX System V/386 (Release 2.2.0 and later) and *Microsoft* XENIX System V/286 (Release 2.0 and later) can be run on AT&T UNIX System V/386 without having to recompile the applications.
- The structure of the AT&T UNIX System V/386 file system allows both XENIX System V and UNIX System V binary applications to be executed. It also supports the mounting of a XENIX System removable file system (with the XENIX file system add-on installed).
- Support for XENIX System call extensions enables programs to run as they did under the XENIX System.
- All device driver support routines available under *Microsoft* XENIX System V/386 are available in AT&T UNIX System V/386.

In addition to the program interface, several XENIX System V commands have been added for ease of use and compatibility. Tools are provided that allow the installation of all existing UNIX System and XENIX System packages.

The following sections describe the XENIX System calls that are not supported in Release 3.2 and those that are supported, but have slightly different functionality.

XENIX System Calls that Function Differently in Release 3.2

The following XENIX System calls are supported in AT&T UNIX System V/386 Release 3.2, but function differently in the XENIX System:

execseg()
The XENIX **execseg()** system call has been added to Release 3.2. This system call provides a means by which data can be executed. The **execseg()** system call returns a far pointer (selector and offset) to the start of the data segment. To execute the data, you must set the offset portion of the far pointer returned by **execseg** to the address of the data to be executed. Then, you must perform a far (intersegment) call through the far pointer. Because the AT&T compiler does not support the "near" or "far" keywords (which correspond to intra- and intersegment addressing, respectively), it is not possible to use the **execseg()** system call directly in C language. To use **execseg()**, you must use in-line assembly instructions, as shown in the following example.

```
typedef struct fcall    {
                int     (*fc_offset)();
                short   fc_selector;
} fcall_t ;
fcall_t         codeitem;             /* far pointer */
fcall_t         *fcptr = &codeitem;   /* pointer to far pointer */
extern char     code_in_data[];       /* data to be executed */
extern void     execseg();            /* void due to lack of far */
                                      /* keyword */

execseg();                            /* execseg returns far pointer */
asm("pushl      %edi");               /* in eax,edx.  This assembly  */
asm("movl       fcptr,%edi);          /* code places this return val */
asm("movl       %eax,(%edi)");        /* in *fcptr */
asm("movl       %edx,4(%edi)");
asm("popl       %edi");

                                        /* set the offset of the far pointer
                                          of the data to exec */

fcptr->fc_offset = (int (*)())code_in_data;

asm("pushl      %edi");                 /* do a far call into the data */
asm("movl       fcptr, %edi");          /* pointed to by fcptr */
asm("lcall      *(%edi)");
asm("popl       %edi");
```

Note that any data to be executed must return by means of an "lret" instruction.

In addition, only one call to **execseg()** is required for executing data. After the initial call, multiple **execseg()** calls have no effect.

To remove the ability to execute data, use the **unexecseg()** system call. This system call disables **execseg()** by invalidating the selector **execseg()** returns. As a result, any attempt to perform a far (intersegment) call through the pointer returned by an earlier **execseg()** call causes a segmentation violation. As with **execseg()**, multiple calls to **unexecseg()** have no effect after the initial call.

Neither **execseg()** nor **unexecseg()** accepts arguments or returns errors.

fcntl() and **lockf()**

In Release 3.2, file locks placed using the **fcntl()** or **lockf()** system calls are always enforced, if the source is compiled on *Microsoft* XENIX System V/386. However, these file locks are not always enforced when the source is compiled on Release 3.2. If the source is compiled on Release 3.2, you must use the **chmod**(1) command to guarantee enforcement of the file locks.

Microsoft XENIX System V/386 binaries that call **fcntl()** with cmd LK_GETLK and flock l_type F_UNLCK can receive blocking information. This feature is not supported in Release 3.2.

Microsoft XENIX System V/386 binaries that call **fcntl()** and/or **lockf()** always have their read and write access permissions checked. When compiled on Release 3.2, however, read and write permissions are checked only when trying to set a lock.

Microsoft XENIX System V/386 binaries that call **fcntl()** to set a non-blocking lock will receive EAGAIN if the call would block. On Release 3.2, binaries receive EACCES.

locking()

In Release 3.2, file locks placed using the XENIX **locking()** system call are always enforced.

ptrace()

In Release 3.2, the **ptrace()** system call is not supported for XENIX System binaries. XENIX System binaries that rely on **ptrace** to work will not work on Release 3.2. To make their code run on Release 3.2, you will have to modify their XENIX System code to use the UNIX System version of **ptrace**.

ulimit()

Microsoft XENIX System V/386 binaries that call **ulimit()** with the cmd argument set to 2, cannot increase their limit beyond the maximum number of blocks that are representable in a 512-byte block file system. This restriction is not enforced when the source is compiled on Release 3.2.

uname()

The utsname structure returned from **uname()** is a different size, depending on whether you compile on *Microsoft* XENIX System V/386 or on Release 3.2. On *Microsoft* XENIX System V/386, there are extra fields at the end of the structure.

XENIX System Calls Not Supported in Release 3.2

The following XENIX System calls are not supported on AT&T UNIX System V/386 Release 3.2:

brkctl()

nfs_sys()

proctl()

shutdown()
In Release 3.2, use the **uadmin()** system call with its **A_SHUTDOWN** command.

stkgrow()
Users will probably not be affected by the fact that this system call is not supported in Release 3.2.

swapadd()
In Release 3.2, use the **sysi86(SI86SWPI)** system call.

xlist() and **fxlist()**
The XENIX **xlist()** and **fxlist()** C-library functions are not supported by Release 3.2. Users must rewrite programs that use these two subroutines, using the 3.2 **nlist()** subroutine, as described in the *Programmer's Guide* and *Programmer's Reference Manual.*

Installing XENIX System Devices

XENIX System users should note that XENIX System device names (such as floppy drive devices) have been linked to their equivalent device names on AT&T UNIX System V/386 Release 3.2.

XENIX-286 Application Execution

This release of the UNIX System contains a *Microsoft* XENIX System V/286 utility that allows *Microsoft* XENIX System V/286 (*Microsoft* Release 2.3 and SCO™ Release 2.3.2) programs to run on the *Intel* 80386 processor under AT&T UNIX System V/386 Release 3.2.

For more information about the XENIX-286 feature, see the **x286emul**(1) manual page in the *Programmer's Reference Manual.*

Features from Microsoft XENIX System V/386

This section describes XENIX System utilities that have been added and Release 3.1 utilities that have been modified to support *Microsoft* XENIX System in Release 3.2.

New Utilities from XENIX System

The following utilities from *Microsoft* XENIX System V/386 are included in Release 3.2:

- clear(1) -- clears the terminal screen
- copy(1) -- copies multiple files, including directories
- csh(1) -- invokes a shell command interpreter with a C-like syntax
- ctags(1) -- creates a tags file for the **vi**(1) editor
- custom(1M) -- installs specific portions of XENIX System packages
- fixperm(1M) -- corrects or initializes XENIX System file permissions and ownership
- gethz(3C) -- returns the frequency of the system clock in ticks per second
- hd(1) -- displays files in hexadecimal format
- more(1) -- views a file one full screen at a time
- random(1) -- generates a random number
- settime(1) -- changes the access and modification dates of files
- strings(1) -- finds the printable strings in an object file
- sulogin(1M) -- allows access to single-user mode
- tset(1) -- provides information for setting terminal modes
- x286emul(1) -- emulates XENIX 80286

- xinstall(1M) -- XENIX System installation shell script
- xrestore(1M) -- invokes XENIX incremental file system restorer
- yes(1) -- repeats "yes" string to prompts

Release 3.2 Utilities with New XENIX System Support

The following utilities from Release 3.1 have been modified to include XENIX System support in Release 3.2:

- ascii(5) -- reflects addition of decimal table
- asy(7) -- supports XENIX System "exclusive open" capability
- cc(1) -- supports new **-Zp** option for packing structure members into memory
- console(1) -- supports XENIX System compatibility
- convert(1) -- supports conversion of XENIX System archives
- core(4) -- **usize** parameter has been changed to **USIZE**
- cpp(1) -- supports new **#pragma pack [1¦2¦4]** option
- crash(1M) -- supports XENIX System IFNAM files
- cron(1M) -- supports XENIX System **/etc/default/cron**
- df(1M) -- supports **-v** option from XENIX System
- display(7) -- supports XENIX System functionality, including ANSI escape sequences
- echo(1) -- supports **-n** option from XENIX System
- egrep(1) -- supports **-h** and **-y** options from XENIX System
- fd(7) -- supports XENIX System device names
- fgrep(1) -- supports **-h** and **-y** options from XENIX System
- file(1) -- supports XENIX System IFNAM binaries, archives, and other files
- fsck(1M) -- recognizes files of type IFNAM and allows their recovery

- grep(1) -- supports **-h** and **-y** options from XENIX System
- hd(7) -- supports XENIX System device names
- init(1) -- supports **sulogin** utility
- ipcs(1) -- supports new **-X** option for XENIX System compatibility
- keyboard(7) -- supports XENIX System ioctls
- limits(4) -- supports XENIX System-specific constants
- login(1) -- reflects changes in password aging and how the tty type is set in the environment
- ls (1) -- supports **lc** command from XENIX System, and lists XENIX System shared data and semaphores
- mdevice(4) -- supports ability to specify halt and poll routines in the functional field; also supports ability to share interrupts and DMA channels
- mount(1M) -- supports XENIX file system
- mountall(1M) -- supports XENIX file system
- passwd(1) -- supports three options (MINWEEK, MAXWEEK, and PASSLENGTH) read from */etc/default/passwd*
- pwck(1M) -- supports XENIX System password information
- sdb(1) -- recognizes IFNAM type files
- sdevice(4) -- supports ability to share interrupt vectors
- sh(1) -- supports the **-n** option to **echo**
- stty(1) -- supports XENIX System console mode
- su(1M) -- supports the */etc/default/su* file
- sysi86(2) -- supports 286 **x.out** emulation
- tar(1) -- supports XENIX System archives
- termio(7) -- supports XENIX System IOCTLs
- touch(1) -- merged with the XENIX System **settime** command

AT&T UNIX System V/386 Release 3.2 Base System Devices

The following list defines the contents of the master device file (*/etc/conf/cf.d/mdevice*) before any add-on packages are installed.

	Character		Device or Software Module
	(block)	Nodes In	Controlled By
Device	Device	/dev	"device driver"
Name	Number	Directory	(See Note 4)
asy	3	/dev/tty*	Serial Port (com1, com2)
fd	1 (1)	/dev/*dsk/f*	Floppy Disk
hd	0 (0)	/dev/*dsk/*s*	Hard Disk
kd	5	/dev/console	Keyboard
lp	7	/dev/lp*	Lineprinter (parallel interface)
mem	2	/dev/*mem	Kernel Memory Driver
rtc	8	(See Note 1)	Real time Clock
du	0	(See Note 2)	Distributed UNIX System (RFS) stubs
fp	0	(See Note 1)	Floating Point Support
cram	18	(See Note 1)	CMOS RAM (memory)
gentty	16	(See Note 2)	Generic tty (STREAMS support)
s52k	0	(See Note 2)	2K File System Support stubs
ipc	0	(See Note 1)	Interprocess Communications
msg	0	(See Note 1)	IPC Messages
sem	0	(See Note 1)	IPC Semaphores
shm	0	(See Note 1)	IPC Shared Memory
sxt	14	/dev/sxt*	Shell Layers
xt	13	/dev/xt*	Layers (Bit Mapped Terminal)
prf	6	/dev/prf	Kernel Profiler
cpyrt	0	(See Note 1)	System Initialization Messages
weitek	0	(See Note 1)	Numeric Chip Support stubs
vx	0	(See Note 1)	SimulTask 386 stubs
osm	17	(See Note 3)	Kernel printout Monitor
nmi	0	(See Note 1)	Nonmaskable Interrupt (NMI) Support
xsd	0	(See Note 1)	XENIX System Shared Data
xsem	0	(See Note 1)	XENIX System Semaphores

Notes:

1. Nodes are not required for this device driver. See Note 4.
2. These devices are required to support add-on packages, or are stubs (place holders) for add-on device driver packages. The add-on packages may install nodes in the */dev* directory.
3. The base system does not have nodes for this device. Nodes can be added later via the */etc/mknod* command.
4. Several *device drivers* are software only drivers. That is, they provide a kernel software function packaged as a *device* that can be added to or removed from the system.

Differences Between XENIX System and Release 3.2

This section points out some important differences between XENIX System V and AT&T UNIX System V/386 Release 3.2 that XENIX System users should know and keep in mind.

Using the ftime() System Call

Users should stop using the XENIX **ftime()** system call, and begin using the **time()** call provided in Release 3.2.

Using Shared Data and Semaphore Facilities

For 286 processes, there are significant differences between the XENIX operating system and the UNIX operating system in the positioning of shared data and its effect on future memory allocation. For this reason, it is recommended that XENIX System users discontinue using XENIX System-specific shared data and semaphore facilities and begin using the equivalent UNIX System facilities.

Understanding Terminal Types

XENIX System users should note that Release 3.2 supports terminal types that were not supported in the XENIX System. For example, terminal type AT386–M is the default terminal type for AT&T UNIX System V/386. Use this terminal type instead of ansi for all console virtual terminals. The terminal type at386 should be used with color consoles.

UNIX System users should be aware that the $TERM environment variable can be set automatically, along with the capabilities and attributes associated with the terminal type, by using the **tset**(1) command in the user's *.profile* and by administration of the */etc/ttytype* file. For additional information on using the **tset**(1) command and the */etc/ttytype* file, see **tset**(1) in the *User's/System Administrator's Reference Manual.*

Shutting Down the System

Release 3.2 does not support the XENIX System **haltsys** utility. In Release 3.2, administrators must use the **/etc/shutdown** utility to shut the system down.

Using the curses Utility

The XENIX System **curses** is "termcap" **curses**, whereas the Release 3.2 **curses** is "terminfo" **curses**. The **terminfo** and **termcap** facilities co-exist in Release 3.2. For this reason, the XENIX System **(termcap) curses** libraries and header files have been renamed in AT&T UNIX System V/386 Release 3.2. For example, **libxcurses** is the XENIX System **termcap** based **curses** library. The corresponding header file is **xcurses.h**.

UNIX System to XENIX System Floppy Diskette Sharing

By default, the UNIX System can read data from raw devices in multiples of variable size, whereas the XENIX System reads data from raw devices in multiples of 512-byte blocks. Therefore, when writing UNIX System media that will be read from a XENIX System raw device (such as **rfd0**, the XENIX System primary floppy disk drive), you must specifically set the UNIX System write blocking factor to be a multiple of 512 bytes, so the XENIX System raw device will recognize the blocksize.

One way to avoid this problem is to always read from the non-raw forms of XENIX System devices, instead of using the raw devices. For example, to read a UNIX System tar diskette from a XENIX System primary floppy disk drive, specify **/dev/fd0** on the tar command line, rather than **/dev/rfd0**.

Installation Notes

Overview

These installation notes provide information concerning the installation of an AT&T UNIX System V/386 Release 3.2 base system on your computer. Instructions are provided for the following conditions:

- New Installation
    - How to install AT&T UNIX System V/386 Release 3.2 over an AT&T 386 UNIX System V Release 3.1 or 3.1 Update system.
    - How to install AT&T UNIX System V/386 Release 3.2 over an already installed AT&T UNIX System V/386 Release 3.2.
    - How to install AT&T UNIX System V/386 Release 3.2 over a Non-AT&T UNIX System. (For example, a system with XENIX System or MS-DOS installed.)
    - How to install AT&T UNIX System V/386 Release 3.2 on a computer that has never had an operating system installed.
- Non-destructive installation
    - How to upgrade your AT&T 386 UNIX System V Release 3.1 or 3.1 Update base system software to AT&T UNIX System V/386 Release 3.2 without destroying any user files or non-Foundation Set packages. This is called a non-destructive upgrade installation.
    - How to overlay an AT&T UNIX System V/386 Release 3.2 over a previously installed AT&T UNIX System V/386 Release 3.2 without destroying any user files or non-Foundation Set packages. This is called a non-destructive overlay installation.

New Installation Notes

There are two ways you may be installing a new AT&T UNIX System V/386 Release 3.2.

- The AT&T UNIX System V/386 Release 3.2 base system is installed as "new" over any previous installed UNIX system. This includes AT&T 386 UNIX System V Release 3.1, Release 3.1 Update, or AT&T UNIX System V/386 Release 3.2. In this case you should begin with the section entitled "Initial Procedure".
- The AT&T UNIX System V/386 Release 3.2 base system is installed as the first system installed on your computer or Release 3.2 is installed over a non-AT&T UNIX System. In this case you may skip "Initial Procedure" and begin with the section entitled "Procedure for New Installation".

A new installation will destroy all files on the existing system. The Procedure For New Installation contains references to the installation procedures found in Chapter 2 of the *Operations/System Administration Guide.*

Installation of any new add-on packages, is covered in the "Install Optional Add-on Packages" section in the *Operations/System Administration Guide.*

Special Instructions for Upgrade and Overlay Installations

NOTE

If a new installation of AT&T UNIX System V/386 Release 3.2 is to be performed, this section may be skipped.

This section contains precautions and notes relating to a non-destructive upgrade or overlay installation. It is divided into three categories:

- Non-Destructive installation functionality

- Pre-installation procedures
- Post-installation procedures.

Non-Destructive Installation Functionality

The non-destructive installation performs the following high-level functions:

- Non-destructive upgrade installation
  - Upgrades from AT&T 386 UNIX System V Release 3.1 (or 3.1 Update) to AT&T UNIX System V/386 Release 3.2.
  - Does not destroy user data or non-Foundation Set add-on packages.
  - Preserves the current disk partitions and file systems.
  - Preserves the current user groups, logins, and passwords.
  - Removes the source files and data files associated with the *adm* command.
  - Requires removal of ,Foundation Set add-on packages.
  - Requires removal of existing line printer system, existing printers and classes, and any jobs in the printer queue.
  - Resets tunable parameters to default values.
- Non-destructive overlay installation
  - Overlays an AT&T UNIX System V/386 Release 3.2 over a previously installed AT&T UNIX System V/386 Release 3.2.
  - Does not destroy user data or non-Foundation Set add-on packages.
  - Preserves the current disk partitions and file systems.
  - Preserves the current user groups, logins, and passwords.
  - Does not require removal of Foundation Set add-on packages.
  - Does not require removal of existing line printer system, existing printers and classes, and any jobs in the printer queue.

- Resets tunable parameters to default values.

Pre-Installation Procedures

Before beginning a non-destructive installation, you should perform the following procedures:

- You will be asked in the procedure if you wish to do a system backup. It is recommended that a system backup be done prior to starting the installation in case anything goes wrong. Backups of the required files can be made from the command line. Chapter 4 of the *Operations/System Administration Guide* provides the details.
- Record the current system configuration. This includes the add-on software device drivers installed on the system, serial port configurations, line printer configurations, etc. This information will be used in the post-installation procedures.
- If you are performing a system upgrade (not an overlay), you will be required to remove all Foundation Set add-on packages. It is recommended to remove these packages prior to starting the installation. This will make the installation flow more smoothly. The "Remove Add-on Software Package" section in the *Operations/System Administration Guide* may be used to remove these packages.
- To ensure that the system will be properly configured in the post-installation procedures, it is recommended that all currently installed add-on driver packages be removed before performing a non-destructive installation. This is recommended since the installation procedure does not rebuild the UNIX System kernel to incorporate currently installed software device drivers. Also, the files containing the tunable parameters and other system configuration information are overwritten with default information. The "Remove Add-on Software Package" section in the *Operations/System Administration Guide* may be used to remove these packages.

Post-Installation Procedures

After the installation is completed, you will want to return the system to it's previous configuration. The following procedures should be followed using the system configuration that was recorded in the pre-installation procedures.

- The software device drivers that were removed prior to installation must be installed. The "Install Optional Add-on Packages" section in the *Operations/System Administration Guide* may be used to install these packages.
- The AT&T UNIX System V/386 Release 3.2 equivalents of the Foundation Set add-on packages that were removed prior to an upgrade installation should be installed. The "Install Optional Add-on Packages" section in the *Operations/System Administration Guide* may be used to install these packages.
- Configure the line printer subsystem, second serial ports, etc. Configuration changes can be made to the required files from the command line. Chapter 4 of the *Operations/System Administration Guide* provides the details.

Installation Procedure

Initial Procedure

1. If you have either AT&T 386 UNIX System V Release 3.1 (or 3.1 Update) or AT&T UNIX System V/386 Release 3.2, perform the procedures starting at the beginning of Chapter 2 of the *Operations/System Administration Guide* through Step 5 in the section "Boot System to Single User Mode".
2. You will receive the following message:

```
Is this a new installation or a release upgrade to your
existing system?  (Strike "n" (new) or "u" (upgrade)
followed by ENTER).
```

3. If you enter **n**, use the "Procedure for New Installation" in this document.
4. If you enter **u**, use the "Procedure For Installing Upgrade" in this document.

Procedure for New Installation

NOTE

This procedure assumes you have made the **n** response to Step 2 in the Initial Procedure.

or

The AT&T UNIX System V/386 Release 3.2 base system is installed as the first system installed on your computer or Release 3.2 is installed over a non-AT&T UNIX System. In this case, perform the procedures starting at the beginning of Chapter 2 of the *Operations/System Administration Guide* through Step 5 in the section "Boot System to Single User Mode" and return to Step 1 of this procedure.

or

You struck ENTER in Step 2 of the Procedure for Installing Upgrade.

1. You will receive the following message:

```
WARNING:  A new installation of the UNIX System will destroy
all files currently on the system.  Do you wish to continue (y or n)?
```

To continue type **y** and proceed to Chapter 2 of the *Operations/System Administration Guide*. Perform the "Partition the Hard Disk" procedure.

If you type **n**, you will be placed at the single-user shell prompt.

If you wish to start this procedure over and not reboot the system, type:

```
INSTALL
```

If you wish to start the procedure again and reboot the system:

a. Type:

```
uadmin 2 0
```

b. Boot the system off the floppy by striking CTRL, ALT, and DEL simultaneously.

c. Start the installation process again by returning to the "Initial Procedure" section in this document.

2. If you type y, you will then receive the following message:

```
A surface analysis will now be done.
This will destroy all data on the hard disk.
Strike ENTER to continue or DEL to abort.
```

If you wish to continue, strike ENTER.

If you type DEL, you will be placed at the single-user shell prompt.

If you wish to start this procedure over and not reboot the system, type:

```
INSTALL
```

If you wish to start the procedure again and reboot the system:

a. Type:

```
uadmin 2 0
```

b. Boot the system off the floppy by striking CTRL, ALT, and DEL simultaneously.

c. Start the installation process again by returning to the "Initial Procedure" section in this document.

3. If you strike ENTER, you will see the following message:

```
UNIX System file system(s) will now be created on the hard disk ...
```

4. Go to Step 10 in the section "Create UNIX System File Systems" in Chapter 2 of the *Operations/System Administration Guide* and complete the procedure for installing a base system on your computer.

Procedure for Installing Upgrade

NOTE

This procedure assumes you have made the **u** response to Step 2 in the initial procedure.

1. A sanity check of the file systems will be done to determine if there is any uncorrectable file system damage that cannot be upgraded. If there is, an upgrade installation cannot be performed.

   NOTE

   The following screen may be ignored if a system backup was performed prior to starting this installation. You may continue by striking ENTER.

   If the sanity check determines that there is no file systems damage, you will receive the following message:

```
WARNING:  A system backup is suggested prior to doing a
system upgrade.  To ensure a correctly functioning system,
various add-on packages supplied with the foundation set
will need to be removed.  Also, if any packages are
installed that reconfigure the UNIX System kernel, they
should be removed before proceeding with the new installation.
To proceed with the installation strike ENTER, otherwise,
strike DEL to abort.
```

   To continue with an upgrade installation, strike ENTER.

   To abort the installation procedure, strike DEL. You will then be given the single-user shell prompt. Do the following:

   a. Type:

```
uadmin 2 0
```

b. Remove the floppy from the disk drive.

c. Boot the system off the hard disk by striking CTRL, ALT, and DEL simultaneously.

d. Back up your files [use **face** menus or use the command level (refer to Chapter 4 in the *Operations/System Administration Guide*)].

e. Shutdown the system.

f. Reinsert the floppy and boot off the floppy by striking CTRL, ALT, and DEL simultaneously.

g. Start the installation process again by returning to the "Initial Procedure" section in this document.

2. If the sanity check determines that there is file system damage, you will receive the following message:

```
You will be unable to do an upgrade because your system
does not contain a valid UNIX System.  Please consult your
"Release Notes" for further information.
Strike ENTER to continue with a new installation
or DEL to abort the installation procedure.
```

To continue strike ENTER. You will perform a new installation procedure. Go to the Procedure for New Installation.

To abort the installation procedure, strike DEL. You will then be given the single-user shell prompt. You may not be able to reboot the system. If you wish to start this procedure over without a reboot, type:

```
INSTALL
```

3. If you strike ENTER in Step 1, the root (/) file system will be checked to make sure that there is enough space for the files to be saved. Ten free blocks are needed. If there is not enough space, you will receive the following message:

```
There is not enough space in the root filesystem on your hard disk
to back up files for a system upgrade.  Please remove some files and
try again.  Consult your "Release Notes" for further information.
```

If there is not enough space, you will be given the single-user shell prompt. Do the following:

a. Type:

```
        uadmin 2 0
```

b. Remove the floppy from the disk drive.

c. Boot the system off the hard disk by striking CTRL, ALT, and DEL simultaneously.

d. When the system comes up, remove enough root files to free 10 blocks. If you have created any files under /, consider these for removal.

e. Shut down the system.

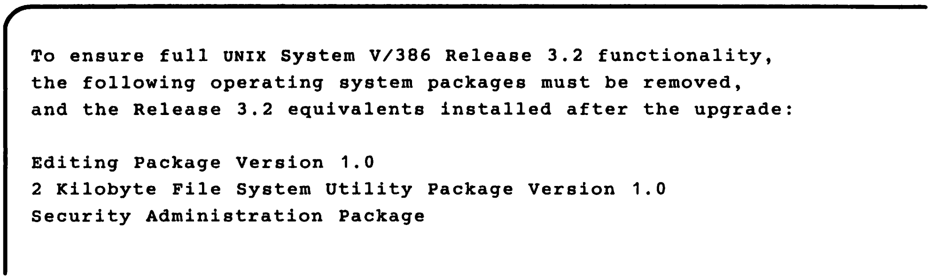
f. Reinsert the floppy and boot off the floppy by striking CTRL, ALT, and DEL simultaneously.

g. Start the installation process again by returning to the "Initial Procedure" section in this document.

4. If you are upgrading from Release 3.1 or 3.1 Update and if any Foundation Set add-on packages are installed, you will receive a message similar to the following:

NOTE	If your Foundation Set add-on packages were removed prior to starting this procedure, the following screen will not be seen.

NOTE You may have different packages installed from those shown in the following screen.



```
To ensure full UNIX System V/386 Release 3.2 functionality,
the following operating system packages must be removed,
and the Release 3.2 equivalents installed after the upgrade:

Editing Package Version 1.0
2 Kilobyte File System Utility Package Version 1.0
Security Administration Package
```

If not removed prior to the start of installation, you must remove all installed Foundation Set add-on packages before proceeding to ensure full Release 3.2 functionality. If you do not remove these packages all at once, the next time you select **u** to upgrade your system, you will receive the same message with the current packages to be removed.

If you are running Release 3.2, and doing the overlay upgrade, you will not be required to remove the Foundation Set add-on packages.

If any of these packages are installed, you will be put in single-user shell prompt.

a. Type:

```
uadmin 2 0
```

b. Remove the floppy from the disk drive.

c. Boot the system off the hard disk by striking CTRL, ALT, and DEL simultaneously.

d. When the system comes up, execute "removepkg" as covered in "Remove Add-on Software Package" in Chapter 2 of the *Operations/System Administration Guide* to remove the listed packages.

e. Shutdown the system.

f. Reinsert the floppy and boot off the floppy by striking CTRL, ALT, and DEL simultaneously.

g. Start the installation process again by returning to the "Initial Procedure" section in this document.

NOTE

The packages that are discussed below should have been backed up as discussed in the pre-installation procedures.

5. If all the Foundation Set add-on packages are removed, but there are still some other packages installed (e.g., *Crystal-Writer* and STARLAN Network), you will receive the following message:

```
WARNING:  If any of the packages currently on the system fail to
work after the upgrade, remove the package and then re-install it.
```

6. If the Release 3.1 Line Printer (LP) System is on your machine, you will receive the following message:

```
The 3.1 LP system exists on this machine.
This installation will remove the following:

Existing LP System
Existing Printers and Classes
Any Jobs in the Printer queues

Strike ENTER to continue or DEL to abort
```

If you wish to preserve your LP files, strike DEL. You will then be put in single-user shell prompt.

a. Type:

```
        uadmin 2 0
```

b. Remove the floppy from the disk drive.

c. Boot the system off the hard disk by striking CTRL, ALT, and DEL simultaneously.

d. Backup or save the LP files required.

e. Reinsert the floppy and boot off the floppy by striking CTRL, ALT, and DEL simultaneously.

f. Start the installation process again by returning to the "Initial Procedure" section in this document.

7. A UNIX System will be installed on the hard disk. You will see the following message:

```
A UNIX System will now be installed on your hard disk ...
```

8. After a delay you will see the following message:

```
Please standby

When you are prompted to reboot your system,
remove the floppy disk from the diskette drive,
and strike CTRL-ALT-DEL.
Please wait for the prompt.
```

9. Your prompt to reboot will appear as follows:

```
Reboot the system now.
```

10. To install the remainder of the base system, perform the procedures in the section "Install the Remainder of the Base System" in Chapter 2 of the *Operations/System Administration Guide.*

11. Configure your system as instructed in the sections "Pre-Installation Procedures" and "Post-Installation Procedures".

Overview of Basic Procedures

The following sections provide brief descriptions of some basic UNIX System procedures.

Booting the System

The following list of instructions will boot an AT&T UNIX System V/386 Release 3.2 system after the Base System has been installed. Apply power to the system and wait 1 or 2 minutes; the second-stage boot should be loaded automatically by the PROMs.

The boot program will load and give you the following message:

```
Booting UNIX System . . .
```

If you immediately press ANY key, the boot process will be interrupted and the following message will be displayed:

```
Enter name of a kernel to boot:
```

If you enter the name of a kernel to boot, the boot process will continue. If you do not enter anything, after 1 minute **/unix** will boot automatically; if you want to boot **/unix** immediately, press the Enter key.

Shutting Down the System

The following steps are necessary to shut down an AT&T UNIX System V/386 Release 3.2 computer properly:

1. Log in as **root** and change your directory to /.. Note that you must be on the console (/dev/console).
2. Run the shutdown program with the following options:

   shutdown -y -gTIME

   where TIME is the number of seconds to be allotted before the system is actually halted. A time factor of at least 120 seconds (2 minutes) is recommended for your systems that are being used as multi-user sites. The time factor allows you to exit editors and save programs before the system goes down.

The system will proceed to shut itself down after the allotted time, and when the message

```
Reboot the system now
```

appears, the system can be turned off. The [Ctrl] [Alt] [Delete] key combination can be pressed, or the system can be turned off and then turned on again in order to reboot at this point.

Instructions for Updating Selected Files from the Release

If necessary, selected files may be extracted from the installation diskettes. Appendix A contains a list of all the files on the Foundation Set diskettes. All the installation diskettes, with the exception of diskette 1 of 7 of the Base System and diskette 1 of 1 of the Remote Terminal Package, are cpio diskettes. Diskette 1 of 7 of the Base System and diskette 1 of 1 of the Remote Terminal Package are mountable file systems. If the file that you want to extract is on a cpio diskette, you should insert the diskette into the floppy diskette drive and use the following command to extract the selected file:

```
cpio -icBvd filename < /dev/dsk/f0
```

If the file you want to extract is on a mountable file system diskette, you should insert the diskette into the floppy diskette drive and use the following commands to extract the selected file:

```
/etc/mount -r /dev/dsk/f0 /mnt
cp /mnt/<filename> <newfile>
/etc/umount /dev/dsk/f0
```

Miscellaneous UNIX System Reminders

The following sections provide reminders and general system troubleshooting information.

Converting to getopts by Hand

getoptcvt [see **getopts**(1)] adds about 30 lines of code to a shell script, so you may want to convert scripts by hand instead. Converting by hand probably will make the code cleaner and easier to understand. Also, you do not have to worry about parsing option-arguments that are also options.

Follow these guidelines to convert most scripts that currently use the **getopt**(1) command.

Step 1 Delete the old invocation line and the **if** statement that checks the exit code.

Step 2 Change the **for** loop to a **while** loop that invokes **getopt**(1).

Step 3 Change the patterns in the **case** statement from **-option** to single option letters.

Step 4 Delete the case for --.

Step 5 Add a case for **'?'**. This case may be used to print the usage message and to exit with a non-zero exit code. Note that the **?** is quoted since it is interpreted for filename expansion.

Step 6 Remove all **shift** commands within the **case** statement.

Step 7 Change **$2** to **$OPTARG** for cases that require an option argument.

Step 8 Add the statement **shift 'expr $OPTIND - 1'** after the **while** loop so the remaining arguments can be referenced as before. Following is an example of a script before and after conversion.

```
# before conversion
set -- 'getopt abo: $*'
if [ $? != 0 ]
then
        echo $USAGE
        exit 2
fi
for i in $*
do
        case $i in
        -a | -b)        FLAG=$i; shift;;
        -o)     OARG=$2; shift 2;;
        --)     shift; break;;
        esac
done
```

```
# after conversion
while getopts abo: i
do
        case $i in
        a | b)  FLAG=$i;;
        o)      OARG=$OPTARG;;
        ?)      echo $USAGE
                exit 2;;
        esac
done
shift 'expr $OPTIND - 1'
```

If you want your script to be compatible with earlier UNIX systems (that is, use either **getopts** or **getopt**), convert it as the following example shows:

```
if [ "$OPTIND" = 1 ]
then
        while getopts abo: i
        do
                case $i in
                a | b)  FLAG=$i;;
                o)      OARG=$OPTARG;;
                ?)      echo $USAGE
                        exit 2;;
                esac
        done
        shift 'expr $OPTIND - 1'
        echo $*
else
        set -- 'getopt abo: $*'
        if [ $? != 0 ]
        then
                echo $USAGE
                exit 2
        fi
        for i in $*
        do
                case $i in
                -a | -b)        FLAG=$i; shift;;
                -o)     OARG=$2; shift 2;;
                --)     shift; break;;
                esac
        done
        echo $*
fi
```

edit, ex, vedit, vi, view

The **edit, ex, vedit, vi,** and **view** commands allow separate **.exrc** files in any directory. In addition, if you change directory to another user's directory and use any of these editors to edit a file in that other user's directory, the editor will execute the **.exrc** file if it exists in the second user's directory. This functionality has security implications depending on the contents of the **.exrc** file, because the commands are executed as the user invoking the editor and not as the person who owns the **.exrc** file.

In this release a new option has been added to the **vi/ex** commands to allow you the option of reading the **.exrc** file in the current directory. Initially, the flag is NOT set. That is, the **vi/ex** command will NOT read the **.exrc** file if it exists in the current working directory. You can modify this option by inserting the line

```
set exrc
```

or the abbreviation

```
set ex
```

in the **$HOME/.exrc** file which is read when one of these editors is executed if the **EXINIT** variable is not set in the **.profile**. If you want to set the **EXINIT** variable, add the following lines to your **.profile**:

```
EXINIT="set exrc"
export EXINIT
```

However, you should note that executing **vi/ex** as another user with **su** could result in your files being compromised, since certain variables in the environment are passed when **su** is executed without the "-".

For more information, see the **ex**(1) manual page in the *User's/System Administrator's Reference Manual.*

Floating Point Emulation

Two floating point emulators are provided in the product. The default emulator, called */etc/emulator.dflt*, is linked to */etc/emulator* and provides better performance than the second emulator, kept in */etc/emulator.rel1*. However, */etc/emulator.dflt* does NOT emulate all the instructions of the 80387 processor, specifically the following:

FCOS	cosine function (80387 only)
FDECSTP	decrement stack pointer
FINCSTP	increment stack pointer
FPREM1	partial remainder (80387 only)
FRSTOR	restore saved state
FSAVE	save state
FSETPM	set protected mode
FSIN	only sine function (80387 only)
FSINCOS	sine & cosine function (80387 only)
FUCOM	unordered comparison (80387 only)
FUCOMP	unordered comparison and pop (80387 only)
FUCOMPP	unordered comparison and double pop (80387 only)

The second emulator, */etc/emulator.rel1*, provides complete emulation of the 80387 instruction set. However, this emulator has less performance than the default emulator. This second emulator should be used if problems occur with UNIX System 286 applications. The problems will manifest themselves by the application core dumping with a floating point exception.

If you need to change from the */etc/emulator.dflt* emulator to the */etc/emulator.rel1* emulator, log in as root and type the following command:

```
ln /etc/emulator.rel1 /etc/emulator
```

Similarly, if you need to change from the */etc/emulator.rel1* floating point emulator to */etc/emulator.dflt*, log in as root and type the following command:

```
ln /etc/emulator.dflt /etc/emulator
```

After you relink the proper emulator, the machine must be rebooted for the system to start using the other emulator.

Floppy Disk Operations

The following enhancements have been made to the floppy disk subsystem for AT&T UNIX System V/386 Release 3.2:

- Enhanced 3.5 inch floppy disk support now allows the operating system to access a 3.5 inch diskette drive as either diskette 0 or diskette 1. In the previous release, the 3.5 inch floppy drive could be accessed only as diskette 1.
- Automatic format detection of floppy diskettes allows access to floppies using generic device names regardless of the format of the diskette inserted into the drive. The following nodes have been added to the system to support this enhancement:

  ```
  /dev/(r)dsk/f0
  /dev/(r)dsk/f0t
  /dev/(r)dsk/f1
  /dev/(r)dsk/f1t
  ```

  For example, if drive 0 is a 5.25 inch floppy drive and you want to create a cpio file on a 1.2Mb diskette, you would insert a formated 1.2Mb floppy diskette into the drive and type the following command:

  ```
  find . -print | cpio -ocv > /dev/rdsk/f0t
  ```

  Similarly, you can insert a formated 360 Kb diskette into the drive and issue the same command. The system will automatically detect that a formatted 360 Kb diskette is in the drive and process the command appropriately.

The following list describes important reminders and troubleshooting information for floppy disk operations:

- Some error messages from floppy operations appear only at the console, regardless of which terminal invoked the floppy command from which the error originated. This is true for all errors detected and displayed by the device drivers.
- Sometimes reading from a 360 Kb disk drive may fail and it will appear as though the floppy disk drive door is not closed. If this should occur, reinsert the floppy disk and close the drive door.

- On some 80386 computers, the system will not recognize that the floppy disk drive door is open if the floppy disk is completely inserted into the disk drive and the door is left open. To correct this, make sure that the floppy drive door is always closed when a floppy disk is completely inserted.

Kernel Operations

The following list describes important reminders and troubleshooting information for performing kernel procedures:

- The process accounting computation of a process's memory usage is incorrectly maintained. The value calculated for process memory usage is slightly less than the real result.
- When a program executes integer division by zero, the following error message is displayed:

  ```
  floating exception - core dumped
  ```

  This message does not accurately describe the error.
- Processes spawned by the kernel at boot time (**sched, /etc/init, vhand, bdflush**) have start times (**STIME**), that is the time the system was last brought down, not the time they were spawned.
- Some core dumps may have possible file size errors reported by **fsck**, but these are only warnings and can be ignored. To determine whether the possible file size errors reported are resulting from core dumps, execute: **ncheck -i** *i-number* where *i-number* is given in the **fsck** message:

  ```
  POSSIBLE FILE SIZE ERROR I=i-number
  ```

  ncheck will generate the path name of a file from its inode number, *i-number*.
- If the operating system runs out of free **clists**, all input/output activity from/to terminal ports and the console will cease. No warning message is printed by the system to show that it is out of **clists**.
- The value of the **SHMALL** tunable parameter specifies the maximum number of in-use shared memory segments allowable systemwide. This parameter is not checked by the system [that is, **shmget**(2) does not check this limit].

login

To discourage intruders, the encrypted password and password aging information formerly found in */etc/passwd* has been moved to */etc/shadow.* This file can be read only by the superuser. You will still be able to change their passwords using the **passwd**(1) command. Password and aging information is added to */etc/shadow* by running a new program, **pwconv**(1M). This program can be executed only by the superuser.

If you have an application or program that writes password and/or aging information into */etc/passwd,* the program will have to be modified so that **pwconv**(1M) is executed after the information is appended to */etc/passwd.* Until the modification can be made, the administrator with superuser privilege will have to run the program before the user who has been added or whose password information has been modified can log in.

To set a variable in an */etc/default* file, the name of the variable is followed by an "=" and the value of the variable, with no embedded spaces or tabs. In */etc/default/login,* the following variables may be set:

CONSOLE — If set, only the superuser may login on the terminal defined as the console. For example,

```
CONSOLE=/dev/console
```

means that only root may login on the console. If CONSOLE is not in */etc/default/login,* the superuser may login on any terminal.

ALTSHELL — If set to YES, the SHELL environment variable will be set to the users shell, if that shell is not */bin/sh.* If set to NO, the names of nonstandard shells will not be put in the SHELL environment variable. The default value for this variable is NO.

PASSREQ — If set to YES, all users must have have a password. Any user without a password will be asked for one at the first opportunity permitted by the password aging set for that user (i.e., users without passwords may not change their NULL passwords if password aging is enabled for them, and the minimum time before a password can be changed has not elapsed).

TIMEZONE This variable sets the TZ variable in the environment of the user. It must match the timezone set in */etc/TIMEZONE.*

HZ This variable sets the environment HZ, the rate of the system clock, for the user logging in.

PATH This variable sets a default path for a user who does not have uid 0.

SUPATH This variable sets the default path for the superuser logging in. Another default path for the superuser is in */etc/default/su*, which is set for superusers who did not login as such.

ULIMIT This variable sets the maximum file size for a user. It is in units of 512-byte blocks.

TIMEOUT This variable is the length of time which "login" will wait for a password after receiving a user name. It is in units of seconds.

UMASK This variable is the default umask for users.

IDLEWEEKS This variable is the number of weeks which an account may remain idle before its login is disabled.

passwd

The */etc/default/passwd* file has the following variables which may be set:

PASSLENGTH This variable is the minimum length of a password. Any password shorter than this length will be disallowed. The default length, if the variable is not set in the defaults file, is 6.

MINWEEKS This variable is the number of weeks, after changing a passsword, during which the password may not be changed again.

MAXWEEKS This variable is the number of weeks, after changing a password, after which you will be requested to change your password when you next log on.

The MINWEEKS and MAXWEEKS variables may be overridden with the **passwd** program, by explicitly setting password aging for a particular user.

Changing the ULIMIT Parameter

In AT&T 386 UNIX System V Release 3.1, the default ulimit is a tunable parameter settable via ID/TP in the operating system. If an administrator wanted larger values for the users of the system, the ID/TP ulimit value would be changed by the administrator to reflect the new value. The system would then need to be rebooted before the new default value went into effect.

In AT&T UNIX System V/386 Release 3.2, it is possible to override the ulimit set in the operating system by setting the ULIMIT parameter in */etc/default/login*. The current */etc/default/login* sets the ULIMIT parameter to 4096. If an administrator wishes to change the default ulimit, both the ID/TP value and the */etc/default/login* value would need to be changed to reflect the new value.

Longest Allowed Path Names

The longest path name is restricted to 1024 bytes. System calls that require path names as arguments will now fail, setting **errno** to **ENOENT**, if a longer path name is given.

Previously, the path name was not restricted by the UNIX operating system; however, most programs gave an ad hoc limit to the length. Generally, these limits were well below 1024 bytes, so most programs should not be affected by this change.

The **limits.h** file defines a macro **PATH_MAX** to be the longest length of a path name. In Release 3.1 this file incorrectly sets the macro to 256, but it will probably be changed in a future release to 1024. Local system administrators can safely change the value for **PATH_MAX** to 1024 without harm, since the Release 3.1 system internally uses the longer limit.

You are encouraged to include the *limits.h* file with a statement like

```
#include <limits.h>
```

and to refer to the **PATH_MAX** macro for the longest path name allowed.

Saving Device Files When Backing Up root File System

When you back up the **root** (/) file system using the **backup**(1M) command, the device files (*/dev* directory) are not saved as part of the backup. To save the device files, become the superuser, mount a blank formatted floppy that has a file system on it, and enter the following commands:

```
# mount /dev/... /mnt
# find /dev -print | cpio -pdl /mnt
```

where */mnt* is the directory on which the floppy disk file system is mounted. The **cpio** options are lowercase letters **p**, **d**, and **l**.

To restore the files, insert the floppy on which the files were saved, and enter the following commands:

```
#  /etc/mount /dev/dsk/f0 /mnt
•
•
•
#  cd /mnt
#  find . -print ¦ cpio -pdl /dev
•
•
•
#  /etc/umount /dev/dsk/f0q15d
```

Shell Scripts

It is strongly recommended that all applications convert any shell scripts into binary programs if specific user (group) permissions are required in the shell script command lines. To pass permissions, the binary program must have the setuid (gid) mode bit on and the owner (group) of the binary program set to the ID required. Then the **exec**(2) system call can be invoked with the binary program as the argument and the correct permissions will be passed.

If it is not possible to convert the shell scripts into binary programs, then a binary interface program must be written that would have the setuid mode bit on and the owner of the file set to **root**. Next, the process would have to do a **setuid (gid)** system call internally with the uid (gid) that must be passed to a sub-shell. This is only possible because the **setuid (setgid)** system call sets both the real and effective uid (gid) when called by a process with the effective uid of **root**. Finally, the binary interface would then call the shell script. This is a potentially dangerous procedure unless the programmer is aware of all the implications.

Invoking Bourne Shell Scripts from CSH

A single line comprised of a colon (:) should be added as the first line of Bourne Shell scripts when these scripts are to be invoked from **csh**(1). This will cause **csh**(1) to recognize the script as a Bourne Shell Script and execute it appropriately.

Software Notes

This section describes problems that may occur with Release 3.2, and in some cases, workarounds to those problems.

mknod(1M)

The usage message for the **mknod**(1M) command is incorrect. It does not agree with the documentation since it does not display the **p** option for creating named pipes. The *User's/System Administrator's Reference Manual* manual page for **mknod**(1M) correctly documents the usage. **mknod**(1M).

layers(1)

XENIX termcap-based applications, such as SCO Professional, does not work properly under the UNIX System **xt-layers**. The problem is when TERM=630 or 5620 is declared, the size of the screen is defined at 70 lines. When **xt-layers** is invoked, and windows of a smaller size are created, SCO Professional still thinks that is dealing with a full screen. Since the screen is not 80 characters wide, double lines are created.

Workaround: The problem is that SCO Professional applications look in */etc/termcap* directly for the value of the lines and columns. Therefore, it does not paint the screen correctly when a window with fewer lines and columns is created.

In order to have a */etc/termcap* entry work on a terminal that has layers invoked, you will have to create a new entry in the */etc/termcap* file. To create the new termcap description, edit the */etc/termcap* file and search for the terminal name that you want to use with layers. The first line of a termcap entry begins in column one, the actual terminal description follows. The lines that describe that terminal start with a <TAB> and end with a backslash (\). Copy that termcap definition up to the beginning of the next terminal description. The following procedure will accomplish this goal.

1. Change the name that you would set your TERM to.

   EXAMPLE:
   for 5620/dmd terminal, change the following line from:

   ```
   att5620|ATT5620|dmd|tty5620|ttydmd|5620|5620
   terminal 88 columns @(#)5620.ti 1.1:
   ```

to:

```
att5620-s|ATT5620-s|dmd-s|tty5620-s|ttydmd-
s|5620-s|5620 terminal < 88 columns @(#)5620.ti
1.1:
```

This allows you to set your TERM to "dmd-l" which is used in this example to specify a dmd invoked with layers.

2. Modify the variables for lines "li" and columns "co". Search for the line that has the lines and columns defined. It will look similar to:

   ```
   :co#88:li#70:kn#4:
   ```

   In this example, change the "88" to the number of columns you predict will be in your average size window. Also, change the number of lines from "70" to what you would expect them to be.

3. Using "dmd-s" as an example, set your TERM to the name you gave your new termcap definition before invoking the SCO application. However, this TERM name is not recognized by curses applications (i.e., **vi**, **ed**, etc). You will be able to set your TERM to one type if you link */usr/lib/terminfo/d/dmd* to */usr/lib/terminfo/d/dmd-s*.

ps(1)

In AT&T 386 UNIX System V Release 3.1, the usage of **ps** changed. The **ps** command now checks and sets the user's effective UID to the real UID and the effective GID to the real GID. Therefore, only users with a real user id of **root** or a real group id of **sys** will be able to use the -l options to **ps**.

/etc/sulogin(1M)

/etc/sulogin(1M) is intended to be invoked by **init** when the root user invokes **init** to enter single user mode. This command should never be invoked directly from the command line.

When invoked by the root from the command line, **/etc/sulogin** will invoke a second shell for the root user but will not place the system in maintenance mode.

When **/etc/sulogin** is run by a normal user from the command line, the following error message is printed:

> **** NO ENTRY FOR root IN SHADOW FILE! ****
> Entering system maintenance mode".

This command does nothing except display the error message which should be ignored.

System Startup

If */tmp* is a separate mount point, then */tmp* is never cleaned up when a reboot occurs. This may lead to wasted disk space. This change is only necessary if the machine is configured with the tmp file systems (*/tmp* and */usr/tmp*) on separate file systems.

Workaround: The following modification should be made to the */etc/init.d/RMTMPFILES* file by root.

- If */tmp* is a separate file system, add the following two lines after the `chown sys /tmp` line:

```
        else
                rm -rf /tmp/*
```

- If */usr/tmp* is a separate file system, add the following two lines after the `chown sys /usr/tmp` line:

```
        else
                rm -rf /usr/tmp/*
```

kernel

The kernel will not rebuild properly if "weitek" is turned off in */etc/conf/cf.d/sdevice*. If you try to rebuild the kernel using the **idbuild**(1M) command, the build will fail with the following symbols undefined:

get87
weitek_intr
weitek_reg

Workaround: Edit */etc/conf/cf.d/sdevice* and search for "weitek". Change the second field on the line from "N" to "Y". Write the file and quit the editor. Then rebuild the kernel using the **/etc/conf/bin/idbuild**(1M) command. This will turn *Weitek* back on.

If it is absolutely necessary to turn off *Weitek,* these lines must be added to the */etc/conf/pack.d/weitek/stubs.c* file. Immediately after the line

```
/* Weitek stubs */
```

add the following lines:

```
int  get87() {}
weitek_intr() {}
char *weitek_reg = (char *)0;
```

Installation and backup(1M)

A complete or partial system **backup**(1M) will back up Foundation Set add-on packages in addition to the user files. This is not a serious problem, but may result in a new package being overwritten by an older version if a complete restore is done. The new package would then have to be reinstalled. It also causes the backup to take more time and to use more media than simply backing up user files.

Workaround: If a complete restore is to be done, it should be done immediately after the Base System is installed, and before any of the Foundation Set add-on packages are installed. It may be necessary to use **removepkg**(1M) to remove an old package before the new version can be installed.

If a selective restore is done, and if only user files are to be restored, the order of installation is not important. However, it is still advisable that the restore be done before add-on packages are installed.

backup(1M) and restore(1M)

If you are doing a complete restore of a backup, it is not advisable to use the overwrite option since old versions of files may overwrite new versions. If you want to overwrite, do a selective restore so that you know exactly which files are to be overwritten.

If, when doing a complete or partial backup, the number of floppies needed appear to be unnecessarily large, press "delete" and remove the files */etc/.lastbackup* and */etc/.lastpartial*. A complete backup should be started.

Enhanced EGA Support

Some video boards do not work correctly. The Zenith/Heath "Enhanced" EGA is an example of such a board not supported by the operating system.

Application Installation

The normal XENIX SGS is not present. Installation procedures that call the XENIX System language tools directly must be modified to use the UNIX System tools. Application installation procedures that must use the XENIX System linkage editor to link pre-existing object modules will not work.

System Startup (/etc/rc files)

There is no longer a single */etc/rc* file. Installation procedures that attempt to modify */etc/rc* or */etc/rc.local* must be modified to alter the appropriate *rc* file for the appropriate run level.

Uid for bin

The numeric uid for the user *bin*, and the numeric gid for the group *bin*, have changed. Installation scripts which depend on the old values must be changed to accept the new values. The values for the XENIX System were 3 and 3; the values are now 2 and 2.

ls(1)

The UNIX System command **ls** and the XENIX System command **ls** differ in the meaning of the size of a file in blocks (as provided by the **-s** option). The XENIX System understands that all filesystems have a block size of 1024 bytes, and thus correctly calculates the number of direct and indirect blocks. The UNIX System assumes a 512-byte block size when making the calculation.

passwd(1)

For enhanced security, encrypted passwords have been moved out of the file */etc/passwd*. Installation procedures that examine or modify the password file directly must be modified to use the new scheme. Under the new scheme, user passwords and aging information are stored in */etc/shadow*. Refer to Chapter 5 of the *Operations/System Administrator's Guide* for more information.

uname(1)

The default settings for the fields of the **uname** structure are different. This will result in different output for the **-m** and **-s** options of the **uname** command. Installation procedures that use the output of the **uname** command to determine machine type or system type must be modified to handle the new output correctly. Under the XENIX System, the "sys" entry defaults to "XENIX", and the "cpu" field defaults to "iAPX386". Under the UNIX System, the respective entries default to "unix" and "i386". Under both XENIX System and UNIX System, the remaining entries have release-dependent defaults.

Differences in Support of XENIX-286 Execution

XENIX-286 Emulation

Support of XENIX-286 executables is now handled by a user-level emulator, rather than by the kernel. As a result, attempts to execute a text file that is open for writing (which fail with the error ETXTBSY under the XENIX System) will succeed, and the emulator will then refuse to run the executable.

init(1M)

The shell that is provided when the user puts the system in single-user mode (via "init" s) has a useless path. Immediately after entering single-user mode, the shell search path should be set to whatever the user wants.

mountall(1M)

The documentation for the **mountall** command states that the command can take a list of files (or "–" for stdin) as command line arguments. This is not true. The **mountall** command ignores its arguments, and always reads the file */etc/fstab*.

nlsadmin(1M)

In the *Operations/Systems Administration Guide*, the RFS chapter states that to start the Starlan listener, the command is:

```
nlsadmin -S starlan
```

The correct command is:

```
nlsadmin -s starlan
```

uname(1)

In the *Operations/Systems Administration Guide*, the RFS chapter states that typing the command

```
uname -S nodename
```

will result in the user being prompted for the new name. In fact, the system's name is set to *nodename*.

Future Directions

awk, nawk, oawk

With Release 3.1 there is a new **awk** (**nawk**). For this release **awk** is linked to **oawk** and is the default when you type **awk**. In the next major release of UNIX System V, **nawk** will be the default and will be linked to **awk**, but **oawk** will still be available.

NOTE **nawk** must be used with the international version of the software.

getdents(2)

The implementation of **getdents**(2) does not match the description in the **dirent**(4) manual page. The field **d_off** in **struct dirent** does not contain the file offset of the current directory entry, but rather the file offset of the following entry. This will be corrected in the next major UNIX System V release. The correction may require the re-compilation or re-linking of programs using the directory-management library routines described in **directory**(3X) (**opendir, closedir, readdir, telldir, seekdir, rewinddir**) and may require source changes to programs using the **getdents**(2) system call directly.

Regular Expressions

In Release 3.1 the implementation of regular expressions (for example in **ex, egrep, regexp.h,** and **sh**) has been extended to support 8-bit characters. The semantics for the range notation, which currently uses ordinal values for the character, are the same as in previous releases. This permits ranges to include: 7-bit characters, 8-bit characters, and 7- and 8-bit characters.

Remote File Sharing Notes

Software Description

Remote File Sharing (RFS) Release 1.2 is a software package that allows computers running AT&T 386 UNIX System V Release 3.1 or later releases to share resources (directories containing files, subdirectories, devices, and named pipes) selectively across a network. Administrators for computers on an RFS network can choose directories on their systems they want to share and add them to a list of available resources on the network. From this list, they can choose resources from remote computers that they would like to use on their computers.

Each computer on a Remote File Sharing system can be grouped with others in a "domain" or can operate as an independent domain. The domain can provide a central point for administering a group of computers. Unlike other distributed file systems used with the UNIX operating system, Remote File Sharing is built into the operating system. This approach has several advantages:

Compatibility — Once you mount a remote resource on your system, it will look to your users as though it is part of the local system. You will be able to use most standard UNIX System features on the resource. Standard commands and system calls, as well as features such as File and Record Locking, work the same on remote resources as they do locally. Applications should be able to work on remote resources without modification.

Security — Standard UNIX System file security measures will be available to protect your resources. Special means for verifying computers and restricting remote user permissions have been added for Remote File Sharing.

Flexibility — Since you can mount a remote resource on any directory on your system, you have a lot of freedom to set up your computer's view of the world. You do not have to open up all your files to every host on the network. Likewise, you do not have to make all files on the network available to users of your computer.

New Features

Two major features have been added to Remote File Sharing Release 1.2: client caching and loop-back.

Client Caching

The client caching feature of RFS provides substantial performance improvements over non-caching systems by reducing the number of times data must be read across the network. Client refers to the computer that is using a remote resource, while caching refers to the client's ability to store data in local buffer pools.

The first time a client process reads a block of data from a remote resource, it is placed in local buffer pools. Subsequent client processes reading a server file can avoid network access by finding the data already present in local buffers. This generally causes a large reduction in network messages, resulting in improved performance.

In order for client caching to work simply and reliably, the following features were built into it.

- **Cache consistency.** Checking mechanisms are used to ensure that the cache buffers accurately reflect the contents of the remote file the user is accessing.
- **Transparency.** The only difference users should see between caching and non-caching systems is improved response time. RFS-based applications do not have to be changed to run on a Remote File Sharing system that caches remote data.
- **Administration.** By default, client caching is on. However, options are available to turn off caching for an entire system or for a particular resource. (You would probably only do this if you have an application that does its own network buffering.) There are also some tunable parameters available to fine tune your system to the way you use RFS.

Loop-Back

The loop-back feature allows you to simulate the higher levels of RFS within one computer. For example, you could advertise a resource and mount it in a different location on the same machine.

The main use of the loop-back feature is to test application programs and give RFS demonstrations with only one computer. For information describing how to use this feature, see the **-o** option on the **rfadmin**(1M) manual page in the *User's/System Administrator's Reference Manual.*

Software Notes

This section describes problems that may occur with Remote File Sharing and, in some cases, workarounds to those problems.

acct

The accounting file passed to the **acct**(2) system call cannot be remote. This restriction applies to user software that uses the system call directly and to the process accounting software. RFS does not allow the **acct** system call; if passed a remote path name, **acct** will return an errno of **EINVAL**.

chroot

If you use the **chroot** command to change to the *root* directory of a remote machine, the **ps** command will not work properly.

Client Caching

Reads and writes of block special files are not cached because they could duplicate other data in the cache (for example, data from a regular file residing on the block device defined by the special file). However, if you write to a block device special file on a server machine, the contents of regular files on the block device may be changed. Client-cache buffers associated with these files may be out of date. (Local disk accesses do not suffer from this problem, since block-device writes go through the local buffer pool).

Users who write to block special files on file systems that are advertised remotely in a way that affects the contents of regular files should turn off caching when the resource containing the device is first mounted. (See the **-c** option of **mount**(1M).)

df

If **df** is used without options, it will list each occurrence of a remote resource that is mounted on a system and place an asterisk next to the word blocks for the second and each subsequent resource that was advertised under the same remote file system (for example, */usr/mail* and */usr/bin*). This signifies that the identical block counts for the resources reside under the same file system.

The problem is that if **df** is used with multiple remote resources passed as arguments, the asterisk never appears. In this example, the two resources are on the same remote file system, though the asterisk does not appear:

```
$df USRMAIL USRBIN
/mnt/(USRMAIL)   30402 blocks
/mnt/(USRBIN)   30402 blocks
```

fumount

The **-w** option to the **fumount** command allows you to specify a grace period between warning clients that a resource is to be removed and actually removing the resource. The **atoi** subroutine [**strtol**(3C)] calculates the number of seconds. This routine looks for an initial numeric string and converts it to an integer. Any non-numeric character in the argument terminates the argument. For example, the argument **-w 123abc** gives a grace period of 123 seconds. Missing arguments and arguments without an initial numeric string produce an error message.

fuser

The **fuser** command does not find remote users with open local files. For example, mount a 1.2 Mbyte floppy file system as */tmp/a*. Advertise the */tmp* directory as resource **TMP**. Mount the **TMP** resource on a remote system and open a file residing in the *TMP/a* directory you just mounted. On the local system, run **fuser /dev/dsk/f0** (the floppy). The remote user will not be found.

fuser

fuser may miss a process if that process gets a reference to the resource after **fuser** has begun its search. In this case, the offending process can be killed explicitly with the **kill** command. When all processes using the resource are gone, the resource can be unmounted.

idload

Many ID mapping features do not function properly with the loop-back function. Only use **global** blocks of information in mapping files (**uid.rules** and **gid.rules**). Within global blocks only **default transparent** works as intended. Specific mapping (**map** lines) or attempts to use **host** blocks will result in users and groups being mapped to **60002**.

labelit

labelit performs a check to ensure that the destination device path begins with */dev/r??* for tape devices. However, remote tape devices on an RFS network are typically mounted under a local directory or in */dev*. As a result, **labelit** cannot be used over RFS for remote tape devices.

Logs

These log files may contain information relating to RFS activities:

/usr/adm/rfuadmin.log
/usr/adm/log/rfuadmin.log
/usr/net/servers/rfs/rfs.log
/usr/net/nls/netspec/log

These files are for internal use only! Customers should not rely on the contents of these files because the information may change or the file may be deleted in future releases. Any tool written that takes advantage of the information contained in these files is not guaranteed to work in the future. (In the list above, *netspec* is replaced by the transport provider used by RFS. For the STARLAN NETWORK, the *netspec* is **starlan**.) The *rfuadmin.log* files are NOT automatically truncated. You may want to monitor them to make sure they do not get too large.

lseek

Using **lseek** with a negative offset on a remote file behaves differently than on a local file. On a local file, the call fails and returns **EINVAL**, as it always has. However, on a remote file the call succeeds and returns the negative offset.

mount

When a mount fails because of a password mismatch, the error message can be confusing. The following error messages result from a remote mount failure due to mismatched passwords:

```
negotiate: An event requires attention
mount: negotiations failed
mount: possible cause: machine password incorrect
mount: could not connect to remote machine
```

mount

When a remote resource is disconnected by a **fumount**(1M) command or a broken link, the default action in the client **rfuadmin** script is to try to remount the resource as it was mounted before. Therefore, if a resource that was originally advertised by the server as read/write is readvertised read-only, the client's automatic mount will never succeed.

An administrator can always enter **mount** directly using the latest advertised mode.

mount

The programs that automatically remount remote resources (*/etc/rmount*, */etc/rmountall*, and */usr/nserve/rfuadmin*) do not recognize the **-c** option of the **/etc/mount** command as valid. The result is that when they attempt to remount a resource that was disconnected and that was originally mounted with client caching turned off [**mount**(1M) with the **-c** option], the resource is remounted with client caching turned on (the default).

One possible workaround for this problem is for the administrator to wait until the program that does the remount completes successfully, then to unmount and to remount the resource manually using the **mount -c** command. The administrator might also consider killing the process that does the remount, but there is no assurance that other actions that the process must perform will complete successfully.

mount

The **mount** command returns a different **ERRNO** and error message than it did in UNIX System V Release 3.0. The error occurs when you try to mount a local device on a subdirectory of a Remote File Sharing resource you mounted from another machine. The previous errno was **EMULTIHOP** and the error message was:

```
mount: Multihop attempted
mount: cannot mount device
```

The new errno is **EREMOTE** and the error message is:

```
mount: Object is remote
mount: cannot mount device
```

Name Server

When the primary and secondary name servers are under heavy load, the normal passing of name server information between these machines may cause them to hang because the 1K Streams buffers have been depleted. There is one long-term and one short-term solution to the problem.

For the long term, you can increase the number of 1K Streams buffers in */etc/conf/cf.d/mtune*. The parameter is **NBLK1024**. Increasing 2K Streams buffers (**NBLK2048** parameter) may also help because the system will try to use 2K buffers when 1K buffers are depleted.

The short-term solution is that you can stop Remote File Sharing on any secondary name server that is hung and then bring it back up again; that will clear the **NBLK1024** buffers.

nsquery

The resource list printed by **nsquery** does not always reflect the current state of the domain. If a resource is advertised and the server goes down, a subsequent **nsquery** from a client may still list the resource as being available, even though it is not. An attempt to mount the resource will fail, because it is unable to contact the server. You will have to wait for the remote machine to make the resource available again.

Programs

If a program creating remote directories or files loses its link to the remote machine, and the remote resource is unmounted, the program may begin to create local directories and files. For example, if you are using the **find** command piped to **cpio** to a remote machine and the link to the remote machine goes down and the resource is then unmounted, **cpio** may begin writing on the local machine—the target directory now looks just like an ordinary local directory.

Recovery

Remote File Sharing limits the size of a resource name to 14 characters. This is intended to limit only the resource name and not the domain name (if addressing a resource from another domain).

When a resource is mounted, the stated length of domain and resource name combinations can be used. However, when a link is broken to a server machine, the client's recovery mechanism tries to unmount the server resources. In doing so, it calculates the size of the resource name including the domain and the "." that separates them. If this string exceeds 14 characters, recovery fails to unmount the resource and may not complete. The result is that users will not be able to access the resource, but the resources will still be listed when you print the names of locally mounted resources (**mount** command).

This problem can be solved in one of two ways. If the problem occurs, the administrator can manually unmount the resource that is posing the problem and then recovery will complete properly (**umount -d** *resource*). To avoid the problem altogether, the administrator should use domain names and resource names that, when concatenated together, will not exceed a total length of 14 characters.

Recovery

If a client loses its link to a server, any attempt to **umount** one of that server's file systems from the client tree will fail until recovery runs. Recovery from a link failure is handled by **rfuadmin**(1M) and **rfudaemon**(1M).

Recovery runs automatically when the link breaks, but not until someone tries to access the link or until a maximum of 11 minutes have passed. (The 11-minute time interval applies if you are using STARLAN network. The time may be different for other transport providers.)

If the **umount** fails because the link is gone, the **umount** will start recovery. After recovery runs, a second **umount** will succeed.

rfadmin

The **rfadmin -p** command should not be used to pass name server responsibilities back to the primary while the RFS domain is in an unstable state. After RFS is restarted on the primary, the administrator on the secondary machine that is the current domain name server should wait until all requests to mount resources from the primary are completed. (When the primary went down **rmount** requests were placed in the background on all client machines that had mounted resources from the primary.) The secondary administrator can safely use **rfadmin -p** when the mount requests are completed.

rfadmin

The **-p** option of the **rfadmin** command returns a value of 2 if the command fails because no computer is available to take over primary name server responsibility.

In earlier releases, the **-p** option of the **rfadmin** command returned a 0 upon success or a 1 upon failure. In RFS Release 1.1, the **-p** option returns a 0 upon success, a 2 if the command fails because there is no computer available to assume primary name server responsibilities, or a 1 if the command fails for some other reason. This change gives the **rfstop**(1M) command the ability to warn the administrator if the machine being stopped will cause the domain to be without a primary name server.

rfmaster

The acting domain name server is responsible for distributing important name service information to all other accessible (secondary) name servers that are serving the same domain, with no more than a 15-minute lag, so that if the acting name server should fail, another host could assume the name server role with a minimal loss of information. However, changes to the *rfmaster* file after **rfstart** has been run are not included in the information that is distributed in this way. Because the designation of hosts as primary and secondary name servers is made in the *rfmaster* file, this has the consequence of not allowing a change to the configuration of which hosts are the primary and secondary name servers for a domain without stopping and re-starting RFS on the affected hosts. For example, adding a new secondary name server to the *rfmaster* file will not take effect until RFS is taken down on all of the existing (primary and secondary) name servers, as well as the newly designated secondary and then re-started.

This limitation should not be confused with the temporary transfer of name server responsibility to another one of the hosts already listed in the *rfmaster* file as a primary or secondary name server; this temporary transfer is performed with the **rfadmin -p** command.

rfmaster

When **rfstart** is executed, the name server process (**nserve**) scans the *rfmaster* file and reports possible errors. However, if there is an entry in the *rfmaster* file that identifies a computer as a name server (either primary or secondary), but there is not a corresponding address entry in the *rfmaster* file for that computer, it is not reported as an error, even though it should be. In attempting to contact a name server (for example, during **rfstart** or **rfadmin -p**), name servers with no address listed will be silently skipped.

rfpasswd

The **rfpasswd** command is used to change the host password used for RFS, and it is intended to parallel the **passwd** command in the way it prompts for old and new passwords. However, if a host has no password (for example, it has a null password), the **rfpasswd** command will still prompt for the old password before asking for the new one, although it should ask only for the new one.

rfstart

This problem only happens when you are issuing an **rfstart** command from one terminal and some other RFS command from another terminal.

Sometimes when you run the **rfstart** command it will prompt for a password. RFS has actually started when the password prompt occurs, so an **adv** command from another terminal would be accepted. After the password is entered, **rfstart** completes, which includes clearing the advertise table. The result is that the domain advertise table would show the advertised resource as available, while the local advertise table would not.

This problem can be avoided by not issuing additional RFS commands until the **rfstart** completes and exits to the shell. If the problem does occur, use the **adv -m** or **unadv** commands to put the two advertise tables back in agreement.

rfudaemon

User-level recovery of resources that are disconnected gracefully (the remote system shuts down) may fail if the number of lost resources exceeds half of the value of the tunable parameter **MAXGDP** in */etc/conf/cf.d/mtune* By default, **MAXGDP** is 24. The failure is accompanied by one or more of the following messages:

```
rfs user-daemon queue overflow:
make sure rfudaemon is running
```

STREAMS

The three system calls related to STREAMS—**getmsg**, **putmsg**, and **poll**—will not operate with a file descriptor associated with a remote file. If this is attempted, the system call will fail with errno equal to **ENOSTR**.

swap

Swap devices cannot be remote, including the swap device configured initially and any swap devices added using the **swap**(1M) command.

System Calls

When a Remote File Sharing server does not recognize an incoming request, it will return an **EREMOTE**. In Release 3.0, unrecognized system calls were ignored by the server.

Network Support Utilities Notes

Introduction

The Network Support Utilities (NSU) Release 1.2 package extends system capabilities to support networking applications. The package includes software support for STREAMS, the AT&T Transport Interface, and the Listener.

The Network Support Utilities package is required to take advantage of the following features of AT&T 386 UNIX System V Release 3.1 and later: the Remote File Sharing package, STREAMS mechanisms and tools, the AT&T Transport Interface, the enhanced Basic Networking Utilities, and the Listener.

STREAMS

STREAMS is a general, flexible facility for developing UNIX System communication services. By defining standard interfaces for character input/output within the kernel, STREAMS supports development ranging from complete networking protocol suites to individual device drivers. The standard interfaces and associated tools enable modular, portable development and easy integration of network services and their components—these were used to develop protocol modules and device drivers for Release 3. STREAMS provides a broad framework that does not impose any specific network architecture. It implements a user interface consistent and compatible with the character I/O mechanism that is also available in the UNIX System.

The power of STREAMS resides in its modularity. The design reflects the layering characteristics of contemporary networking architectures. Each basic component (called a module) in a STREAMS implementation represents a set of processing functions and communicates with other modules via a standard interface. From the user level, kernel resident modules can be dynamically selected and interconnected to implement any rational processing sequence. No additional kernel programming, assembly, or link editing is required. Modularity allows for the following advantages:

- User-level programs (commands such as **uucp**) can be independent of underlying protocols and communications media so the programs need not be changed when new media or protocols between systems become available.

- Network architectures and higher-level protocols are independent of underlying protocols, drivers, and media.
- Higher-level services can be created by selecting and connecting lower-level services and protocols.

In addition to the standard interfaces, STREAMS provides a set of software tools that help source customers build modules and drivers.

AT&T Transport Interface

With Release 3, UNIX System V supports a Transport Interface based on the Transport Service Definition (Level 4) of the International Organization for Standardization (ISO) Open Systems Interconnection (OSI) reference model. The transport service supports two modes of transfer: connection mode and connectionless mode. Connection mode is circuit-oriented and supports data transfer over an established connection in a reliable, sequenced manner. The connectionless mode is message-oriented (datagrams) and supports data transfer in self-contained units with no logical relationship required among units.

The AT&T Transport Interface defines how to accesses the services of a transport protocol, called a Transport Provider. An example of a Transport Provider is the ISO. Application programs access the Transport Provider by using the Transport Interface routines in the new Network Services Library. These routines support access to a Transport Provider in a media and protocol-independent manner. The Transport Provider uses kernel level programs to send the information to the desired physical device, such as the STARLAN Network Access Unit (NAU). By using the AT&T Transport Interface, application programs will be able to access other Transport Providers that may be available in the future.

For more information, see the *Network Programmer's Guide* and the *Programmer's Reference Manual.*

Listener

The "listener" is a program that can be used with Transport Providers on a system. The purpose of the listener is to receive requests for services from another system, interpret which service is needed, and start a process that has been named to provide the requested service. The listener then drops out of the communications path and continues to listen for new service requests.

For more information about the listener, see **nlsadmin**(1M) in the *User's/System Administrator's Reference Manual.* Also, see the Basic Network Utilities and Remote File Sharing sections of the *Operations/System Administration Guide* for information on how to set up the listener for these products.

Software Notes

This section describes problems that may occur with the Network Support Utilities and, in some cases, workarounds for these problems.

Listener

- In general, when the listener has trouble binding its listening addresses during its start sequence, the actual addresses bound will be chosen by the protocol and hence will differ from the ones requested. There are two cases where this can happen: when the address requested to be bound is either already bound for listening on that network, or when that address does not conform to whatever criteria the protocol is expecting for an address.

  In both of these cases, the listener will output a message to the console stating the address returned by the protocol and

  ```
  TLI bound a different name than requested.
  ```

  Additionally, the requested address appears in the listener's log file along with the messages sent to the console. The listener will then exit.

- The listener process can now parse quoted strings in the command line in the data base file.

  In the previous release, the listener did not understand quoted strings as arguments to servers. If quoted strings were required, it was necessary to specify a shell script as the server and embed the real server and the quoted string arguments there. This change eliminates the need for that indirection.

- The listener process now runs as **root** and sets both the user and group ID as specified by the user ID in the data base file.

  In the previous release, the listener ran as user ID **listen**. Because of this, servers that required special permissions needed to be owned by the appropriate user with the setuser or setgroup ID bits set accordingly. The proliferation of privileged servers was deemed to be a potential security hazard. This change allows the administrator to specify in the listener data base file the user ID under which the server should run:

nlsadmin -a *svc* **-c***cmd* **-w***id* **-y** *comment netspec*

The listener will perform **setuid** and **setgid** system calls based on this ID before **exec**'ing the server.

STREAMS

open

A race condition exists in clone opens from different inodes. This problem exists when two or more disk inodes with the major of the clone device and equal minors are being opened at the same time. If the window is hit, then another **open** after the first **open** may bypass the clone device entirely, thus failing. For example, if */dev/node1* was major 63 and minor 57, and */dev/node2* was also major 63 and minor 57, and if they were two different inodes, then simultaneous opens of the two devices may result in failure of the second open.

If two or more separate files are needed on disk, they should be created as links to one disk inode, thereby closing the window. In the previous example, */dev/node2* should be linked to */dev/node1* instead of being a separate inode.

read

For a STREAMS file, when a message containing 1 or more bytes is read and the first message block of the message contains 0 bytes, the message will be erroneously treated as a 0 byte message.

write

For a STREAMS file, if **write** is interrupted by a signal after some data have been written, it should return the number of bytes written. However, it returns -1 and sets errno to **EINTR**.

Even when there are no flow control restrictions, **write** will block if STREAMS internal resources are not available, regardless of the state of O_NDELAY.

STREAMS Programmer's Guide

In Appendix C, the initial lines of **qenable** should read as follows:

> **qenable** – enable a queue
>
> **int qenable(q)**
> **queue_t *q;**
> **qenable** places the queue pointed at by *q* ...

Transport Interface Library

Network Programmer's Guide

The reference to **tirdwr**(7) in the caution note on Page 5-2 should reflect the change (in the manual page) that popping the module from a *stream* no longer causes a disconnect.

t_accept

The description of **TBADDATA** has been modified to clarify the condition under which it occurs.

[TBADDATA] The amount of user data specified exceeds the allowed amount as returned in the **info.connect** field by **t_open** or **t_getinfo**.

t_connect

The description of **TBADDATA** has been modified to clarify the condition under which it occurs.

[TBADDATA] The amount of user data specified exceeds the allowed amount as returned in the **info.connect** field by **t_open** or **t_getinfo**.

t_getstate

t_getstate will not work following an **exec**(2)/**t_sync**(3N) call sequence because there is insufficient information available to determine the state. The problem arises because the state is maintained in a user process data space, but an **exec** will overwrite that data space and the state information will be lost. If **t_getstate** is called in this case, it will fail with **t_error** set to **TSTATECHNG**.

t_rcv

The processing of expedited transport data units (ETSDU) by **t_rcv** has been modified. The function will now set the **T_EXPEDITED** flag on return, if the data are part of an expedited message. The sentence in the fourth paragraph of the DESCRIPTION now reads:

> Subsequent calls to retrieve the remaining ETSDU will have **T_EXPEDITED** set on return.

t_snddis

The description of **TBADDATA** has been modified to clarify the condition under which it occurs.

[TBADDATA] The amount of user data specified exceeds the allowed amount as returned in the **info.discon** field by **t_open** or **t_getinfo**.

t_sndudata

The paragraph describing the **EPROTO** error has been expanded as follows:

> If **t_sndudata** is issued from an invalid state, or if the amount of data specified in **udata** exceeds the **TSDU** size as returned by **t_open** or **t_getinfo**, the provider will generate an **EPROTO** protocol error (see **TSYSERR** below). However, the **t_sndudata** may not fail because **EPROTO** errors may not be reported immediately. In this case, a subsequent call that accesses the transport endpoint will fail with the associated **TSYSERR**.

t_sync

The description of the **TBADF** value **t_errno** has been changed to the following:

[TBADF] The specified file descriptor does not refer to a transport endpoint.

Appendix A: Installation Diskette Files

Base System Package, Diskette 1 of 7

INSTALL
INSTALL2
LABEL
bin
bin/-sh
bin/chgrp
bin/chmod
bin/chown
bin/cp
bin/cpio
bin/dd
bin/ed
bin/expr
bin/find
bin/ln
bin/mkdir
bin/mv
bin/rm
bin/sh
bin/sleep
bin/stty
bin/su
bin/sync
dev
dev/console
dev/dsk
dev/dsk/0s0
dev/dsk/0s1
dev/dsk/0s2
dev/dsk/0s3
dev/dsk/0s4
dev/dsk/1s0
dev/dsk/1s1
dev/dsk/1s2
dev/dsk/1s3
dev/dsk/1s4
dev/dsk/f0
dev/dsk/f0q15d
dev/dsk/f0q15dt
dev/dsk/f0t
dev/fd0
dev/fd096
dev/fd096ds15
dev/hd00
dev/hd01
dev/hd02
dev/hd10
dev/hd11
dev/hd12
dev/hd13
dev/hd14
dev/install
dev/mem
dev/null
dev/pmem
dev/rdsk
dev/rdsk/0s0
dev/rdsk/0s1
dev/rdsk/0s2
dev/rdsk/0s3
dev/rdsk/0s4
dev/rdsk/1s0
dev/rdsk/1s1
dev/rdsk/1s2
dev/rdsk/1s3
dev/rdsk/1s4
dev/rdsk/f0
dev/rdsk/f0q15d

```
dev/rdsk/f0q15dt
dev/rdsk/f0t
dev/rfd0
dev/rfd096
dev/rfd096ds15
dev/rhd00
dev/rhd01
dev/rhd02
dev/rhd10
dev/rhd11
dev/rhd12
dev/rhd13
dev/rhd14
dev/rinstall
dev/root
dev/rroot
dev/rswap
dev/swap
dev/syscon
dev/systty
dev/tty
dev/vt00
dev/vt01
dev/vt02
dev/vt03
dev/vt04
dev/vt05
dev/vt06
dev/vt07
dev/vtmon
etc
etc/.packagedate
etc/TIMEZONE
etc/boot
etc/default
etc/default/boot
etc/default/default.at386
etc/default/default.att
etc/default/default.att512
etc/default/default.cpq
etc/disksetup
etc/emulator
etc/emulator.dflt
etc/fdisk
etc/fixswap
etc/fsck
etc/group
etc/init
etc/initprog
etc/initprog/at386
etc/initprog/att
etc/initprog/compaq
etc/inittab
etc/inittab2
etc/inittab3
etc/ioctl.syscon
etc/labelit
etc/machine_type
etc/memsize
etc/mkfs
etc/mknod
etc/mkpart
etc/mnttab
etc/mount
etc/passwd
etc/sulogin
etc/uadmin
etc/umount
mnt
readfloppy
shlib
shlib/libc_s
shlib/libnsl_s
tmp
unix
yes
```

Base System Package, Diskette 2 of 7

bin
bin/acctcom
bin/ar
bin/basename
bin/cat
bin/clear
bin/cmp
bin/copy
bin/crypt.nf
bin/csh
bin/date
bin/df
bin/diff
bin/dirname
bin/du
bin/echo
bin/env
bin/false
bin/file
bin/format
bin/grep
bin/hd
bin/i286emul
bin/idas
bin/idld
bin/ipcrm
bin/ipcs
bin/kill
bin/line
bin/login
bin/ls
bin/mail
bin/mesg
bin/mt
bin/newgrp
bin/nice
bin/nohup
bin/passwd
bin/pr
bin/ps
bin/pwd
bin/rmdir
bin/sed
bin/setpgrp
bin/sort
bin/strings
bin/su
bin/tail
bin/tee
bin/test
bin/time
bin/touch
bin/true
bin/tset
bin/tty
bin/uname
bin/wc
bin/who
bin/write
bin/x286emul
bin/xrestor
bin/xrestore
bin/yes
dev
dev/cram
dev/dsk/0s5
dev/dsk/0s6
dev/dsk/1s5
dev/dsk/1s6
dev/dsk/f03d
dev/dsk/f03dt
dev/dsk/f03h
dev/dsk/f03ht
dev/dsk/f05d16

```
dev/dsk/f05d16t
dev/dsk/f05d4
dev/dsk/f05d4t
dev/dsk/f05d8u
dev/dsk/f05q
dev/dsk/f05qt
dev/dsk/f0d8d
dev/dsk/f0d8dt
dev/dsk/f0d9d
dev/dsk/f0d9dt
dev/fd048ds8
dev/fd048ds9
dev/hd03
dev/hd04
dev/hd13
dev/hd14
dev/dsk/f1
dev/dsk/f13d
dev/dsk/f13dt
dev/dsk/f13h
dev/dsk/f13ht
dev/dsk/f15d16
dev/dsk/f15d16t
dev/dsk/f15d4
dev/dsk/f15d4t
dev/dsk/f15d8u
dev/dsk/f15q
dev/dsk/f15qt
dev/dsk/f1d8d
dev/dsk/f1d8dt
dev/dsk/f1d9d
dev/dsk/f1d9dt
dev/dsk/f1q15d
dev/dsk/f1q15dt
dev/dsk/f1t
dev/fd048
dev/fd1
dev/fd148
dev/fd148ds8
dev/fd148ds9
dev/fd196
dev/fd196ds15
dev/install1
dev/kmem
dev/lp
dev/lp0
dev/lp1
dev/lp2
dev/prf
dev/rdsk/0s5
dev/rdsk/0s6
dev/rhd03
dev/rhd04
dev/rdsk/1s5
dev/rdsk/1s6
dev/rdsk/f03d
dev/rdsk/f03dt
dev/rdsk/f03h
dev/rdsk/f03ht
dev/rdsk/f05d16
dev/rdsk/f05d16t
dev/rdsk/f05d4
dev/rdsk/f05d4t
dev/rdsk/f05d8u
dev/rdsk/f05q
dev/rdsk/f05qt
dev/rdsk/f0d8d
dev/rdsk/f0d8dt
dev/rdsk/f0d9d
dev/rdsk/f0d9dt
dev/rdsk/f1
dev/rdsk/f13d
dev/rdsk/f13dt
dev/rdsk/f13h
dev/rdsk/f13ht
dev/rdsk/f15d16
dev/rdsk/f15d16t
dev/rdsk/f15d4
dev/rdsk/f15d4t
dev/rfd048
```

dev/rfd048ds8
dev/rfd048ds9
dev/rhd13
dev/rhd14
dev/rinstall1
dev/clock
dev/rdsk/f15d8u
dev/rdsk/f15q
dev/rdsk/f15qt
dev/rdsk/f1d8d
dev/rdsk/f1d8dt
dev/rdsk/f1d9d
dev/rdsk/f1d9dt
dev/rdsk/f1q15d
dev/rdsk/f1q15dt
dev/rdsk/f1t
dev/rfd1
dev/rfd148
dev/rfd148ds8
dev/rfd148ds9
dev/rfd196
dev/rfd196ds15
dev/root
dev/rtc
dev/sxt
dev/sxt/000
dev/sxt/001
dev/sxt/002
dev/sxt/003
dev/sxt/004
dev/sxt/005
dev/sxt/006
dev/sxt/007
dev/sxt/010
dev/sxt/011
dev/sxt/012
dev/sxt/013
dev/sxt/014
dev/sxt/015
dev/sxt/016
dev/sxt/017
dev/sxt/020
dev/sxt/021
dev/sxt/022
dev/sxt/023
dev/sxt/024
dev/sxt/025
dev/sxt/026
dev/sxt/027
dev/sxt/030
dev/sxt/031
dev/sxt/032
dev/sxt/033
dev/sxt/034
dev/sxt/035
dev/sxt/036
dev/sxt/037
dev/sxt/040
dev/sxt/041
dev/sxt/042
dev/sxt/043
dev/sxt/044
dev/sxt/045
dev/sxt/046
dev/sxt/047
dev/sxt/050
dev/sxt/051
dev/sxt/052
dev/sxt/053
dev/sxt/054
dev/sxt/055
dev/sxt/056
dev/sxt/057
dev/sxt/060
dev/sxt/061
dev/sxt/062
dev/sxt/063
dev/sxt/064
dev/sxt/065
dev/sxt/066

dev/sxt/067
dev/sxt/070
dev/sxt/071
dev/sxt000
dev/sxt001
dev/sxt002
dev/sxt003
dev/sxt004
dev/sxt005
dev/sxt006
dev/sxt007
dev/sxt010
dev/sxt011
dev/sxt012
dev/sxt013
dev/sxt014
dev/sxt015
dev/sxt016
dev/sxt017
dev/sxt020
dev/sxt021
dev/sxt022
dev/sxt023
dev/sxt024
dev/sxt025
dev/sxt026
dev/sxt027
dev/sxt030
dev/sxt031
dev/sxt032
dev/sxt033
dev/sxt034
dev/sxt035
dev/sxt036
dev/sxt037
dev/sxt040
dev/sxt041
dev/sxt042
dev/sxt043
dev/sxt044
dev/sxt045
dev/sxt046
dev/sxt047
dev/sxt050
dev/sxt051
dev/sxt052
dev/sxt053
dev/sxt054
dev/sxt055
dev/sxt056
dev/sxt057
dev/sxt060
dev/sxt061
dev/sxt062
dev/sxt063
dev/sxt064
dev/sxt065
dev/sxt066
dev/sxt067
dev/sxt070
dev/sxt071
dev/tty
dev/tty00
dev/xt
dev/xt/000
dev/xt/001
dev/xt/002
dev/xt/003
dev/xt/004
dev/xt/005
dev/xt/006
dev/xt/007
dev/xt000
dev/xt001
dev/xt002
dev/xt003
dev/xt004
dev/xt005
dev/xt006
dev/xt007

etc
etc/.cpiopc
etc/.fboot
etc/adduser
etc/bcheckrc
etc/brc
etc/checklist
etc/chroot
etc/ckbupscd
etc/cleanup
etc/clri
etc/conf
etc/conf/bin
etc/conf/bin/idbuild
etc/conf/bin/idcheck
etc/conf/bin/idconfig
etc/conf/bin/idinstall
etc/conf/bin/idmkenv
etc/conf/bin/idreboot
etc/conf/bin/idtune
etc/conf/cf.d
etc/conf/cf.d/init.base

Base System Package, Diskette 3 of 7

etc/conf/bin/idmaster
etc/conf/bin/idmkinit
etc/conf/bin/idmknod
etc/conf/bin/idmkunix
etc/conf/bin/idspace
etc/conf/cf.d/mdevice
etc/conf/cf.d/mfsys
etc/conf/cf.d/mtune
etc/conf/cf.d/sassign
etc/conf/cf.d/sdevice
etc/conf/cf.d/sfsys
etc/conf/cf.d/stune
etc/conf/cf.d/vuifile
etc/conf/init.d
etc/conf/mfsys.d
etc/conf/mfsys.d/s5
etc/conf/node.d
etc/conf/node.d/asy
etc/conf/node.d/lp
etc/conf/pack.d
etc/conf/pack.d/asy
etc/conf/pack.d/asy/Driver.o
etc/conf/pack.d/asy/space.c
etc/conf/pack.d/clock
etc/conf/pack.d/clock/space.c
etc/conf/pack.d/cpyrt
etc/conf/pack.d/cpyrt/Driver.o
etc/conf/pack.d/cpyrt/space.c
etc/conf/pack.d/cram
etc/conf/pack.d/cram/Driver.o
etc/conf/pack.d/disp
etc/conf/pack.d/disp/Driver.o
etc/conf/pack.d/disp/space.c
etc/conf/pack.d/du
etc/conf/pack.d/du/stubs.c
etc/conf/pack.d/fd
etc/conf/pack.d/fd/Driver.o
etc/conf/pack.d/fp
etc/conf/pack.d/fp/Driver.o
etc/conf/pack.d/gentty
etc/conf/pack.d/gentty/Driver.o
etc/conf/pack.d/hd
etc/conf/pack.d/hd/Driver.o
etc/conf/pack.d/ipc
etc/conf/pack.d/ipc/Driver.o
etc/conf/pack.d/kd
etc/conf/pack.d/kd/Driver.o
etc/conf/pack.d/kd/space.c
etc/conf/pack.d/kernel
etc/conf/pack.d/kernel/io.o
etc/conf/pack.d/kernel/locore.o
etc/conf/pack.d/kernel/os.o
etc/conf/pack.d/kernel/space.c
etc/conf/pack.d/kernel/start.o
etc/conf/pack.d/lp
etc/conf/pack.d/lp/Driver.o
etc/conf/pack.d/mem
etc/conf/pack.d/mem/Driver.o
etc/conf/pack.d/msg
etc/conf/pack.d/msg/Driver.o
etc/conf/pack.d/msg/space.c
etc/conf/pack.d/msg/stubs.c
etc/conf/pack.d/nmi
etc/conf/pack.d/nmi/Driver.o
etc/conf/pack.d/osm
etc/conf/pack.d/osm/Driver.o
etc/conf/pack.d/pic
etc/conf/pack.d/pic/space.c
etc/conf/pack.d/prf
etc/conf/pack.d/prf/Driver.o
etc/conf/pack.d/prf/space.c
etc/conf/pack.d/prf/stubs.c
etc/conf/pack.d/rtc
etc/conf/pack.d/rtc/Driver.o

etc/conf/pack.d/s5
etc/conf/pack.d/s5/Driver.o
etc/conf/pack.d/s52k
etc/conf/pack.d/s52k/stubs.c
etc/conf/pack.d/sem
etc/conf/pack.d/sem/Driver.o
etc/conf/pack.d/sem/space.c
etc/conf/pack.d/sem/stubs.c
etc/conf/pack.d/shm
etc/conf/pack.d/shm/Driver.o
etc/conf/pack.d/shm/space.c
etc/conf/pack.d/shm/stubs.c
etc/conf/pack.d/sxt
etc/conf/pack.d/sxt/Driver.o
etc/conf/pack.d/sxt/space.c
etc/conf/pack.d/sxt/stubs.c
etc/conf/pack.d/vx
etc/conf/pack.d/vx/Driver.o
etc/conf/pack.d/vx/space.c
etc/conf/pack.d/vx/stubs.c
etc/conf/pack.d/weitek
etc/conf/pack.d/weitek/Driver.o
etc/conf/pack.d/weitek/space.c
etc/conf/pack.d/weitek/stubs.c
etc/conf/pack.d/xsd
etc/conf/pack.d/xsd/Driver.o
etc/conf/pack.d/xsd/space.c
etc/conf/pack.d/xsd/stubs.c
etc/conf/pack.d/xsem
etc/conf/pack.d/xsem/Driver.o
etc/conf/pack.d/xsem/space.c
etc/conf/pack.d/xsem/stubs.c
etc/conf/pack.d/xt
etc/conf/pack.d/xt/Driver.o
etc/conf/pack.d/xt/space.c
etc/conf/pack.d/xt/stubs.c
etc/conf/rc.d
etc/conf/sd.d
etc/conf/sdevice.d
etc/conf/sdevice.d/asy
etc/conf/sdevice.d/cpyrt
etc/conf/sdevice.d/cram
etc/conf/sdevice.d/du
etc/conf/sdevice.d/fd
etc/conf/sdevice.d/fp
etc/conf/sdevice.d/gentty
etc/conf/sdevice.d/hd
etc/conf/sdevice.d/ipc
etc/conf/sdevice.d/kd
etc/conf/sdevice.d/lp
etc/conf/sdevice.d/mem
etc/conf/sdevice.d/msg
etc/conf/sdevice.d/nmi
etc/conf/sdevice.d/osm
etc/conf/sdevice.d/prf
etc/conf/sdevice.d/rtc
etc/conf/sdevice.d/s52k
etc/conf/sdevice.d/sem
etc/conf/sdevice.d/shm
etc/conf/sdevice.d/sxt
etc/conf/sdevice.d/vx
etc/conf/sdevice.d/weitek
etc/conf/sdevice.d/xsd
etc/conf/sdevice.d/xsem
etc/conf/sdevice.d/xt
etc/conf/sfsys.d
etc/conf/sfsys.d/s5
etc/crash
etc/cron
etc/cshrc
etc/custom
etc/dcopy1K
etc/dcopy2K
etc/dcopy512
etc/default
etc/default/login
etc/default/su
etc/default/tar
etc/default/xrestor
etc/deluser

etc/dfsck
etc/dfspace
etc/diskadd
etc/dumpsave
etc/ff
etc/filesave
etc/fixperm
etc/fsanck
etc/fsck2K
etc/fsck512
etc/fsdb
etc/fsstat
etc/fstab
etc/fstyp
etc/fstyp.d
etc/fstyp.d/S51Kfstyp
etc/fstyp.d/S52Kfstyp

Base System Package, Diskette 4 of 7

etc/fsdb2K
etc/fsdb512
etc/fstyp.d/xxfstyp
etc/fuser
etc/getclk
etc/getty
etc/gettydefs
etc/grpck
etc/idrc.d
etc/idsd.d
etc/init.d
etc/init.d/README
etc/init.d/RMTMPFILES
etc/init.d/disks
etc/init.d/firstcheck
etc/init.d/lp
etc/issue
etc/killall
etc/ldsysdump
etc/link
etc/links
etc/magic
etc/mkfs2K
etc/mkfs512
etc/motd
etc/mountall
etc/mvdir
etc/ncheck
etc/perms
etc/prfdc
etc/prfld
etc/prfpr
etc/prfsnap
etc/prfstat
etc/profile
etc/pwck
etc/rc0
etc/rc0.d
etc/rc0.d/K00ANNOUNCE
etc/rc0.d/K70uucp
etc/rc0.d/K75cron
etc/rc1
etc/rc2
etc/rc2.d
etc/rc2.d/S01MOUNTFSYS
etc/rc2.d/S05RMTMPFILES
etc/rc2.d/S20sysetup
etc/rc2.d/S21perf
etc/rc3
etc/rc3.d
etc/rstab
etc/setclk
etc/setmnt
etc/shutdown
etc/stdprofile
etc/sulogin
etc/swap
etc/sysdef
etc/termcap
etc/ttytype
etc/umountall
etc/unlink
etc/volcopy
etc/wall
etc/whodo
etc/xinstall
lib
lib/idcomp
lib/idcpp
usr
usr/adm
usr/adm/acct
usr/adm/acct/fiscal
usr/adm/acct/nite

usr/adm/acct/sum
usr/adm/sa
usr/bin
usr/bin/300
usr/bin/300s
usr/bin/4014
usr/bin/450
usr/bin/adv.nf
usr/bin/asa
usr/bin/at
usr/bin/awk
usr/bin/backup
usr/bin/banner
usr/bin/batch
usr/bin/bc
usr/bin/bdiff.nf
usr/bin/cal
usr/bin/calendar
usr/bin/cancel
usr/bin/captoinfo
usr/bin/checkeq
usr/bin/chrtbl
usr/bin/cpset
usr/bin/crontab
usr/bin/ct
usr/bin/cu
usr/bin/cut
usr/bin/displaypkg
usr/bin/factor
usr/bin/greek
usr/bin/installpkg

Base System Package, Diskette 5 of 7

usr/bin/dc
usr/bin/disable
usr/bin/enable
usr/bin/getopt
usr/bin/graph
usr/bin/hp
usr/bin/id
usr/bin/infocmp
usr/bin/ismpx
usr/bin/join
usr/bin/jterm
usr/bin/jwin
usr/bin/layers
usr/bin/logname
usr/bin/lp
usr/bin/lpstat
usr/bin/mailx
usr/bin/mcs
usr/bin/message
usr/bin/more
usr/bin/news
usr/bin/nlsadmin.nf
usr/bin/oawk
usr/bin/passmgmt
usr/bin/paste
usr/bin/pg
usr/bin/pwconv
usr/bin/pwdmenu
usr/bin/random
usr/bin/removepkg
usr/bin/restore
usr/bin/sag
usr/bin/sar
usr/bin/shl
usr/bin/spline
usr/bin/tabs
usr/bin/tar
usr/bin/tic
usr/bin/timex
usr/bin/tplot
usr/bin/tput
usr/bin/tr
usr/bin/units
usr/bin/uucp
usr/bin/uulog
usr/bin/uuname
usr/bin/uupick
usr/bin/uustat
usr/bin/uuto
usr/bin/xargs

Base System Package, Diskette 6 of 7

usr/bin/uux
usr/bin/xtd
usr/bin/xtract
usr/bin/xts
usr/bin/xtt
usr/include
usr/include/agent.h
usr/include/pn.h
usr/include/shadow.h
usr/include/sys
usr/include/sys/acct.h
usr/include/sys/adv.h
usr/include/sys/alttbl.h
usr/include/sys/ascii.h
usr/include/sys/asy.h
usr/include/sys/at_ansi.h
usr/include/sys/bitmasks.h
usr/include/sys/bootinfo.h
usr/include/sys/buf.h
usr/include/sys/callo.h
usr/include/sys/cdump.h
usr/include/sys/cirmgr.h
usr/include/sys/clock.h
usr/include/sys/clockcal.h
usr/include/sys/cmn_err.h
usr/include/sys/comm.h
usr/include/sys/conf.h
usr/include/sys/cram.h
usr/include/sys/crtctl.h
usr/include/sys/debug.h
usr/include/sys/debugreg.h
usr/include/sys/dir.h
usr/include/sys/dirent.h
usr/include/sys/dma.h
usr/include/sys/elog.h
usr/include/sys/emap.h
usr/include/sys/erec.h
usr/include/sys/err.h
usr/include/sys/errno.h
usr/include/sys/fblk.h
usr/include/sys/fcntl.h
usr/include/sys/fd.h
usr/include/sys/fdisk.h
usr/include/sys/file.h
usr/include/sys/filsys.h
usr/include/sys/flock.h
usr/include/sys/fp.h
usr/include/sys/fs
usr/include/sys/fs/prfcntl.h
usr/include/sys/fs/s5dir.h
usr/include/sys/fs/s5fblk.h
usr/include/sys/fs/s5filsys.h
usr/include/sys/fs/s5inode.h
usr/include/sys/fs/s5macros.h
usr/include/sys/fs/s5param.h
usr/include/sys/fsid.h
usr/include/sys/fstyp.h
usr/include/sys/gate.h
usr/include/sys/gdpstr.h
usr/include/sys/getpages.h
usr/include/sys/hd.h
usr/include/sys/hetero.h
usr/include/sys/idtab.h
usr/include/sys/immu.h
usr/include/sys/inline.h
usr/include/sys/ino.h
usr/include/sys/inode.h
usr/include/sys/iobuf.h
usr/include/sys/ioctl.h
usr/include/sys/ipc.h
usr/include/sys/ipl.h
usr/include/sys/istk.h
usr/include/sys/ivlab.h
usr/include/sys/jioctl.h

usr/include/sys/kd.h
usr/include/sys/lock.h
usr/include/sys/locking.h
usr/include/sys/lp.h
usr/include/sys/macro.h
usr/include/sys/map.h
usr/include/sys/message.h
usr/include/sys/mount.h
usr/include/sys/msg.h
usr/include/sys/nami.h
usr/include/sys/nserve.h
usr/include/sys/open.h
usr/include/sys/page.h
usr/include/sys/param.h
usr/include/sys/pfdat.h
usr/include/sys/pic.h
usr/include/sys/pit.h
usr/include/sys/poll.h
usr/include/sys/proc.h
usr/include/sys/proctl.h
usr/include/sys/que.h
usr/include/sys/queue.h
usr/include/sys/ramd.h
usr/include/sys/rbuf.h
usr/include/sys/rdebug.h
usr/include/sys/recover.h
usr/include/sys/reg.h
usr/include/sys/region.h
usr/include/sys/rfsys.h
usr/include/sys/rtc.h
usr/include/sys/sd.h
usr/include/sys/seg.h
usr/include/sys/sem.h
usr/include/sys/sema.h
usr/include/sys/shm.h
usr/include/sys/signal.h
usr/include/sys/stat.h
usr/include/sys/statfs.h
usr/include/sys/stermio.h
usr/include/sys/stream.h
usr/include/sys/stropts.h
usr/include/sys/strstat.h
usr/include/sys/swap.h
usr/include/sys/sxt.h
usr/include/sys/sysi86.h
usr/include/sys/sysinfo.h
usr/include/sys/sysmacros.h
usr/include/sys/systm.h
usr/include/sys/termio.h
usr/include/sys/tihdr.h
usr/include/sys/timeb.h
usr/include/sys/times.h
usr/include/sys/timod.h
usr/include/sys/tiuser.h
usr/include/sys/trace.h
usr/include/sys/trap.h
usr/include/sys/tss.h
usr/include/sys/ttold.h
usr/include/sys/tty.h
usr/include/sys/tuneable.h
usr/include/sys/types.h
usr/include/sys/uadmin.h
usr/include/sys/ulimit.h
usr/include/sys/user.h
usr/include/sys/ustat.h
usr/include/sys/utsname.h
usr/include/sys/v86.h
usr/include/sys/var.h
usr/include/sys/vt.h
usr/include/sys/vtoc.h
usr/include/sys/weitek.h
usr/include/sys/x.out.h
usr/include/sys/xdebug.h
usr/include/sys/xque.h
usr/include/sys/xt.h
usr/include/sys/xtproto.h
usr/include/tiuser.h
usr/include/windows.h
usr/lbin
usr/lbin/Install.sh

```
usr/lbin/fsinfo
usr/lbin/installpkg.r
usr/lbin/removepkg.r
usr/lib
usr/lib/455_filter
usr/lib/473_filter
usr/lib/475_filter
usr/lib/5310
usr/lib/ATT_s_filter
usr/lib/HP_filter
usr/lib/accept
usr/lib/acct
usr/lib/acct/acctcms
usr/lib/acct/acctcon1
usr/lib/acct/acctcon2
usr/lib/acct/acctdisk
usr/lib/acct/acctdusg
usr/lib/acct/acctmerg
usr/lib/acct/accton
usr/lib/acct/acctprc1
usr/lib/acct/acctprc2
usr/lib/acct/acctwtmp
usr/lib/acct/chargefee
usr/lib/acct/ckpacct
usr/lib/acct/diskusg
usr/lib/acct/dodisk
usr/lib/acct/fwtmp
usr/lib/acct/holidays
usr/lib/acct/lastlogin
usr/lib/acct/monacct
usr/lib/acct/nulladm
usr/lib/acct/prctmp
usr/lib/acct/prdaily
usr/lib/acct/prtacct
usr/lib/acct/ptecms.awk
usr/lib/acct/ptelus.awk
usr/lib/acct/remove
usr/lib/acct/runacct
usr/lib/acct/shutacct
usr/lib/acct/startup
usr/lib/acct/turnacct
usr/lib/acct/wtmpfix
usr/lib/calprog
usr/lib/cron
usr/lib/cron/.proto
usr/lib/cron/at.allow
usr/lib/cron/at.deny
usr/lib/cron/cron.allow
usr/lib/cron/cron.deny
usr/lib/cron/logchecker
usr/lib/cron/queuedefs
usr/lib/custom
usr/lib/custom/help
usr/lib/diffh
usr/lib/getoptcvt
usr/lib/hp2631a
usr/lib/layersys
usr/lib/layersys/lsys.8;7;3
usr/lib/layersys/lsys.8;7;5
usr/lib/layersys/lsys.8;8;6
usr/lib/layersys/relogin
usr/lib/layersys/set_enc.j
usr/lib/layersys/wtinit
usr/lib/lib.b
usr/lib/lib300.a
usr/lib/lib300s.a
usr/lib/lib4014.a
usr/lib/lib450.a
usr/lib/libgen.a
usr/lib/libp
usr/lib/libplot.a
usr/lib/libsec.a
usr/lib/libvt0.a
usr/lib/libwindows.a
usr/lib/lpfilter
usr/lib/lpforms
usr/lib/mailx
usr/lib/mailx/mailx.help
usr/lib/mailx/mailx.help.~
usr/lib/mailx/rmmail
```

usr/lib/more.help
usr/lib/pprx
usr/lib/prx

Base System Package, Diskette 7 of 7

usr/lib/lpadmin
usr/lib/lpmove
usr/lib/lpsched
usr/lib/lpshut
usr/lib/lpusers
usr/lib/mv_dir
usr/lib/reject
usr/lib/sa
usr/lib/sa/sa1
usr/lib/sa/sa2
usr/lib/sa/sadc
usr/lib/t300
usr/lib/t300s
usr/lib/t4014
usr/lib/t450
usr/lib/terminfo
usr/lib/terminfo/1
usr/lib/terminfo/2
usr/lib/terminfo/3
usr/lib/terminfo/4
usr/lib/terminfo/4/40-132-6
usr/lib/terminfo/4/40-132-8
usr/lib/terminfo/4/40-80-6
usr/lib/terminfo/4/40-80-8
usr/lib/terminfo/4/435
usr/lib/terminfo/4/43ro
usr/lib/terminfo/4/442
usr/lib/terminfo/4/444
usr/lib/terminfo/4/446
usr/lib/terminfo/4/447
usr/lib/terminfo/4/455
usr/lib/terminfo/4/457
usr/lib/terminfo/4/458
usr/lib/terminfo/4/470
usr/lib/terminfo/4/471
usr/lib/terminfo/4/473
usr/lib/terminfo/4/474
usr/lib/terminfo/4/475
usr/lib/terminfo/4/476
usr/lib/terminfo/4/477
usr/lib/terminfo/4/477-455
usr/lib/terminfo/4/477-470
usr/lib/terminfo/4/477ibmc
usr/lib/terminfo/4/477ibmg
usr/lib/terminfo/4/477qume
usr/lib/terminfo/4/478
usr/lib/terminfo/4/479
usr/lib/terminfo/4/495hp
usr/lib/terminfo/4/495ibm
usr/lib/terminfo/4/495qume
usr/lib/terminfo/5
usr/lib/terminfo/5/5310
usr/lib/terminfo/5/5320
usr/lib/terminfo/6
usr/lib/terminfo/7
usr/lib/terminfo/7/7475
usr/lib/terminfo/8
usr/lib/terminfo/9
usr/lib/terminfo/A
usr/lib/terminfo/A/AT386
usr/lib/terminfo/A/AT386-M
usr/lib/terminfo/A/AT386-UL
usr/lib/terminfo/B
usr/lib/terminfo/M
usr/lib/terminfo/P
usr/lib/terminfo/a
usr/lib/terminfo/b
usr/lib/terminfo/c
usr/lib/terminfo/d
usr/lib/terminfo/e
usr/lib/terminfo/f
usr/lib/terminfo/g
usr/lib/terminfo/h
usr/lib/terminfo/h/hplaserjet

usr/lib/terminfo/i
usr/lib/terminfo/i/ibmgraphics
usr/lib/terminfo/i/ibmproprinter
usr/lib/terminfo/j
usr/lib/terminfo/k
usr/lib/terminfo/l
usr/lib/terminfo/m
usr/lib/terminfo/n
usr/lib/terminfo/o
usr/lib/terminfo/p
usr/lib/terminfo/q
usr/lib/terminfo/r
usr/lib/terminfo/s
usr/lib/terminfo/s/sprint11
usr/lib/terminfo/t
usr/lib/terminfo/u
usr/lib/terminfo/u/unknown
usr/lib/terminfo/v
usr/lib/terminfo/w
usr/lib/terminfo/x
usr/lib/terminfo/y
usr/lib/terminfo/z
usr/lib/unittab
usr/lib/uucp
usr/lib/uucp/Devconfig
usr/lib/uucp/Devices
usr/lib/uucp/Dialcodes
usr/lib/uucp/Dialers
usr/lib/uucp/Maxuuscheds
usr/lib/uucp/Maxuuxqts
usr/lib/uucp/Permissions
usr/lib/uucp/Poll
usr/lib/uucp/SetUp
usr/lib/uucp/Sysfiles
usr/lib/uucp/Systems
usr/lib/uucp/Uutry
usr/lib/uucp/nttysrv
usr/lib/uucp/remote.unknown
usr/lib/uucp/uucheck
usr/lib/uucp/uucico
usr/lib/uucp/uucleanup
usr/lib/uucp/uudemon.admin
usr/lib/uucp/uudemon.cleanu
usr/lib/uucp/uudemon.hour
usr/lib/uucp/uudemon.poll
usr/lib/uucp/uugetty
usr/lib/uucp/uusched
usr/lib/uucp/uuxqt
usr/lib/vplot
usr/mail
usr/mail/:saved
usr/news
usr/pub
usr/pub/ascii
usr/spool
usr/spool/cron
usr/spool/cron/atjobs
usr/spool/cron/crontabs
usr/spool/cron/crontabs/adm
usr/spool/cron/crontabs/root
usr/spool/cron/crontabs/sys
usr/spool/locks
usr/spool/lp
usr/spool/lp/admins
usr/spool/lp/admins/lp
usr/spool/lp/admins/lp/classes
usr/spool/lp/admins/lp/forms
usr/spool/lp/admins/lp/interfaces
usr/spool/lp/admins/lp/logs
usr/spool/lp/admins/lp/printers
usr/spool/lp/admins/lp/pwheels
usr/spool/lp/bin
usr/spool/lp/bin/alert.proto
usr/spool/lp/bin/drain.output
usr/spool/lp/bin/lp.cat
usr/spool/lp/bin/lp.page
usr/spool/lp/bin/lp.set
usr/spool/lp/bin/lp.tell
usr/spool/lp/bin/lpsched.jr
usr/spool/lp/bin/slow.filter

usr/spool/lp/fifos
usr/spool/lp/fifos/private
usr/spool/lp/fifos/public
usr/spool/lp/logs
usr/spool/lp/model
usr/spool/lp/model/1640
usr/spool/lp/model/5310
usr/spool/lp/model/dqp10
usr/spool/lp/model/dumb
usr/spool/lp/model/f450
usr/spool/lp/model/hp
usr/spool/lp/model/lqp40
usr/spool/lp/model/ph.daps
usr/spool/lp/model/pprx
usr/spool/lp/model/prx
usr/spool/lp/model/standard
usr/spool/lp/requests
usr/spool/lp/system
usr/spool/lp/temp
usr/spool/lp/temp/435_table
usr/spool/lp/temp/455_table
usr/spool/lp/temp/473_table
usr/spool/lp/temp/475_table
usr/spool/lp/temp/HP_table
usr/spool/uucp
usr/spool/uucp/.Admin
usr/spool/uucp/.Corrupt
usr/spool/uucp/.Log
usr/spool/uucp/.Log/uucico
usr/spool/uucp/.Log/uucp
usr/spool/uucp/.Log/uux
usr/spool/uucp/.Log/uuxqt
usr/spool/uucp/.Old
etc/.installdate
etc/.installstart
etc/emulator.rel1
usr/spool/uucp/.Sequence
usr/spool/uucp/.Status
usr/spool/uucp/.Workspace
usr/spool/uucp/.Xqtdir
usr/spool/uucppublic
usr/tmp

Editing Package, Diskette 1 of 1

Size
spellprog
spellin
hashmake
hashcheck
exrecover
expreserve
diff3prog
vi
unpack
uniq
split
sdiff
pack
nl
newform
fgrep
egrep
deroff
ctags
csplit
comm
col
bfs
bdiff
sum
od
spellhist
tr
pg
paste
join
cut
touch
tail
exstrings
compress
hstop
hlistb
hlista
spell
dircmp
diff3
Install
Name
Remove
Files

Network Support Utility Package, Diskette 1 of 1

Size
libnsl_s.a
clone/Master
clone/System
clone/Driver.o
log/Master
log/System
log/Node
log/Space.c
log/Driver.o
timod/Master
timod/System
timod/Space.c
timod/Driver.o
tirdwr/Master
tirdwr/System
tirdwr/Space.c
tirdwr/Driver.o
ldterm/Master
ldterm/System
ldterm/Space.c
ldterm/Driver.o
ptem/Master
ptem/System
ptem/Space.c
ptem/Driver.o
ptm/Master
ptm/System
ptm/Space.c
ptm/Node
ptm/Driver.o
pts/Master
pts/System
pts/Node
pts/Driver.o
libpt.a
pt_chmod
listen
listen.h
libnls.a
nlsadmin
strace
strerr
strclean
Files
Name
Install
Remove
ldterm.h
ptem.h
ptms.h
lihdr.h
log.h
strlog.h

Remote File Sharing Package, Diskette 1 of 1

Size
du/System
du/Space.c
du/Driver.o
dufst/Master
dufst/System
dufst/Mfsys
dufst/Sfsys
dufst/Driver.o
sp/Master
sp/System
sp/Node
sp/Space.c
sp/Driver.o
adv
dname
fumount
fusage
idload
nserve
nsquery
n_rmount
n_rumount
n_rmnttry
rfadmin
rfpasswd
rfsetup
rfstart
rfstop
rfuadmin
rfudaemon
rmntstat
rmount
rmountall
rumountall
unadv
fumounts
rfs
rumounts
adv.init
Files
Name
Install
Remove

Remote Terminal Package, Diskette 1 of 1

install
install/INSTALL
install/Rlist
install/UNINSTALL
install/adds.ti
install/annarbor.ti
install/ansi.ti
install/att.ti
install/beehive.ti
install/cdc.ti
install/colorscan.ti
install/contel.ti
install/datamedia.ti
install/dec.ti
install/diablo.ti
install/fortune.ti
install/general.ti
install/hardcopy.ti
install/hazeltine.ti
install/hds.ti
install/heath.ti
install/homebrew.ti
install/hp.ti
install/lsi.ti
install/microterm.ti
install/misc.ti
install/pc.ti
install/perkinelmer.ti
install/print.ti
install/setup
install/special.ti
install/sperry.ti
install/tektronix.ti
install/teleray.ti
install/televideo.ti
install/ti.ti
install/tymshare.ti
install/visual.ti
new
new/usr
new/usr/lib
new/usr/lib/tabset
new/usr/lib/tabset/3101
new/usr/lib/tabset/beehive
new/usr/lib/tabset/std
new/usr/lib/tabset/teleray
new/usr/lib/tabset/vt100
new/usr/lib/tabset/xerox1720
new/usr/lib/terminfo
new/usr/options
new/usr/options/terminf.name

Security Administration Package, Diskette 1 of 1

Size
libcrypt_d.a
makekey
crypt
Remove
Name
Files
Install

2 Kilobyte File System Utility Package, Diskette 1 of 1

Size
Driver.o
Master
Mfsys
Sfsys
System
Space.c
Stubs.c
Name
Files
Install
Remove
fsba

XENIX File System Utility Package, Diskette 1 of 1

Size
Driver.o
Master
Mfsys
Sfsys
System
Stubs.c
Name
Files
Install
Remove
xfsck
xxfblk.h
xxfilsys.h

Extended Terminal Interface Utility Package, Diskette 1 of 1

Size
Files
Install
Name
Remove
usr/lib/tamhelp
usr/lib/libxtermcap.a
usr/lib/libxcurses.a
usr/lib/libcurses.a
usr/lib/libtam.a
usr/lib/libpanel.a
usr/lib/libmenu.a
usr/lib/libform.a
usr/include/xcurses.h
usr/include/eti.h
usr/include/tam/sys/signal.h
usr/include/tam/sys/mouse.h
usr/include/tam/sys/iohw.h
usr/include/tam/sys/window.h
usr/include/tam/temp.h
usr/include/tam/sys/font.h
usr/include/tam/tamwin.h
usr/include/tam/pbf.h
usr/include/tam/form.h
usr/include/tam/tam.h
usr/include/tam/wind.h
usr/include/tam/print.h
usr/include/tam/message.h
usr/include/tam/chartam.h
usr/include/tam/subcurses.h
usr/include/tam/kcodes.h
usr/include/tam/menu.h
usr/include/tam/track.h
usr/lib/llib-ltam.ln
usr/lib/llib-ltam
usr/include/panel.h
usr/lib/llib-lpanel.ln
usr/lib/llib-lpanel
usr/include/form.h
usr/lib/llib-lform.ln
usr/lib/llib-lform
usr/include/menu.h
usr/lib/llib-lmenu.ln
usr/lib/llib-lmenu
usr/lib/llib-lcurses.ln
usr/lib/llib-lcurses
usr/include/unctrl.h
usr/include/term.h
usr/include/curses.h
usr/include/windows.h

Appendix B: Documentation Updates

Documentation Updates

The following change pages reflect last minute changes to the AT&T UNIX System V/386 Release 3.2 documentation. These change pages should be inserted into the *Operations/System Administration Guide* per the following instruction.

AT&T UNIX SYSTEM V/386 RELEASE 3.2
OPERATIONS/SYSTEM ADMINISTRATION GUIDE
UPDATES

This update involves the following action:

1. **ACTION:** Replace page 9-11 through page 9-14 of Chapter 9 with the new pages.

Setting Up RFS

In most cases, you will not need the set of tasks described in this section because the basic RFS configuration and reconfiguration can be handled using the commands described earlier in this chapter. These tasks are for those who want to go deeper into the workings of RFS or are having problems with particular components.

These tasks are run from the shell. They should be run initially in the order described.

Once these tasks are completed, go to the "Starting/Stopping RFS" section for information on starting RFS.

Prerequisites

Before you begin setting up RFS, the following must be installed and running: UNIX System V Release 3.1 (or later) software, Remote File Sharing Utilities, Networking Support Utilities, and transport provider software. (See the *Remote File Sharing Release Notes* and the transport provider manuals that accompany the product for installation instructions.)

You must also log in as *root*.

Set Node Name

CAUTION ▽ Changing the node name of your computer requires careful coordination with all machines that communicate with yours using Remote File Sharing or other communications packages that rely on node name.

Check to see if your computer's node name is set to the name you want (**uname -n**). If it's not, set it by typing

```
uname -S nodename
```

A node name that is valid for RFS can consist of up to eight characters of letters (uppercase or lowercase), digits, hyphens (-), and underscores (_). Some networks, such as the STARLAN network, require that every node name in the network be different. RFS, however, only requires that every node name in a domain be different.

Set Up Network Listener

If you have installed the Networking Support Utilities, the AT&T implementation of the STARLAN network, and RFS in the order described in Chapter 2, "Software Installation," you can skip this task. The listener will already be installed and set up to run automatically, and RFS will be listed as an available service.

If you are using another transport provider or suspect that your STARLAN network listener is set up improperly, this task will show how to manually set up the listener. In the following example, the STARLAN network is used. To set up the listener for other networks compatible with the AT&T Transport Interface, you should replace **starlan** with the name of the network (network specification) you are installing. (For more details, see the *nlsadmin*(1M) manual page in the *User's/System Administrator's Reference Manual.*)

To determine if the listener is properly installed and set up for use by RFS, type the following:

```
nlsadmin -v starlan
```

If service code 105 is listed, then the listener is configured to be used for RFS.

Run the following commands if the listener is not properly set up. If you run any of these commands and they have already been run, you will receive a message telling you so. This will not harm your listener configuration. Type

```
nlsadmin -i starlan
```

to initialize the files needed for the listener process for the network specified, in this case **starlan**.

Next, type

```
nlsadmin -a 105 -c /usr/net/servers/rfs/rfsetup -y  "rfsetup" starlan
```

to add the RFS service (**rfsetup**) to the list of services available to the **starlan** listener.

Use the following command line to report the status of the **starlan** listener process installed on this machine (ACTIVE or INACTIVE):

```
nlsadmin -x
```

Next, type

```
nlsadmin -l "nodename.serve" -t "nodename" starlan
```

to register the network addresses of your machine. The listener will listen for requests for these addresses on the network. Only the **-l** address is required by RFS. The **-t** address is used only for terminal services and may not be needed on all networks.

To start the listener, type

```
nlsadmin -S starlan
```

Normally, it will be started automatically when your machine enters multi-user mode (**init 2**).

Set the Domain Name

Set the domain name by typing

dname -D *domain*

where *domain* is replaced by the domain of which your machine will be a member. The domain name must:

- contain no more than 14 characters
- consist of any combination of letters (uppercase or lowercase), digits, hyphens, and underscores
- be different from the name of any other domain used on the network if there is more than one domain on your network

You can check the current domain name by typing:

```
dname
```

Set the Transport Provider

To identify the network, you must tell RFS which network (transport provider) it should use. (In our example, this is **starlan** for the STARLAN network.)

```
     dname -N starlan
```

This command indicates the device, relative to the */dev* directory, that is used for the transport provider.

Create rfmaster File

The *rfmaster* file should only be created manually on the primary. If your machine is not the primary, you should skip this task; the *rfmaster* file for your domain will automatically be placed on your machine the first time you start RFS (**rfstart -p** *primary_addr*).

If you are on the primary, you can create an *rfmaster* file in the */usr/nserve* directory using any standard file editor. The contents of this file will define the following:

- the primary name server for your domain
- secondary name servers for your domain
- network addresses for each of these machines

(See the section on "Multiple Domain Name Service" in this chapter for a description of other information you may want to put into the *rfmaster* file.)

Here is an example of an *rfmaster* file for a domain called **peanuts,** whose primary and secondary name servers' node names are **charlie, linus,** and **lucy**. Adding each machine's domain name (**peanuts**) to its node name, separated by a period, forms its full RFS machine name. Each line of the example translates as follows:

- For domain **peanuts,** the primary is **peanuts.charlie.**
- For domain **peanuts,** a secondary is **peanuts.linus.**
- For domain **peanuts,** another secondary is **peanuts.lucy.**

Index

AT&T UNIX SYSTEM V/386
RELEASE 3.2
SOFTWARE DEVELOPMENT SET
RELEASE NOTES

Introduction

Overview

These *Release Notes* contain information about the Software Development Set (SDS) package. The SDS package is useful to programmers who:

- Want to develop C language programs
- Do extensive programming in the C language
- Want to enhance the efficiency of a C program written in a UNIX system environment
- Need tools to do advanced programming and symbolic debugging
- Want to work with shared libraries
- Work in an environment where it is necessary to track and maintain versions of files and programs
- Want to optimize and streamline development of interactive, character-oriented, C application programs.

The Software Development Set runs on a computer running AT&T 386 UNIX System V/386 Release 3.2.

The SDS software package is made up of two parts as follows:

- C Software Development Set (CSDS)
- Extended Terminal Interface (ETI).

Conventions Used in This Document

In these *Release Notes,* certain typesetting conventions are followed when command names, command line format, files, and directory names are described. There are also conventions for displays of terminal input and output.

- You must type words that are in **bold** font exactly as they appear. Also, commands, filenames, and directory names appear in **bold.**
- Words in *italics* are variables; you substitute the appropriate values. These values may be filenames or they may be data values.
- CRT or terminal output and examples of source code are presented in `constant-width` font.
- In output and source code examples, a backslash (\) at the end of a line indicates that the line wraps around without a break.
- A command name followed by a number, for example, **prof**(1), refers you to that command's manual page, where the number refers to the section of the manual. These manual pages appear in the *AT&T UNIX System V/386 Release 3.2 Programmer's Reference Manual* unless otherwise noted.

Contents of the Release

The Software Development Set (SDS) comes in one set of five diskettes (four diskettes for CSDS and one diskette for ETI), the contents of which are displayed in the following table.

Table 1: SDS Utilities

Directory	Files		
/bin	ar	cprs	lorder
	as	dis	make
	cc	dump	mkshlib
(CSDS)	chkshlib	gencc	nm
	conv	ld	size
	convert	list	strip
/etc	install		
(CSDS)			
/lib	basicblk	crtn.o	libx.a
	cm4defs	libc.a	mcrt0.o
	comp	libc_s.a	mcrt1.o
(CSDS)	cpp	libld.a	optim
	crt0.o	libm.a	pcrt1.o
	crt1.o	libPW.a	pcrt0.o
/usr/add-on/include	chartam.h	pbf.h	temp.h
	form.h	print.h	track.h
(ETI)	kcodes.h	subcurses.h	wind.h
	menu.h	tam.h	
	message.h	tamwin.h	
/usr/add-on/include/sys	font.h	mouse.h	window.h
(ETI)	iohw.h	signal.h	

Table 1: SDS Utilities (Continued)

Directory	Files		
/usr/bin	admin	delta	sact
	cb	get	sccsdiff
	cdc	lex	sdb
	cflow	lint	tsort
	comb	lprof	unget
(CSDS)	cscope	m4	val
	ctc	prof	vc
	ctcr	prs	what
	ctrace	regcmp	yacc
	cxref	rmdel	
/usr/bin (ETI)	captoinfo	infocmp	tic
	tput		

Table 1: SDS Utilities (Continued)

Directory	Files		
/usr/include	a.out.h	malloc.h	sgtty.h
	aouthdr.h	math.h	signal.h
	ar.h	memory.h	stand.h
	assert.h	mnttab.h	stdio.h
	core.h	mon.h	storclass.h
	ctype.h	nan.h	string.h
	dial.h	nlist.h	stropts.h
	dirent.h	nsaddr.h	strselect.h
	errno.h	nserve.h	syms.h
(CSDS)	fatal.h	poll.h	sys.s
	fcntl.h	prof.h	termio.h
	filehdr.h	pwd.h	time.h
	ftw.h	regexp.h	tp_defs.h
	grp.h	reloc.h	ttysrv.h
	ieeefp.h	rje.h	unistd.h
	ldfcn.h	scnhdr.h	ustat.h
	limits.h	sd.h	utmp.h
	linenum.h	search.h	values.h
	macros.h	setjmp.h	varargs.h
/usr/include	curses.h	menu.h	tiuser.h
(ETI)	eti.h	panel.h	unctrl.h
	form.h	term.h	

Table 1: SDS Utilities (Continued)

Directory	Files		
/usr/lib	basicblk	liby.a	llib-port
	dag	lint1	llib-port.ln
	flip	lint2	lpfx
	libcrypt.a	llib-lc	nmf
(CSDS)	libg.a	llib-lc.ln	xcpp
	libl.a	llib-lm	xpass
	libmalloc.a	llib-lm.ln	yaccpar
	libprof.a	llib-lmalloc.1	
/usr/lib	libcrypt.a	llib-lcurses	llib-lmenu.ln
	libform.a	llib-lcurses.a	llib-lpanel
	libmenu.a	llib-lcurses.ln	llib-lpanel.ln
(ETI)	libpanel.a	llib-lform	llib-ltam
	libtam.a	llib-lform.ln	llib-ltam.ln
	libtermcap.a	llib-lmenu	tamhelp
	libtermlib.a		
/usr/lib/ctrace (CSDS)	runtime.c		
/usr/lib/help	ad	co	prs
	bd	de	rc
(CSDS)	cb	default	un
	cm	ge	ut
	cmds	he	vc
/usr/lib/help/lib	help	help2	

Table 1: SDS Utilities (Continued)

Directory	Files		
/usr/lib/lex (CSDS)	ncform	nrform	
/usr/lib/libp (CSDS)	libc.a libx.a	libm.a	libmalloc.a
/usr/lib/tabset (ETI)	3101 beehive	std teleray	vt100 xerox1720
/usr/options (CSDS)	csoftw.name		
/usr/options (ETI)	graphi.name	terminf.name	
/usr/src/lib/eti/demo (ETI)	form0.c form1.c	form2.c menu0.c	menu1.c

Software Overview

The SDS package has two major parts: the C software development set (CSDS) and the extended terminal interface (ETI). CSDS can be used for developing, debugging, and improving the efficiency of C language programs. ETI is a set of libraries that promotes fast development of screen management applications. These two parts of the SDS package are discussed in the following subsections.

CSDS

CSDS is a collection of tools and utilities that aid you in:

- Developing C language programs
- Advanced programming, symbolic debugging, and improving C language program efficiency.
- Keeping a history of source code files by recording changes made to these files along with comments on each version.

The C Programming Language Development Tools

The main C programming language development tool is the compiler, and is called by the command **cc**. The other programming development tools discussed in this section are the C preprocessor, optimizer, assembler, link editor, tools for manipulating object files, and libraries.

C Compiler

The C compiler supports the C language as specified in *The C Programming Language*. The significant extensions to the language include the following:

- Arbitrary length names for variables and function names
- Structure assignments and arguments
- Functions returning structure values
- Enumerated data types
- Multiple external variable declarations

- Assembly language escapes from C
- Insertion of arbitrary strings into object modules (useful for version control)
- Floating point support in conformance with the *Standard for Binary Floating-Point Arithmetic,* ANSI/IEEE Std 754-1985
- Data type **void**
- Additional preprocessor directives.

cc Command

The **cc** command, the major command of CSDS, calls the C compiler. The **cc** command also controls the other phases of compilation, and, unless programmers use options to specify otherwise, **cc** automatically calls the C preprocessor, assembler, and link editor phases. The command options have many uses, such as suppressing the assembler or link editor or invoking the optimizer. The **cc** command also passes some options to these other programs.

The **cc** command accepts files containing C source code as input. The result of the compilation process is an executable module named **a.out** that reflects the contents of the source files and any referenced library routines. The **cc** command also accepts source files that contain assembly language code as input and passes these files directly to the assembler.

C Preprocessor

The C preprocessor [**cpp**(1)] is automatically called whenever the **cc** command is given C source input. The preprocessor performs file inclusion and macro substitution.

Optimizer

The optimizer, an optional component in the compilation process, improves the efficiency of compiler-generated assembly language code. The optimizer reduces the space requirements and speeds the execution time of the resulting object code.

Assembler and Assembly Language

The assembler [**as**(1)] is available for developing applications that require close interaction with hardware, such as those needed to handle input/output devices and interrupts. The assembler converts assembly language code into a relocatable object module composed of machine code and symbolic information. This component provides assembly language programmers access to predefined macros using the UNIX operating system **m4** macro processor.

Link Editor

The link editor [**ld**(1)] combines relocatable object modules and libraries to produce either an absolute, executable load module or a relocatable object file for use in further link edits. Executable load modules are in the Common Object File Format (COFF). The link editor performs relocation, resolves external references, and incorporates symbolic debugging information into its output file. It searches libraries to resolve all external references and only loads library routines that define an unresolved external reference.

Tools for Manipulating Object Files

CSDS provides a variety of commands used to read and manipulate object files. Here is a list of some utilities with brief descriptions of their use:

ar Groups files into a single, portable archive file commonly used as a library

cprs Compresses object files by removing duplicate structure and union symbolic information

dis Disassembles object files to allow assembly level debugging

dump Prints selected parts of the named object files

lorder Generates an ordered listing of object files for efficient library link editing

nm Prints the symbolic information in an object file

size Reports the number of bytes of text, initialized data, and uninitialized data (and their sum) included in an object module

strip Reduces file storage overhead by removing symbolic information from an object file.

Libraries

CSDS comes with libraries for object files, access to system calls, input/output, string manipulation, mathematical functions, and memory allocation.

Advanced Programming Tools and Utilities

The CSDS package contains an extensive set of tools useful for advanced application programming, debugging, improving the efficiency of your programs, and aiding you in keeping track of the different versions of your programs.

Programming and Debugging Utilities

The programming utilities are specialized utilities helpful in the design and development of application programs and systems. The following list gives a short description of the major programming utilities.

cxref is a C cross-reference listing generator.

ctrace is a statement-by-statement execution trace facility.

cflow produces a graph of program dependencies.

lint detects faulty and non-portable code.

cb is a C code beautifier

regcmp compiles regular expressions

mkshlib(1) makes a shared library. Shared libraries is a feature of UNIX System V Release 3.0, and beyond, that allow several **a.out** files to simultaneously use the same object code.

chkshlib(1) checks a shared library.

sdb(1) a symbolic debugger used to examine C language executable files and core files and to provide a controlled environment for their execution. When testing C language programs symbolically, breakpoints can be set at executable lines of the source code. These breakpoints force the program to pause at the specified point so that an inspection can be made of the current state of the program.

make(1) a tool that helps you build and maintain up-to-date versions of programs. **make** simplifies the job of keeping track of which files depend on other files, recently modified files, files that need recompiling after changes, and the sequence of operations needed to make a new version of a program.

lex(1) a tool that generates programs to be used in simple lexical analysis of text. The **lex** tool reads a file containing specifications of strings to be matched and associated C code. Whenever the lexical analyzer produced by **lex** matches a specified string in its input, it executes the associated C code.

yacc(1) a tool (Yet Another Compiler-Compiler) that accepts both an LALR(1) grammar specification and associated C code fragments that represent actions to be taken when a found grammar rule is reduced, and then produces a parser.

All of these utilities are described in the *AT&T UNIX System V/386 Release 3.2 Programmer's Guide* and the *UNIX System V/386 Release 3.2 Programmer's Reference Manual.*

Productivity Utilities

The CSDS package has three utilities that can help an experienced programmer enhance the efficiency of a C program written in a UNIX operating system environment. These utilities are a browser called **cscope** and two profilers, **lprof** and **prof**.

A browser is an interactive program that helps you examine source files by searching for functions, function calls, macros, and variables that you specify. When it finds them, the browser puts you into an editor at the specified location. Thus, instead of thumbing through a stack of printouts to learn code or locate a bug, you can specify a function or text string and let the browser find it. Then you have the option of examining that portion of code or editing it. Whether you want to familiarize yourself with a program or edit a source file, a browser can help you accomplish your task without your reading the code line by line.

The browser in CSDS, designed for use with C code, is called **cscope**. Programmers responsible for writing programs or maintaining existing programs will be able to edit their source code more efficiently with **cscope**. It is especially helpful for a programmer working on someone else's code.

A profiler is a tool that performs dynamic analysis or analysis of a program at run time; it accomplishes this in two phases. First, the profiler collects data about the code while a program is being executed. Then it displays this data in a readily accessible format. The profiler **lprof** provides line-by-line frequency profiling, reporting how many times each line of source code is executed. To obtain a more representative sample of program performance, you can run a program profiled with **lprof** more than once and then merge the data from the multiple runs. This information can be useful in every stage of software development: designing, prototyping, coding, testing, debugging, and maintenance.

The profiler **lprof** can also be used to determine which lines of source code are executed and how much of the code is exercised. These types of output can be obtained by using the **-x** option and the **-s** option, respectively. These options are convenient for programmers who are interested only in execution coverage and who do not need the additional information that **lprof** normally provides. For example, if you are developing a test suite and want to find out how much code is actually tested by your test suite, run **lprof** with either the **-x** or **-s** option, depending on the level of detail you want.

Another CSDS profiler you may find useful is **prof**. The **prof** profiler reports the amount of time spent in various parts of a program during execution. The use of **prof** is not required for using **lprof**, but by using these profilers together you can increase the efficiency of **lprof**. The **prof** profiler allows you to identify the most time-consuming parts of a program. By running **lprof** on only those parts of code, you can avoid generating uninformative output while targeting sections of code that need pruning. It is therefore recommended that you use **prof** and **lprof** together.

To use these utilities, you must know how to use CSDS in the UNIX system environment. These utilities do not modify code for you; they enable you to find parts of code that deserve further work on your part. For programmers who have not compiled C code or used CSDS before, the basics are covered in the *AT&T UNIX System V/386 Release 3.2 Programmer's Guide.*

Source Code Control Utilities

A subset of the CSDS utilities, sometimes called the source code control system (SCCS), is specifically designed for source code control. These utilities can be used to record all enhancements and changes to files, along with comments on each version, thus maintaining a history of the changes made. The major SCCS functions include:

- Retrieving any recorded version of a file with comments
- Storing a new version of a file
- Comparing two versions of an SCCS file.

SCCS takes custody of a file, and, when changes are made, identifies and stores them in the file with the original source code and/or documentation. As other changes are made, they too are identified and retained in the file. Each separate set of changes is called a delta. History data can be stored with each version: why the changes were made, who made them, when they were made, etc.

Retrieval of the original or any set of changes is possible. Any version of the file as it develops can be reconstructed for inspection or additional modification.

Here is a list of SCCS commands.

get Retrieves versions of SCCS files.

unget Undoes the effect of a **get -e** prior to the file being delta'd.

delta Applies deltas (changes) to SCCS files and creates new versions.

admin Initializes SCCS files, manipulates their descriptive text, and controls delta creation rights.

prs Prints portions of an SCCS file in user-specified format.

sact Prints information about files that are currently out for edit.

help Gives explanations of error messages.

rmdel Removes a delta from an SCCS file. Allows removal of deltas created by mistake.

cdc Changes the commentary associated with a delta.

what Searches any UNIX operating system file(s) for all occurrences of a special pattern and prints out what follows that pattern. Useful in finding identifying information inserted by the **get** command.

sccsdiff Shows differences between any two versions of an SCCS file.

comb Combines consecutive deltas into one to reduce the size of an SCCS file.

val Validates an SCCS file.

vc Is a filter that may be used for version control.

For instructions on how to use SCCS and detailed descriptions of SCCS commands, see the "Source Code Control System" chapter in the *AT&T UNIX System V/386 Release 3.2 Programmer's Guide.*

Extended Terminal Interface

ETI is a set of libraries that promote fast development of screen management applications. The ETI libraries are a software tool that enable you to incorporate screen management and data entry capabilities into your programs. ETI contains the following libraries:

- *Curses/Terminfo Low Level Function Library*: This library consists of routines for writing character-oriented screen management applications independent of the terminal type. Basic routines are provided for writing to a screen, reading from a screen and building windows. Advanced features are used to change screen attributes, draw line graphics and work with more than one terminal. A major new feature is the incorporation of color. You can specify both the background color for each character and the color of the character itself.
- *High-Level Function Libraries*: The high level function libraries are built on top of curses. They consist of functions that create, manipulate, and display panels, forms, and menus.
  - *Panels*: A panel is a rectangular area containing a curses window that may be displayed in whole or in part on the terminal. Panels provide a depth relationship between curses windows. Panels which are logically below other panels are properly obscured.
  - *Forms*: A form is a multi-page display that contains a set of fields. These fields may be used for data entry, labels, or messages. You can customize the look and behavior of a form or field. The rich set of form commands includes the following: inter-field and intra-field navigation, field editing, data entry, and validation.
  - *Menus*: A menu is a display presenting a collection of items. The end-user can select one or more items and this information is available to the application. You can customize the look and behavior of a menu. Menu commands are provided for item navigation, menu scrolling, and item matching.
- *Terminal Access Method (TAM) Transition Library*: The TAM Transition Library enables character mode applications developed for the UNIX PC using TAM to run on other processor/terminal configurations. The library maps TAM calls to curses routines.

Software Features

The CSDS package supports character classification and conversion and international date and time formats. The **ctype**(3C), **ctime**(3C) and **cftime**(4) routines have been modified as described in the following subsections. Also, the dynamic tables of the CSDS components **comp** (compiler) and **as** (assembler) are described. Other CSDS features discussed in the following subsections include referencing a shared library from within a shared library, the #hide and #export directives, checking shared library versions with chkshlib(1), and a proposed C language standard.

ctype(3C)

The classification of characters (what constitutes alphabetic, printable, uppercase or lowercase) varies from language to language. The **ctype**(3C) library routines that are used to classify character-coded integer values have been enhanced to recognize other code sets or classifications. Among these is the routine **setchrclass**(3C), which is a new routine used to initialize the character classification and conversion table. It is invoked at program startup and can be invoked directly from users' programs. This means the character set specific table can change dynamically.

ctime(3C)

The **ctime**(3C) routines allow the user to manipulate date and time formats. Several new library functions (**cftime, ascftime**, and an enhanced **tzset**) have been added to **ctime**(3C). These routines support the following features:

- The ability to specify fractional time zones
- The ability to specify start and end dates and times of alternate time zones
- The ability to specify time and date formats with new format field descriptors
- The ability to specify native language translations of month and weekday names.

cftime(4)

The **cftime**(4) manual page describes how to create language specific files. These files contain detailed information such as full and abbreviated month names, full and abbreviated weekday names, and default local time and date formats.

For more information on how to use these features, see **ctime**(3C), **ctype**(3C), **cftime**(4), and **environ**(5) in the *AT&T UNIX System V/386 Release 3.2 Programmer's Reference Manual.*

Dynamic Tables

Though the C language tends to encourage small functions and source files, some existing applications contained very large source files that failed to compile under previous issues of CSDS because of the fixed size of some tables in the compilation system. In this issue, the tables in the compiler and the assembler are allocated dynamically.

In the compiler, successful compilation is no longer constrained by the number of symbols, the number of cases in a switch, the number of arguments to a function, etc., except as limited by the amount of memory on your machine. Similarly, the assembler's constraint on the number of symbols has been removed.

Referencing a Shared Library from Within a Shared Library

At times you might need to allow one shared library to directly reference routines in another shared library. One way to do this is with imported symbols. Another way is to reference routines in one shared library from another shared library; use the keyword **noload**, with the **#objects** directive in the shared library specification file. When the **#objects noload** directive is used, the **mkshlib** command will search the libraries listed for unresolved references. You will want to use this feature only when you cannot import symbols explicitly.

The #hide and #export Directives

Two directives, **#hide** and **#export**, can be used in the library specification file to control the visibility of external symbols.

Checking Shared Library Versions with chkshlib(1)

The **chkshlib**(1) command allows you to compare versions of shared libraries to see if they are compatible. This command accepts various combinations of executable files, target shared libraries, and host shared libraries as input and tells you if the library versions are compatible, or if the specified executable could have been built by or can run with the specified host or target shared library.

For more information about shared libraries, see the chapter on shared libraries in the *AT&T UNIX System V/386 Release 3.2 Programmer's Guide.* The *AT&T UNIX System V/386 Release 3.2 Programmer's Reference Manual* contains more information about **chkshlib**(1) and **mkshlib**(1).

Proposed Standard for C

As these *Release Notes* were published, no official standard for the C programming language existed. The language accepted by AT&T C compilers follows the definition given in *The C Programming Language* by B. Kernighan and D. Ritchie (Prentice-Hall, 1978). The CSDS package also supports the following extensions.

- Flexnames

  This extension allows variable and function name tokens to be distinct to at least the first 100 characters (rather than the first 8 characters).

- Structure assignments and return values

  This extension allows variables of the same structure type to be assigned to one another. The return value of functions can also be a structure.

- Enumeration types
- Multiple external variable declarations

  This extension makes it possible to have the declaration

  ```
  int i;
  ```

  in multiple source files. All these multiple references resolve to the same address at link edit time.

Currently the X3/J11 task force of the American National Standards Institute (ANSI) is defining a standard for the C language (Draft Proposed American National Standard for Information Systems — Programming Language C, October 1986). The standard proposed by ANSI will allow most current legal C programs to be compiled without any changes. Nevertheless, to ease the possible transition process to the standard, the AT&T C compiler included with CSDS warns about the use of some constructs that may not be legal in the future or may cause portability problems. The following are examples of such constructs.

- Declarations, such as,

  ```
  int  i;
   static int i;
  ```

  produce the warning message

  ```
  warning: i previously declared extern, becomes static.
  ```

- Structure definitions missing semicolons, such as

  ```
  struct x {
     int i
  }
  ```

  produce the warning message

  ```
  warning: syntax requires ; at end of struct/union decl
  ```

Installation Notes

The following text describes the space dependencies and version control as it relates to the installation of the SDS package. For complete installation procedures, see the *Operations/System Administration Guide.*

Space Dependencies

The SDS package is installed using the *installpkg*(1) command. The *installpkg*(1) command checks to determine that sufficient free space is available in the **root** and **/usr** file systems. You need approximately 7,900 blocks (512-byte blocks) of memory to install the SDS package.

Version Control

The C software development set portion of the SDS package uses a per file method of version control. If the file being installed already exists on the system and has a release number greater than the file belonging to the package being installed, the existing file will not be overwritten. Files without valid release information are assumed to be older than those belonging to the package being installed.

Verification

After installing SDS, verify the correct SDS version (4.1.5) by using cc -V.

Software Notes

This section offers some tips on using the SDS package and some software tips that enhance the usability of the package.

1. Functions of type *float* or *double* need to be declared in scope whether or not their return values are being used.
2. Elements of type *char* will be sign extended. For zero extension, *unsigned char* must be used.
3. If you are compiling your programs with the -g option enabled so that you can do debugging, it is advisable **NOT** to use the -O option as well. In some cases, the two options invoked jointly will produce multiply defined labels. In addition, you should not use -O when compiling -ql because this in turn turns on the -g flag.
4. The default response to the invalid operation, divide by zero, and overflow exceptions is to take a trap. This behavior may be altered by using the **fpsetmask(3)** function.
5. When an *Intel* 80287 co-processor is installed, use of denormalized floating point numbers results in a core dump. The problem is that the 80287 chip does not normalize a denormal number when it is loaded and produces an invalid operation exception when a denormal number is stored to memory. If such problems are encountered, one work-around is to enable the denormalized operand exception and provide a signal handler which normalizes a denormal number. This signal handler must also recognize any other enabled traps (signals).
6. Without an *Intel* 80287 or 80387 coprocessor installed, the floating point emulator incorrectly returns 0 rather than NaN for any operation on NaN.
7. The IEEE 754 standard for floating point (IEEE Standard for binary Floating-Point Arithmetic, ANSI/IEEE Std. 754-1985) allows several different methods for detecting overflow. As a result, you should not rely on a particular implementation to signal overflow for a particular operation.
8. Floating point comparisons where one operand is an NaN always result in an invalid operation exception. This is because the *Intel* 80287 lacks an instruction to make this comparison without getting the exception.

9. **dis(1)** and **sdb(1)** do not recognize the *Intel* 80387 specific instructions.
10. **pipe(2)** — The documentation states that the maximum number of bytes in a pipe (PIPE_MAX) is defined to be 5120. The system sets PIPE_MAX to 10240.
11. **ioctl(2)** — The **V_ADDBAD** command (notifies the device drivers of bad sectors) in **ioctl(2)** updates only the table currently in memory and does not update the table on the hard disk. Therefore, all the changes made using **ioctl(2)** with **V_ADDBAD** will be lost when the system is rebooted. Also, if an assigned alternate sector goes bad, there is no way to recover.
12. **ioctl(2)** — The **V_GETPARMS** command in **ioctl(2)** returns the incorrect number of sectors for a 360KB device. The number of sectors reported is 1440; however, the correct value is 720.
13. The Graphics Programming Utilities (GPU) has been renamed extended terminal interface (ETI).
14. A new function, **gethz()**, is added to **libc** that gets the HZ value from the environment.
15. The **-s** option is added to the symbolic debugger (**sdb**). The system will not catch the trap specified by the **-s** option. For example, if you specified **sdb -s2**, it will not trap on error number 2.
16. The **-Zp[1¦2¦4]** option is added to **cc**. This option packs structure members in memory. Normally, structure members are aligned as follows: items of type **char** are byte-aligned, items of type **short** are aligned on 2-byte boundaries, and all other types of structure members are word-aligned. Specifying an option to **-Zp** will force alignment on the given byte boundary. If no option is used with **-Zp**, structure members will be packed on 1-byte boundaries. The alignment may be altered with the **#pragma pack** preprocessor directive.
17. The **-x** option is added to **convert**. This is required to convert a *Xenix* archive. Using this option will convert the general archive structure but leave archive members unmodified.

Compatibility

This section describes the changes made in this issue of the SDS package that may have an effect on the compatibility of your programs.

The Compiler and cc

The following compatibility notes concern changes made to the CSDS **cc** command or the compiler, **comp**, in this issue of the SDS. These notes apply only if you are porting C programs compiled on an AT&T compilation system (release number less than 4.1) for a different machine.

- The **-B** and **-t** options have been removed from the **cc** command. Previous releases printed a warning message that these options would disappear.
- The handling of aggregate initialization has been changed to conform to the definition given by Kernighan and Ritchie. Initialization where all braces are specified or where only the outermost braces are specified continues to work as before.
- **cc** and **comp** can no longer take the address of a label.

  The following illegal C code will no longer compile:

  ```
  f(){
       int i;
  lab:
       i = (int) &lab;
  }
  ```

- Bad structure code, such as the following, is disallowed:

  taking the address of the return value of
  a function which returns a structure:

  ```
    pst = &(stcall());
  ```

  using a function return value as an L-value:

  ```
    stcall() = *pst;
  ```

  taking the address of a structure assignment:

  ```
    pst = &(st1=st2);
  ```

cpp

The following change was made to **cpp** in this issue of the SDS.

- A missing or invalid macro name in `ifdef, ifndef, undef,` or `define` is now a fatal error.

  For example:

  ```
  #ifdef 202
  #undef
  #undef 1abc
  ```

Changes in C Library Functions

The following list describes changes made to functions in the C library in this issue of the SDS.

ctime(3C) An **a.out** compiled with previous versions of the **ctime** functions when used with some new legal TZ values will give unexpected results.

ctime(3C) **ctime** now defaults to GMT if TZ is not set.

In previous releases it defaulted to EST.

fgets(3S) A call to **fgets** on a write-only file returns NULL. In earlier releases, **fgets** always returned the address of the buffer passed to it.

fread(3S), **fwrite**(3S)

The **fread** and **fwrite** functions return zero when **size** is zero or huge.

In an earlier release, these two functions always returned **nitems. size** and **count** are multiplied to give the number of bytes to be transferred. If the result is larger than the remaining bytes of the file or is not representable within the precision of an integer, fewer items will be read than requested and the number of items actually read will be returned.

scanf(3S) Calls to **scanf** now return EOF on end-of-file. In an earlier release, **scanf** erroneously returned zero.

Environment Variables

The variables CFTIME, CHRCLASS, and LANGUAGE are environment variables in CSDS. Setting them may cause C library functions to change their behavior. Also, the TZ environment variable may be interpreted differently. The following table lists the library functions affected by these variables.

Function	Environment Variables
ctime	TZ
isalnum	CHRCLASS
isalpha	CHRCLASS
iscntrl	CHRCLASS
isdigit	CHRCLASS
isgraph	CHRCLASS
islower	CHRCLASS
isprint	CHRCLASS
ispunct	CHRCLASS
isupper	CHRCLASS
localtime	TZ
tolower	CHRCLASS
toupper	CHRCLASS

The mkshlib Command

Uninitialized external variables (common symbols) are illegal in a shared library. Previously, the use of common symbols was discouraged by both the documentation and a **mkshlib** warning message. This warning message is now a fatal error.

Future Directions

This section describes areas of the SDS product that are likely to change in future releases.

1. It is likely that some of the functions defined in **libPW** will be removed in a future release of this product. If you have any code that relies on **libPW**, AT&T recommends that you reimplement it using existing functions in the standard C library or that you retain copies of the **libPW** functions that you need.
2. The **list**(1) command will be removed in a future release of this product.
3. AT&T expects, in a future release of the SDS, to support the ANSI Standard for the C language once the standard is accepted. That standard introduces the keywords **const**, **signed**, and **volatile**. Programmers should therefore avoid using these words as identifiers in programs.
4. A major feature of the extended terminal interface (ETI) is the ability to turn on and off any of several video attributes, such as bold, dim, blinking, underlining, reverse video, and others. Future enhancements of ETI will include additional video attributes that enable your programs to use the color capabilities of a wide range of terminals.
5. In keeping with AT&T's ongoing internationalization of the UNIX system, future users will be able to use ETI with keyboards using foreign language character sets, such as Kanga.

Documentation

Essential documentation is provided with the SDS software package when purchased. Additional sets of the Software Development documentation (of which these *Release Notes* are a part) are available and can be ordered. See the Product Overview/Documentation Roadmap for more details. The Product Overview/Documentation Roadmap can be ordered separately by using the 9-digit number 999-300-527.

Appendix A: Documentation Updates

The following change pages reflect last minute changes to the UNIX System V/386 Release 3.2 documentation. These change pages should be inserted into the *Programmer's Reference Manual.*

AT&T UNIX SYSTEM V/386
RELEASE 3.2
PROGRAMMER'S REFERENCE MANUAL
UPDATES

This update involves the following actions:

1. **ACTION:** Replace RMDIR(2) pages 1 and 2 with the new pages.
2. **ACTION:** Replace SEMGET(2) pages 1 and 2 with the new pages.
3. **ACTION:** Replace UNLINK(2) pages 1 and 2 with the new pages.

NAME

rmdir – remove a directory

SYNOPSIS

int rmdir (path)
char *path;

DESCRIPTION

rmdir removes the directory named by the path name pointed to by *path*. The directory must not have any entries other than "." and "..".

The named directory is removed unless one or more of the following is true:

[EINVAL] The current directory may not be removed.

[EINVAL] The "." entry of a directory may not be removed.

[EEXIST] The directory contains entries other than those for "." and "..".

[ENOTDIR] A component of the path prefix is not a directory.

[ENOENT] The named directory does not exist.

[EACCES] Search permission is denied for a component of the path prefix.

[EACCES] Write permission is denied on the directory containing the directory to be removed.

[EBUSY] The directory to be removed is the mount point for a mounted file system.

[EROFS] The directory entry to be removed is part of a read-only file system.

[EFAULT] *Path* points outside the process's allocated address space.

[EIO] An I/O error occurred while accessing the file system.

[ENOLINK] *Path* points to a remote machine, and the link to that machine is no longer active.

[EMULTIHOP] Components of *path* require hopping to multiple remote machines.

In addition, a directory will not be removed when all of the following is true:

the parent directory has the sticky bit set
the parent directory is not owned by the user
the target directory is not owned by the user
the target directory is not writable by the user
the user is not super-user

SEE ALSO

mkdir(2).

rmdir(1), rm(1), and mkdir(1) in the *User's/System Administrator's Reference Manual.*

DIAGNOSTICS

Upon successful completion, a value of 0 is returned. Otherwise, a value of –1 is returned and *errno* is set to indicate the error.

NAME

semget – get set of semaphores

SYNOPSIS

#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>

int semget (key, nsems, semflg)
key_t key;
int nsems, semflg;

DESCRIPTION

The *semget* system call returns the semaphore identifier associated with *key*.

A semaphore identifier and associated data structure and set containing *nsems* semaphores [see *intro*(2)] are created for *key* if one of the following is true:

> *Key* is equal to **IPC_PRIVATE**.
>
> *Key* does not already have a semaphore identifier associated with it, and (*semflg* & **IPC_CREAT**) is "true".

Upon creation, the data structure associated with the new semaphore identifier is initialized as follows:

> **Sem_perm.cuid**, **sem_perm.uid**, **sem_perm.cgid**, and **sem_perm.gid** are set equal to the effective user ID and effective group ID, respectively, of the calling process.
>
> The low-order 9 bits of **sem_perm.mode** are set equal to the low-order 9 bits of *semflg*.
>
> **Sem_nsems** is set equal to the value of *nsems*.
>
> **Sem_otime** is set equal to 0 and **sem_ctime** is set equal to the current time.

The data structure associated with each semaphore in the set is not initialized. The function *semctl* with the command *setval* or *setall* can be used to initialize each semaphore.

The *semget* system call fails if one or more of the following is true:

[EINVAL] *Nsems* is either less than or equal to zero or greater than the system-imposed limit.

[EACCES] A semaphore identifier exists for *key*, but operation permission [see *intro*(2)] as specified by the low-order 9 bits of *semflg* would not be granted.

[EINVAL] A semaphore identifier exists for *key*, but the number of semaphores in the set associated with it is less than *nsems*, and *nsems* is not equal to zero.

[ENOENT] A semaphore identifier does not exist for *key*, and (*semflg* & **IPC_CREAT**) is "false".

[ENOSPC] A semaphore identifier is to be created, but the system-imposed limit on the maximum number of allowed semaphore identifiers system wide would be exceeded.

[EEXIST] A semaphore identifier exists for *key*, but [(*semflg* & **IPC_CREAT**) and (*semflg* & **IPC_EXCL**)] are "true".

SEE ALSO

intro(2), semctl(2), semop(2).

DIAGNOSTICS

Upon successful completion, a non-negative integer, namely a semaphore identifier, is returned. Otherwise, a value of –1 is returned, and *errno* is set to indicate the error.

NAME

unlink – remove directory entry

SYNOPSIS

int unlink (path)
char *path;

DESCRIPTION

unlink removes the directory entry named by the path name pointed to by *path*.

The named file is unlinked unless one or more of the following is true:

[ENOTDIR] A component of the path prefix is not a directory.

[ENOENT] The named file does not exist.

[EACCES] Search permission is denied for a component of the path prefix.

[EACCES] Write permission is denied on the directory containing the link to be removed.

[EPERM] The named file is a directory and the effective user ID of the process is not super-user.

[EBUSY] The entry to be unlinked is the mount point for a mounted file system.

[ETXTBSY] The entry to be unlinked is the last link to a pure procedure (shared text) file that is being executed.

[EROFS] The directory entry to be unlinked is part of a read-only file system.

[EFAULT] *Path* points outside the process's allocated address space.

[EINTR] A signal was caught during the *unlink* system call.

[ENOLINK] *Path* points to a remote machine and the link to that machine is no longer active.

[EMULTIHOP] Components of *path* require hopping to multiple remote machines.

A file will not be unlinked when all of the following is true:

the parent directory has the sticky bit set
the file is not writable by the user
the user does not own the parent directory
the user does not own the file
the user is not super-user

When all links to a file have been removed and no process has the file open, the space occupied by the file is freed and the file ceases to exist. If one or more processes have the file open when the last link is removed, the removal is postponed until all references to the file have been closed.

SEE ALSO

close(2), link(2), open(2).

rm(1) in the *User's/System Administrator's Reference Manual.*

DIAGNOSTICS

Upon successful completion, a value of 0 is returned. Otherwise, a value of −1 is returned and *errno* is set to indicate the error.

Appendix B: Installation Diskette Files

C Software Development Set Utility Package, Contents of 4 Diskettes

Size
Files
Name
Install
Remove
bin
bin/ar
bin/cc
bin/gencc
bin/as
bin/ld
bin/cprs
bin/conv
bin/dis
bin/dump
bin/list
bin/lorder
bin/mkshlib
bin/chkshlib
bin/nm
bin/size
bin/strip
bin/convert
bin/make
lib
lib/libld.a
lib/comp
lib/cpp
lib/optim
lib/cm4defs
lib/libPW.a
lib/crt0.o
lib/crt1.o
lib/crtn.o
lib/mcrt0.o
lib/mcrt1.o
lib/pcrt1.o
lib/libc.a
lib/libm.a
lib/libx.a
lib/libc_s.a
lib/basicblk
lib/pcrt0.o
etc
etc/install
usr
usr/bin
usr/bin/cflow
usr/bin/ctrace
usr/bin/ctcr
usr/bin/ctc
usr/bin/cxref
usr/bin/lex
usr/bin/lint
usr/bin/lprof
usr/bin/m4
usr/bin/prof
usr/bin/regcmp
usr/bin/admin
usr/bin/cdc
usr/bin/comb
usr/bin/delta
usr/bin/get
usr/bin/prs
usr/bin/rmdel
usr/bin/sact

usr/bin/unget
usr/bin/val
usr/bin/vc
usr/bin/what
usr/bin/sccsdiff
usr/bin/sdb
usr/bin/yacc
usr/bin/tsort
usr/bin/cb
usr/bin/cscope
usr/lib
usr/lib/libp
usr/lib/libp/libc.a
usr/lib/libp/libm.a
usr/lib/libp/libmalloc.a
usr/lib/libp/libx.a
usr/lib/ctrace
usr/lib/ctrace/runtime.c
usr/lib/libcrypt_i.a
usr/lib/llib-lc
usr/lib/libg.a
usr/lib/libl.a
usr/lib/libmalloc.a
usr/lib/llib-lmalloc.l
usr/lib/liby.a
usr/lib/dag
usr/lib/lpfx
usr/lib/help
usr/lib/help/lib
usr/lib/help/lib/help2
usr/lib/help/lib/help
usr/lib/help/ad
usr/lib/help/bd
usr/lib/help/cb
usr/lib/help/cm
usr/lib/help/cmds
usr/lib/help/co
usr/lib/help/de
usr/lib/help/default
usr/lib/help/ge
usr/lib/help/he
usr/lib/help/prs
usr/lib/help/rc
usr/lib/help/un
usr/lib/help/ut
usr/lib/help/vc
usr/lib/nmf
usr/lib/flip
usr/lib/xpass
usr/lib/xcpp
usr/lib/llib-port
usr/lib/llib-lc.ln
usr/lib/llib-lm
usr/lib/llib-port.ln
usr/lib/yaccpar
usr/lib/llib-lm.ln
usr/lib/lex
usr/lib/lex/ncform
usr/lib/lex/nrform
usr/lib/lint1
usr/lib/lint2
usr/lib/basicblk
usr/lib/libprof.a
usr/include
usr/include/a.out.h
usr/include/aouthdr.h
usr/include/ar.h
usr/include/assert.h
usr/include/core.h
usr/include/ctype.h
usr/include/dial.h
usr/include/dirent.h
usr/include/errno.h
usr/include/fatal.h
usr/include/fcntl.h
usr/include/filehdr.h
usr/include/ftw.h
usr/include/grp.h
usr/include/ieeefp.h
usr/include/ldfcn.h

usr/include/limits.h
usr/include/linenum.h
usr/include/macros.h
usr/include/malloc.h
usr/include/math.h
usr/include/memory.h
usr/include/mnttab.h
usr/include/mon.h
usr/include/nan.h
usr/include/nlist.h
usr/include/nsaddr.h
usr/include/nserve.h
usr/include/poll.h
usr/include/prof.h
usr/include/pwd.h
usr/include/regexp.h
usr/include/reloc.h
usr/include/rje.h
usr/include/scnhdr.h
usr/include/sd.h
usr/include/search.h
usr/include/setjmp.h
usr/include/sgtty.h
usr/include/signal.h
usr/include/stand.h
usr/include/stdio.h
usr/include/storclass.h
usr/include/string.h
usr/include/stropts.h
usr/include/strselect.h
usr/include/syms.h
usr/include/sys.s
usr/include/termio.h
usr/include/time.h
usr/include/tp_defs.h
usr/include/unistd.h
usr/include/ustat.h
usr/include/utmp.h
usr/include/values.h
usr/include/varargs.h

ORDER FORM

QUANTITY	TITLE/AUTHOR	TITLE CODES	PRICE		TOTAL
________	**1.** Portability Gde., 3/E, 7 volumes, X/OPEN	68581-8	$130.00g	paper	________
________	**2.** Portability Gde., 3/E: Sys. V Spec. Commands & Util., Vol. 1, X/OPEN	68583-4	$30.00g	paper	________
________	**3.** Portability Gde., 3/E: Sys. V Spec. Calls & Libraries, Vol. 2, X/OPEN	68584-2	$30.00g	paper	________
________	**4.** Portability Gde., 3/E: Sys. V Spec. Sup. Defns., Vol. 3, X/OPEN	68585-9	$30.00g	paper	________
________	**5.** Portability Gde., 3/E: Prog. Lang., Vol. 4, X/OPEN	68586-7	$30.00g	paper	________
________	**6.** Portability Gde., 3/E: Data Mgt., Vol. 5, X/OPEN	68587-5	$30.00g	paper	________
________	**7.** Portability Gde., 3/E: Networking, Vol. 6, X/OPEN	68588-3	$30.00g	paper	________
________	**8.** Portability Gde., 3/E: Operating Sys., Vol. 7, X/OPEN	68589-1	$30.00g	paper	________
________	**9.** UNIX® Sys. V/386 Prog.'s Guide, AT&T	94091-6	$34.95g	paper	________
________	**10.** UNIX® Sys. V/386 STREAMS Primer, AT&T	94087-4	$21.95g	paper	________
________	**11.** UNIX® Sys. V/386 STREAMS Prog.'s Guide, AT&T	94088-2	$24.95g	paper	________
________	**12.** UNIX® Sys. V/386 Network Prog.'s Guide, AT&T	94085-8	$24.95g	paper	________
________	**13.** UNIX® Sys. V/386 Prog.'s Ref. Manual, AT&T	94086-6	$34.95g	paper	________
________	**14.** UNIX® Sys. V/386 User's Ref. Manual, AT&T	94093-2	$34.95g	paper	________
________	**15.** UNIX® Sys. V/386 User's Guide, 2/E, AT&T	94092-4	$24.95g	paper	________
________	**16.** UNIX® Sys. V/386 Utilities Release Notes, AT&T	93612-0	$21.95g	paper	________
________	**17.** UNIX® Sys. V/386 Sys. Admin. Guide, AT&T	94089-0	$24.95g	paper	________
________	**18.** UNIX® Sys. V/386 Sys. Admin. Ref. Manual, AT&T	94090-8	$24.95g	paper	________
________	**19.** UNIX® Sys. V Prog.'s Guide, AT&T	94043-7	$36.95g	paper	________
________	**20.** UNIX® Sys. V STREAMS Primer, AT&T	94052-8	$21.95g	paper	________
________	**21.** UNIX® Sys. V STREAMS Prog.'s Guide, AT&T	94053-6	$24.95g	paper	________
________	**22.** UNIX® Sys. V Network Prog.'s Guide, AT&T	94046-0	$24.95g	paper	________
________	**23.** UNIX® Sys. V Prog.'s Ref. Manual, AT&T	94047-8	$36.95g	paper	________

________	**24.** UNIX® Sys. V User's Ref. Manual, AT&T	94048-6	$34.95g	paper	________
________	**25.** UNIX® Sys. V User's Guide, 2/E, AT&T	94054-4	$25.95g	paper	________
________	**26.** UNIX® Sys. V Utilities Release Notes, AT&T	94055-1	$21.95g	paper	________
________	**27.** UNIX® Sys. V Sys. Admin. Guide, AT&T	93613-8	$34.95g	paper	________
________	**28.** UNIX® Sys. V Sys. Admin. Ref. Manual, AT&T	93614-6	$24.95g	paper	________
________	**29.** The C Programming Language, 2/E, Kernighan/Ritchie	11037-9 11036-1	$40.00sf $28.00sf	cloth paper	________ ________
________	**30.** The C Programming Language, 1/E, Kernighan/Ritchie	11016-3	$27.00sf	paper	________
________	**31.** The C Answer Book, 2/E, Tondo/Gimpel	10965-2	$21.33sf	paper	________
________	**32.** The C Answer Book, 1/E, Tondo/Gimpel	10987-6	$21.00sf	paper	________
________	**33.** ANSI C: A Lexical Guide, Mark Williams Co.	03781-2	$35.00sf	paper	________
________	**34.** Doing Business with C, Swartz	21725-7	$26.95g	paper	________
________	**35.** Advanced C Programming, Rochkind	01024-9	$32.95g	paper	________
________	**36.** C Trainer, Feuer	10974-4	$24.33sf	paper	________
________	**37.** C: A Reference Manual, 2/E, Harbison et al.	10980-1	$25.95g	paper	________
________	**38.** C Companion, Holub	10978-3	$22.67sf	paper	________
________	**39.** Programming in C with a Bit of UNIX®, Moore	73009-3	$25.95g	paper	________
________	**40.** Learning To Program in C, Plum	52784-6	$33.00sf	paper	________
________	**41.** C Notes, Zahn	10977-7	$17.95g	paper	________
________	**42.** C Prog. in Berkeley UNIX® Envirmnt., Horspool	10997-5	$27.00sf	paper	________
________	**43.** C Prog.'s Handbook, Bolsky	11007-2	$22.95g	paper	________
________	**44.** Crafting C Tools, Campbell	18841-7	$25.95g	paper	________
________	**45.** C Puzzle Book, Feuer	10992-6	$25.00sf	paper	________
________	**46.** Numerical Software Tools in C, Kempf	62727-3	$28.00sf	paper	________
________	**47.** C Programming Guidelines, Plum	10999-1	$34.00sf	paper	________
________	**48.** A Software Tools Sampler, Miller	82230-4	$26.67sf	paper	________
________	**49.** Systems Software Tools, Biggerstaff	88176-3	$19.95g	paper	________
________	**50.** UNIX® Relational Database Mgt., Manis et al.	93862-1	$34.00sf	paper	________
________	**51.** UNIX® Prog. Envirmnt., Kernighan/Pike	93768-0	$24.95g	paper	________
________	**52.** Advanced UNIX® Prog., Rochkind	01180-9	$29.95g	paper	________

________	**53.** Portable C & UNIX® Sys. Prog., Lapin	68649-3	$24.95g	paper	________
________	**54.** UNIX® Sys. Software Readings, AT&T UNIX® Pacific Co.	93835-7	$21.95g	paper	________
________	**55.** UNIX® Sys. Readings & Applications, Vol. I, AT&T	93853-0	$19.00sf	paper	________
________	**56.** UNIX® Sys. Readings & Applications, Vol. II, AT&T	93984-3	$19.00sf	paper	________
________	**57.** *vi* User's Handbook, Bolsky	94173-2	$18.95g	paper	________
________	**58.** Guide to *vi*, Sonnenschein	37131-0	$19.95g	paper	________
________	**59.** *Troff* Typesetting, Emerson et al.	93095-8	$27.95g	paper	________
________	**60.** Intro. to Compiler Construction, Schreiner et al.	47439-5	$38.00sf	cloth	________
________	**61.** UNIX® C Shell Field Gde., Anderson et al.	93746-6	$27.95g	paper	________
________	**62.** Preparing Documents with UNIX®, Brown et al.	69997-5	$27.95sf	cloth	________
________	**63.** Oper. Sys. Des. & Implementation, Tanenbaum	63740-5	$42.00sf	cloth	________
	MINIX for the IBM PC/XT/AT:				
	1) 512K for the AT	58441-7	$110.00sf	software	________
	2) 640K for the PC/PC XT	58442-5	$110.00sf	software	________
	3) MINIX for the IBM PC/XT/AT Ref. Manual	58440-9	$32.00sf	paper	________
________	**64.** Operating Sys. Design, Vol. I: XINU Approach (PC Ed.), Comer/Fossum	63818-9	$44.00sf	cloth	________
________	**65.** Operating Sys. Design, Vol. I: XINU Approach, Comer	63753-8	$46.00sf	cloth	________
________	**66.** Design of UNIX® O/S, Bach	20179-8	$20.00sf	paper	________
________	**67.** Oper. Sys. Des., Vol. II: Internetworking with XINU, Comer	63741-3	$46.00sf	cloth	________
________	**68.** Internetworking with TCP/IP, Comer	47015-3	$36.00sf	cloth	________
________	**69.** UNIX® Sys. V Network Prog.'s Guide, AT&T	94046-0	$24.95g	paper	________
________	**70.** UNIX® Admin. Guide for Sys. V, Thomas/Farrow	94288-8	$34.95g	paper	________
________	**71.** UNIX® Sys. V Sys. Admin. Guide, AT&T	93613-8	$34.95g	paper	________
________	**72.** UNIX® Sys. V Sys. Admin. Ref. Manual, AT&T	93614-6	$24.95g	paper	________
________	**73.** UNIX® Sys. User's Handbook, Bolsky	93776-3	$18.95g	paper	________
________	**74.** Making Use of the UNIX® O/S, Budgen	R4434-8	$24.00sf	paper	________

________	**75.** UNIX® for People, Birns et al.	93744-1	$28.00g	paper	________
________	**76.** UNIX® Primer, Lomuto et al.	93773-0	$28.95g	paper	________
________	**77.** UNIX® RefGuide, McNulty	93895-1	$27.95g	paper	________
________	**78.** DOS: UNIX® Systems, Seyer/Mills	21864-4	$27.95g	paper	________
________	**79.** Beyond Photography: The Digital Darkroom, Holzmann	07441-9	$26.95g	paper	________
	Digital Darkroom Software, Holzmann	21274-6	$25.00g	software	________
________	**80.** Clipper™ 32-Bit Microproc. User's Manual, Fairchild	13805-7	$23.95g	paper	________
________	**81.** Programmer's Survival Guide, Ruhl	73037-4	$16.95g	paper	________

SPECIAL OFFER!
When ordering 3 or more copies (of the same or different titles) take 10% off the total list price. When ordering 5 or more copies (of the same or different titles) take 15% off the total list price.

SAVE!
If payment accompanies order, plus your state's sales tax where applicable, Prentice Hall pays postage and handling charges. Same return privilege refund guaranteed.

□ PAYMENT ENCLOSED–shipping and handling to be paid by publisher (please include your state's sales tax where applicable).

□ SEND BOOKS ON 15-DAY TRIAL BASIS & bill me (with small charge for shipping and handling).

Name ____

Address ____

City ____________________ State ____________ Zip ____________

I prefer to charge my □ VISA □ MasterCard

Card Number ____________________ Expiration Date ____________________

Signature ____

All prices in this catalog are subject to change without notice.

MAIL TO:
Prentice Hall, Book Distribution Center, Route 59 at Brook Hill Drive, West Nyack, NY 10995

Available at better bookstores or direct from Prentice Hall.
Attention Corporate Customers: For orders in excess of 20 copies, please call 201-767-2498.
For orders of fewer than 20 copies please call 201-767-5937.

For Government Orders please contact LEARNING TRENDS, 201-767-5994.

D-UNIX-BH(2)

AT&T UNIX System V/386 Library

- UNIX System V/386 Release 3.2 Utilities Release Notes **AT&T**
- UNIX System V/386 Release 3.2 Streams Primer **AT&T**
- UNIX System V/386 Release 3.2 User's Guide **AT&T**
- UNIX System V/386 Release 3.2 Programmer's Reference Manual **AT&T**
- UNIX System V/386 Release 3.2 Streams Programmer's Guide **AT&T**
- UNIX System V/386 Release 3.2 Network Programmer's Guide **AT&T**
- UNIX System V/386 Release 3.2 Programmer's Guide Vol. I **AT&T**
- UNIX System V/386 Release 3.2 Programmer's Guide Vol. II **AT&T**
- UNIX System V/386 Release 3.2 System Administrator's Guide **AT&T**
- UNIX System V/386 Release 3.2 System Administrator's Reference Manual **AT&T**

PRENTICE HALL, Englewood Cliffs, N.J. 07632

ISBN 0-13-944943-4