

Note: Shared mem / Swap Space problem fixed in 4.0 - rjh.



---

## Release 4.0 Change Notes *for the Sun Workstation®*

**UNIX** is a registered trademark of AT&T.

**Sun Workstation, SPARC, SunView, Sun-1, Sun-2, Sun-3, and Sun-4** are trademarks of Sun Microsystems, Incorporated.

**Multibus** is a trademark of Intel Corporation.

**VMEbus** is a trademark of VMEbus Manufacturers Group.

Copyright © 1987, 1988 by Sun Microsystems, Inc.

This publication is protected by Federal Copyright Law, with all rights reserved. No part of this publication may be reproduced, stored in a retrieval system, translated, transcribed, or transmitted, in any form, or by any means manual, electric, electronic, electro-magnetic, mechanical, chemical, optical, or otherwise, without prior explicit written permission from Sun Microsystems.

---

# Contents

<b>Chapter 1 Release 4.0 Change Notes</b> .....	3
1.1. Introduction .....	3
Compatibility .....	3
1.2. New Hardware Support .....	3
1.3. System Software Changes and Upgrades .....	3
NFS and the File System Reorganization in 4.0 .....	3
Changes to boot .....	4
General Software Changes .....	4
signal handlers .....	4
getdents .....	4
O_SYNC .....	4
Time Zones .....	4
open files per process .....	5
C .....	5
General C Library Changes .....	5
printf .....	5
regex .....	6
New hostent structure .....	6
1.4. Utilities .....	6
Telnet improvements .....	6
FTP fully implemented .....	6
TFTP defaults to secure .....	6
Verbose etherfind .....	6
on command suspension .....	6

format .....	6
<i>suninstall</i> .....	7
<b>Mail Transport System</b> .....	7
Mailboxes on servers .....	7
Error message improvements .....	7
Mail Exchanger support .....	7
Inverted alias mapping .....	7
Automatic Domain Configuration .....	7
<b>1.5. New Security Features</b> .....	7
<b>1.6. System V Compatibility</b> .....	8
New cron and at .....	8
ed .....	8
sh .....	8
curses/terminfo .....	8
System V STREAMS Interface .....	9
<b>4.3BSD Enhancements</b> .....	9
su .....	9
tcopy(1) .....	9
fsck .....	9
getty .....	9
New Interface structure .....	9
New buffering conventions .....	10
TCP urgent data .....	10
TCP performance .....	10
IP options .....	10
Full ICMP support .....	10
Full IP subnetting .....	10
NIT Streams instead of Sockets .....	10
Alternative Protocol Architectures .....	10
/etc/ttys compatibility .....	10
syslog compatibility .....	11
syslog daemon .....	12
inetd compatibility .....	12

Line Printer Spooler .....	12
select .....	12
System V and 4.3BSD combined functionality .....	12
Shared Libraries and Virtual Memory .....	13
1.7. Light weight processes .....	13
1.8. Programming Environment Changes and Upgrades .....	14
Changes to the C compiler .....	14
Changes to lint .....	14
1.9. SunView Enhancements .....	15
Summary of New SunView Features from Release 3.4 .....	15
Walking Menus are Now the Default .....	15
New Text Menu .....	16
Text Menu Layout .....	16
New Names .....	16
Find and Replace Pop-Up Frame .....	16
Delimiter Matching .....	17
Fields in Text windows .....	18
Word Wrap Mode .....	18
'Extras' Menu .....	19
Keyboard Control of the Caret and Editing .....	19
Alerts .....	21
Keyboard Accelerators for Alerts .....	22
Shadowed Frames and Menus .....	22
File Size Limit on Editing Logs .....	22
Buffer Size Limit when Editing Files .....	23
New mailtool .....	23
Simplified Control Panel .....	23
Multiple Composition Windows .....	23
Separate Mail Composition Window .....	23
Reply Control Panel .....	24
Hierarchical Folders Menu .....	24
Input Focus .....	24
Fields in Outgoing Messages .....	24

New Mail .....	25
Miscellaneous .....	25
8-Bit Support in <code>shelltool</code> and <code>cmdtool</code> .....	25
Underlining and Inverse in <code>shelltool</code> .....	25
Frame Menu Changes .....	26
'Props' Item in the Menu .....	26
Other Name Changes .....	27
Unconstrained Move and Resize .....	27
Other SunView Changes .....	27
Files Renamed .....	27
New SunView "Root" Menu .....	27
SunView Changes Visible to the Programmer .....	27
Alerts .....	27
More File Descriptors .....	27
Lines in Menus .....	27
Props Attribute .....	27
Shadowed Frames .....	27
SunView Incompatibilities .....	28
1.10. Firmware and Diagnostics Changes and Upgrades .....	29
1.11. PROM Changes for Sun-4 Architecture .....	29
Deleted Commands .....	29
Changed Commands .....	29



---

## Release 4.0 Change Notes

### 1.1. Introduction

#### Compatibility

Release 4.0 software is designed to run on Sun-2, Sun-3, and Sun-4 systems. With the elimination of the ND partition and the reorganization of the file system, Sun now has true heterogeneity across these architectures.

#### 1.2. New Hardware Support

Software support is available for the following new hardware:

- The Sun-4 Workstation
- SCSI 1/2 Inch Tape Support
- Interphase VME SMD disk controller
- Xylogics 751 Support
- Sun Type-4 Keyboard

The keyboard is compatible with all Sun-2, Sun-3, Sun-4 and Road Runner workstations.

#### 1.3. System Software Changes and Upgrades

##### NFS and the File System Reorganization in 4.0

ND partitioning has been eliminated and the filesystem has been reorganized. Now you can use the Sun Network File System (NFS) exclusively to support different architectures. The new layout provides the following:

- Easier maintenance of servers and clients.
- Easier mixing of remote and local filesystems.
- Cleaner support of multiple architectures.
- Minimal disruption to existing programs
- Minimal symbolic link confusion.

For complete information on the new filesystem layout, see *Installing the Sun Operating System*, Chapter 1.



## Changes to `boot`

In previous releases booting a stand-alone workstation or a server from a local disk has been done through code in `boot` which knew how to access and interpret a file system on that disk.

The following changes have been made to `boot`.

- ND code has been eliminated, since servers no longer support ND operations.
- The program `boot` now understands how to perform NFS file operations over the net to a server with which the client workstation is registered.

For a complete explanation of this see *System and Network Administration for the Sun Workstation* (800-1733).

## General Software Changes signal handlers

Functions that are specified as handlers for signals are now expected to be of type "function returning 'void'", not "function returning 'int'". This is the type specified by the current IEEE POSIX standard draft. Since no use is made of the return value of a signal handler, "void" is the correct type of such a function, not "int".

Code written assuming that these functions were of type "int" will continue to run and not be recompiled. If this code is recompiled, it does not need to be changed, although warning messages about type clashes will be printed.

## `getdents`

A new SVID compatible `getdents` system call has been added to read directory entries. As with the old `getdirentries` system call, direct use of this call should be avoided. Use the `directory(3)` routines instead.

## `O_SYNC`

The flag `O_SYNC` may be specified in the `flags` argument to "open" and "fcntl"; if this flag is set on a file descriptor that refers to a regular file, all "write"s to that file will block until the data is completely written to disk. This happens regardless of whether the file is on a local file system or is being accessed over NFS.

## Time Zones

The handling of time zones has been upgraded in the following ways:

- The offset from Greenwich Mean time of a time zone.
- The rules that indicate when daylight savings time starts and ends.
- The amount of time that the clock shifts when daylight savings time starts and ends.
- The names of the time zone during standard and daylight savings time are now read from files.

By default, a file that contains the rules for the local time zone is used. If the `TZ` environment variable is set to the name of a file containing the rules for a different time zone, that file is used instead; this applies to all utilities except selected ones, such as `uucico`, that must always use the local time zone.

These files are generated from a textual description of the rules. The text files from which the distributed set of rule files are generated, and the command `zic` used to generate them, are provided.

New routines `timelocal` and `timegm` are provided that convert a date and time, specified as month/day/year/hour/minute/second, from local or Greenwich Mean time respectively to the standard UNIX system date/time format. They perform the inverse of the conversion performed by the `localtime` and `gmtime` routines. These routines use the information from the appropriate time zone file; applications should not perform this conversion themselves.

The structure returned by `localtime` now contains extra fields that indicate the name and the offset from GMT of the current time zone at the time specified by the argument to `localtime`.

The command `tzsetup` attempts to set the kernel's notion of the offset from GMT and DST rule type based on the default time zone rules; this is done so that binaries built prior to 4.0 will run. In some cases, there is no set of DST rules that will work; if this is the case, `tzsetup` will indicate that DST is not observed. This program is run at boot time; there is no longer any need to specify the offset or DST rule type when building a kernel.

open files per process

Before 4.0, the limit was 30 open files per process. In 4.0 the limit is 64 open files per process.

C

The file system code has been changed to permit file names to contain characters with the 8th bit set. Previously, attempts to create or manipulate such files were rejected. Note that such files are impossible to remove using the C shell, as the C shell still uses the 8th bit to quote characters; the Bourne shell, which has been upgraded in 4.0, and now does not use the 8th bit to quote characters, must be used to remove these files.

In a C program, if the main function "main" returns, its return value is used as the exit status of the program. Previously, 0 was used as the exit status if `mail` returned; this change makes SunOS more compatible with 4.3BSD and System V. Note, however, that some erroneous programs do not return any value from `main`. Thus, it may return a non-zero exit status.

General C Library Changes  
`printf`

The `printf` routines now support the `%i` and `%li` format items; they are synonyms for `%d` and `%ld`. The `scanf` routines now support `%i`, `%hi`, `%li`, and `%n`. `%i`, `%hi`, and `%li` are similar to `%d`, `%hd`, and `%ld`, respectively, except that if the number being converted begins with "0x" or "0X" it is assumed to be decimal, and if the number begins with "0" it is assumed to be octal. `%n` returns the total number of characters that have been scanned so far by the current `scanf` call. These are from the current ANSI C draft standard.

`regex` The `regex` regular-expression scanner has been upgraded to support the "<" and ">" characters from `vi`. If a regular expression is enclosed in "<" and ">", it is constrained to match a "word"; the "<" must match the beginning of a "word", i.e. the beginning of a line or just before a letter, digit, or underline and after a character (not one of these), and the ">" must match the end of a "word".

New `hostent` structure Any programs that call `gethostbyname`, `gethostbyaddr` or `gethostent` need to be recompiled. The new `hostent` structure contains a list of addresses instead of just one. Many programs, however, only look at the first entry.

## 1.4. Utilities

Telnet improvements The SunOS 4.0 telnet daemon supports terminal type negotiation. The telnet program now supports optional local X-ON/X-OFF flow control. See `telnet(1)`.

FTP fully implemented The File Transport Protocol is more fully implemented in SunOS 4.0. Several new commands have been added. The default transfer type now is ASCII, so transfer of binary files requires an explicit command. See `ftp(1)` and `ftp(8)`.

TFTP defaults to secure The Trivial File Transfer Protocol should only be used to bootstrap machines over the network. Use programs such as `FTP` or `rdist` to transfer other files. The default transfer mode to TFTP is now ASCII, so binary files need an explicit command. See `TFTP(1)`.

Verbose `etherfind` The `etherfind` program has some new options that print out much more information. This helps in tracking down network problems. `Etherfind` also now understands Sun RPC headers. Refer to `etherfind(8)`.

`on` command suspension Commands started with the remote execution service, the `on` program, can be suspended (for example, by typing `[Ctrl-Z]`) and continued.

`format` `format` is a SunOS™ utility that allows you to format, label, repair and analyze disks on your Sun system. Unlike previous maintenance programs, `format` runs under SunOS™. This offers a user friendly, menu based interface to disk maintenance. For complete instructions on how to use `format` see *System and Network Administration for the Sun Workstation*.

*suninstall*

*suninstall* is the new installation tool replacing *Setup*. It is a terminal based interface that provides a user-friendly installation editor which allows you to customize your systems and configurations.

**Mail Transport System**

## Mailboxes on servers

Workstations can use NFS to mount mailbox directories from file servers. Outgoing mail can be sent through the machine from which the mailbox directories are mounted. Typical diskless workstations should no longer need to run `sendmail` daemons.

## Error message improvements

Several minor improvements were made to the error messages generated by `sendmail`. For example, more messages from mailing lists are sent to the owner of the list instead of the sender of the message.

## Mail Exchanger support

In addition to the normal version of `sendmail` that uses the Yellow Pages to resolve names, another version of `sendmail` is supplied in SunOS 4.0 that uses the domain name resolver directly. This version can be used on the Defense Data Network and to access Mail Exchanger (MX) records.

## Inverted alias mapping

A new mechanism is provided in `sendmail` that can rewrite an address through any Yellow Pages' map. A new map is provided that contains the inverse of the `mail.aliases` map, so that mail going outside of a domain can be simplified.

## Automatic Domain Configuration

`sendmail` can use the domain name set in the kernel instead of having to modify each `sendmail.cf` file.

**1.5. New Security Features**

Sun's Release 4.0 provides the following security enhancements:

- Improved network security, with DES authentication of user and host, and public key cryptography.
- An install-time option to run the system at a moderate level of security, patterned after the widely accepted C2 classification.†

To improve network security, a new set of RPC library routines offers DES authentication to check the validity of both user ID and host address. Previously, UNIX authentication checked only the validity of user ID, which allowed users to impersonate each other over the NFS.

---

† Defined by the National Computer Security Center (a branch of the NSA), the C2 category adds password hiding and security auditing to UNIX.

To meet C2 specifications, Sun's operating system was extended to provide improved password security, and flexible, fail-proof auditing of all events that affect security. Other extensions involve single-user booting, enhanced yellow page security, and stricter permission settings for system files. For more information, see *Security Features on the Sun Workstation*.

*NOTE* While we believe that Release 4.0 meets C2 specifications, Sun Microsystems does not intend to have the system actually certified as C2 secure.

## 1.6. System V Compatibility

New cron and at

The System V, Release 3 (NOT System V, Release 2) cron, at, and crontab utilities have been provided.

The form of the crontab file has not changed; however, each user may now have a crontab file, and jobs run from that crontab file run with that user's privileges. The crontab files are in the directory "/etc/spool/cron/crontabs"; the crontab file for a particular user has that user's login.name as its file name.

The crontab command is used to create, update, delete, and list crontab files; these files must not be edited directly, as cron does not reread them automatically.

The at command is upwardly compatible with the previous version. However, it now supports more than one job queue. One queue that it supports is a "batch" queue; jobs submitted to this queue run "as soon as possible". at can be told to limit the number of jobs that will run simultaneously in any particular queue, so this queue can be used to limit the number of simultaneous background jobs running on the system.

The set of users that may use cron and crontab, or at, may be restricted by the system administrator. By default, all users may use either cron or at.

ed

The /usr/5bin/ed and /bin/ed commands have been merged into one version of ed that is compatible both with the 4.3BSD and the System V, Release 3.0 ed.

sh

The shell has been upgraded to the System V, Release 3.1 version. This version no longer uses the 8th bit of a character to quote that character, so that it can handle command names and arguments containing characters with the 8th bit set.

curses/terminfo

The curses library and terminfo database have been upgraded to the System V, Release 3.1 versions. This version fixes many bugs, and is faster and more compact than the previous version; it also includes many new capabilities. It supports eight-bit character sets.

**System V STREAMS Interface**

STREAMS is a facility with a set of tools for development of UNIX system communication services from networking protocol suites to individual device drivers.

STREAMS defines standard interfaces for character I/O within the UNIX kernel, and between the kernel and the rest of the UNIX system. STREAMS consists of a set of system calls, kernel resources, and kernel utility routines. STREAMS is not limited to a specific network architecture. STREAMS offers the following major features:

- Buffer management
- Flow control
- Scheduling
- Multiplexing
- Asynchronous operations of STREAMS and user processes
- Error and trace loggers for debugging and administrative functions.

There is a new STREAMS-based `tty` driver.

There is a STREAMS-based Network Interface Tap (NIT) The old provisional socket-based facility has been replaced by a set of STREAMS modules and drivers that collectively provide a superset of the old versions functionality. A major enhancement is the addition of a packet filtering module that makes selecting relevant packets out of all incoming packets much more efficient.

**4.3BSD Enhancements**`su`

If the 'wheel' group (group 0) has members, only they can `su` to 'root', even with the root password. Successful and unsuccessful attempts to `su` to root are logged to the `syslog` daemon.

`tcopy(1)`

`tcopy(1)` is a new tape copying program that preserves tape blocking.

`fsck`

`fsck` can now create and grow directories, so that it can rebuild the root of a file system as well as create and enlarge the "lost+found" directory as necessary.

`getty`

`getty` was also upgraded to the 4.3BSD version; it now also uses `/etc/ttytab`.

**New Interface structure**

The kernel `ifnet` structure has changed as in 4.3BSD to allow more generality, such as the use of a single interface by several different address families. All Sunlink products need new releases for SunOS 4.0.

- New buffering conventions      The kernel uses 4.3BSD conventions for handling `mbuf` structures within the socket system. Customer network drivers may need to be rewritten.
- TCP urgent data                  The interpretation of TCP Urgent data has been changed to be closer to the official specification.
- TCP performance                 The TCP software now estimates the round-trip time as well as the variation of round-trip time, so that good performance is maintained on both fast and slow networks.
- IP options                        More general support is provided for options at the Internet level.
- Full ICMP support                Earlier releases did not implement all of the ICMP (Internet Control Message Protocol). This caused some problems on multi-vendor networks such as the Defense Data Network.
- Full IP subnetting               The restrictions on IP subnets in Release 3.3 have been removed. In SunOS 4.0 each interface can have its own network mask. A new YP map `netmasks.byaddr` is used to enable subnets.
- NIT Streams instead of Sockets   The Network Interface Tap is now available through the System-V STREAM mechanism instead of the socket mechanism. A general packet filter is now provided as well as a STREAM module.
- Alternative Protocol Architectures      The kernel can now be configured to support other protocols from a binary release without having to recompile the source to the Ethernet driver. This should ease the installation of Sunlink products, for example.
- `/etc/ttys` compatibility        The system uses a new style `/etc/ttys` file that is now called `/etc/ttytab`. `init` reads this new file and writes an `/etc/ttys` file with the same relative positions within the file for each tty. The `/etc/ttys` file written by `init` is read-only. Users are not allowed to modify it directly since doing so does not have any effect on `init`. `/etc/ttys` is rewritten whenever `init` reads `/etc/ttytab`. Old Sun programs still run compatibly with the `/etc/ttys` emulation.
- Since `init` no longer uses `/etc/ttys`, any administrative procedures that modify it must be changed to modify `/etc/ttytab` instead.
- The new `/etc/ttytab` file format is completely compatible with 4.3BSD, but the file name is different. We might allow `/etc/ttys` to be a symlink to `/etc/ttytab` (or vice versa) in case complete 4.3BSD compatibility is important. Doing this, however, breaks compatibility with old Sun programs. See `tts(5)` and `ttytab(5)` for more information.

`init` is upgraded to the 4.3BSD version. It now runs programs other than `getty`, and starts up a window system under which the program `init` is run. It is now controlled by a file `/etc/ttytab` rather than `/etc/ttys`. This file has the same format as the 4.3BSD `/etc/ttys` file, which includes the information that used to be in `/etc/ttytype` and `/etc/securetty`. For binary compatibility with SunOS 3.x programs, the file is now named and `init` creates a read-only version of `/etc/ttys` in the old format, if `/etc/ttys` does not exist or exists but is not a link (hard or symbolic) to `/etc/ttytype`. Installations that do not require binary compatibility with 3.x, and that want to call this file `/etc/ttys`, can make `/etc/ttys` a symbolic link to `/etc/ttytab`.

## syslog compatibility

In 4.0, the `syslog` facility has been upgraded to be compatible with 4.3BSD. This change has caused some incompatibilities which are described below.

- pre-4.0 program logging to 4.0 syslog daemon

Pre-4.0 programs will log messages with no facility code but with priorities in the range 1 - 9. Since the 4.0 syslog accepts priorities in the range 0 - 7, priorities 8 (`LOG_INFO`) and 9 (`LOG_DEBUG`) will look like priorities 0 (`LOG_EMERG`) and 1 (`LOG_ALERT`) from facility 1 (`LOG_USER`). Unfortunately, this will have the effect of making these low priority messages seem to be much higher priority than they really are.

Also, almost all of the values for logging levels have changed. This will cause, for instance, messages logged at the old `LOG_CRIT` level to be logged at the new `LOG_NOTICE` level. In general, old log messages will appear to be less important than intended.

The 4.0 syslog daemon will force messages that claim to be from the `LOG_KERNEL` facility to look like they came from the `LOG_USER` facility, unless they come from the local kernel. The `syslog.conf` file on the "loghost" machine will be set up to log all `LOG_USER` messages in the log file used to log `LOG_MAIL` messages.

- 4.0 program logging to pre-4.0 syslog daemon

All 4.0 programs using `syslog` will send their log messages to the local `syslog` daemon. The default 4.3BSD `syslog` configuration file causes all `syslog` messages to be logged in local files, although it does provide a facility to forward the `syslog` message on to a `syslog` daemon on another machine. `sendmail` log messages and "authorization system" log messages are forwarded to "loghost".

These forwarded messages include a facility code in their log message. Except for `LOG_USER|LOG_EMERG (== 8)` and `LOG_USER|LOG_ALERT (== 9)` (which, by default, will not be forwarded to the `syslog` daemon at "loghost"), these will all cause the priority field in the message to be two digits. The old `syslog` daemon does not understand multi-digit priority fields, and so will log the message with a



default priority of LOG\_ERR (== 4). Using the default configuration file, this will cause the message to be logged with all the sendmail log messages in /usr/spool/log/syslog. For more information, see syslog(3) and syslog(8).

#### syslog daemon

The syslog daemon has been upgraded to the 4.3BSD version. It reads kernel printf messages from a new device and logs them. Many system daemons have been changed to log to the syslog daemon as well.

#### inetd compatibility

The new inetd uses a different file format in a file of a different name (/etc/inetd.conf vs. /etc/servers).

The new inetd uses different conventions to start the programs it runs. The old inetd would call the program with a single argument which contained the port number the connection was on. The new inetd gets the port number by using getpeername().

The new inetd is completely compatible with the 4.3BSD inetd. Old Sun programs should work without change with the new inetd, as long as the /etc/inetd.conf file has been set up properly. For more information, see inetd.conf(5) and inetd(8).

inetd now supports some of the trivial Internet services such as echo, day, and character generator internally.

#### Line Printer Spooler select

The "select" system call has been upgraded to the 4.3BSD version; it can now handle more than 32 file descriptors. Macros have been provided to manipulate sets of descriptors. (This one should go along with the item about the increase from 30 to 64 file descriptors.)

#### System V and 4.3BSD combined functionality

A new terminal driver is provided. It fully supports the functionality specified in the System V Interface definition, as well as all the functionality of the old V7/4BSD terminal driver, with the exception of the "tilde" mode and the "delayed suspend" character. Those are no longer supported.

The driver can support a full 8-bit data path; it need not strip off the 8th bit on input or output. If a terminal supports an 8-bit character set, it can be operated in 8 bits, no parity bit mode.

If you have a terminal that supports 8-bit characters, and you wish to use it, you should set the terminal modes to a mode that supports an 8-bit data path. This can be done with the command "stty pass8", or can be specified using the "p8" capability in "/etc/gettytab". This even applies to terminals where the 8th bit is controlled by a "meta" key; one should not rely on editors such as EMACS to switch the data path between 7 and 8 bits, as this may not work in all cases.

The "stty" command has been updated to print and set all the new modes.

Some additional functionality is available in the pseudo-tty driver:

- If the baud rate is set to 0 with an "ioctl", all subsequent I/O on the controller gives an EIO error, just as if the slave side had been closed. This is analogous to setting the baud rate to 0 on a real terminal, which causes DTR to drop and causes what's on the other end to hang up; the EIO will normally be treated as a signal to the process on the controller side to exit.
- The number of pseudo-terminals may be configured without source code.

## Shared Libraries and Virtual Memory

Shared libraries extend the benefits obtained from sharing code between processes running the same program to processes running different programs by sharing libraries common to them.

The mechanism that provides our memory sharing is a new Virtual Memory (VM) system for SunOS. The principal features of the new system include:

- file mapping as its principal mechanism, accessed by programs through the *mmap(2)* system call
- sharing at the granularity of a file page
- a per-page copy-on-write facility to allow run time modification of a shared object without affecting other users of the object.

The new VM system uses these features internally, so that the act of *exec'ing* a program is reduced to the establishment of copy-on-write mappings to the file containing the program. A "shared library" is added to the address space in exactly the same way, using the general file mapping mechanism.

## 1.7. Light weight processes

The 4.0 *lightweight process library* provides primitives for manipulating threads of control, as well as for managing events (interrupts and traps). It is an excellent abstraction for implementing service processes which must maintain state for multiple connections, and for programs which manage asynchrony. At present, there is no kernel support for lightweight processes, so concurrent system calls must be implemented by forked UNIX processes. However, both preemptive and non-preemptive scheduling are possible. Briefly, the primitives supported by the library include:

1. Thread creation, destruction, status gathering, priority manipulation, sleeping, suspension and resumption. It is possible to implement your own scheduler as a lightweight process. For example, a high priority lwp can implement time-slicing for lower priority lwps by periodically waking up to reshuffle the lower priority lwp queues. The clock is multiplexed, so many threads can sleep concurrently for different time intervals.
2. Individualized context switching (e.g., it is possible to specify that a given set of threads will touch floating point registers and only those threads will context switch these registers).

3. Monitors and condition variables to synchronize threads.
4. Extended rendezvous (message send-receive-reply) for communication between threads.
5. An exception handling facility that provides both *notify* and *escape* exceptions.
6. A way to map interrupts (asynchronous signals) into extended rendezvous.
7. A way to map traps (synchronous signals) into exceptions.
8. Utilities to allocate red-zone-protected stacks, and to provide some stack integrity checking for environments that lack sophisticated memory management.
9. A non-blocking IO library is available that simulates the effect of concurrent system calls by using asynchronous, non-blocking IO.

*NOTE* For more information, see *lightweight processes tutorial* and `man (31)`.

## 1.8. Programming Environment Changes and Upgrades

### Changes to the C compiler

The following are new changes and enhancement to the C compiler:

- now has an "opaque pointer" type of `(void *)` that is conformable with any other pointer type. It may be assigned to or from any other pointer type without a warning.
- no longer accepts "old-fashioned initialization" and "old-fashioned assignment operators" in the language.
- treats enum types as int.

The C preprocessor has been upgraded to support the `#elif` control line from the proposed ANSI C standard. It now predefines `"sparc"` on Sun-4 computers. A new command option `-B` causes it to handle the C++ comment indicator `"//"`. This symbol, and everything after it on a line, is treated as a comment.

### Changes to lint

Lint incorporates the above changes from the C compiler. The opaque pointer and the enum as int changes are disabled by `-p`.

The new version takes the following new flags `-target=foo` and `-host=bar` where `foo` and `bar` are restricted to `"sparc"`, `"mc68020"`, `"mc68010"`. Both `-host` and `-target` default to the machine type that you are on. For portability help between Sun-3's and SPARC systems you can specify `-host=mc68020` if you are running on a SPARC system or `-target=sparc` if you are running on a Sun3. These flags are for specific portability between these machines and should not be used with the `-p` flag.

Detects and flags different alignment of structure members between Sun-2 C, Sun-3 C, and SPARC C (when host and target are different).

Detects and flags possible alignment problems on structure-pointer coersions. The old version assumed all structure pointers to have the same alignment.

Treats long type as `int`, and unsigned long as unsigned `int`.

Treats a 0 supplied as a parameter value as being conformable with any pointer (disabled by `-p`).

No longer issues error messages about nonportable character comparisons and about functions that return values which are always ignored.

Issues a better message, when using the `-x` flag, about external declarations in ".h" files. The previous version could not figure out the file name, so printed ??? in its place.

Allows `/*VARARGS0*/`. The old version treated this as the absence of `varargs`.

The preprocessor treats the empty comment, `/**/`, just as it is treated by the C compiler.

## 1.9. SunView Enhancements

Release 4.0 has substantial changes in the SunView user interface which was introduced with Release 3.0 and refined in subsequent 3.x releases. This section describes the enhancements made to SunView in Release 4.0. You should read the *Release 3.2* and *Release 3.4* manuals to learn the improvements made to SunView in those two releases.

### Summary of New SunView Features from Release 3.4

In case you do not have the *Release 3.4 Manual*, here are the main new features it introduced to SunView.

- `cmdtool(1)` supports `vi`, `more`, `man`, `su`, and other programs that use "raw" mode and full-screen terminal mode.
- You can set your menus to be *Stay\_up* in `defaultsedit(1)` (along with many other menu defaults). This lets you release the right mouse button while choosing from a menu.

### Walking Menus are Now the Default

In previous releases of SunView, the default menu style in `sunview(1)` was the old-style stacking menus from SunWindows to ensure visual fidelity with Release 1.x and 2.x SunWindows applications.

The default style is now the walking menu style introduced in Release 3.0.

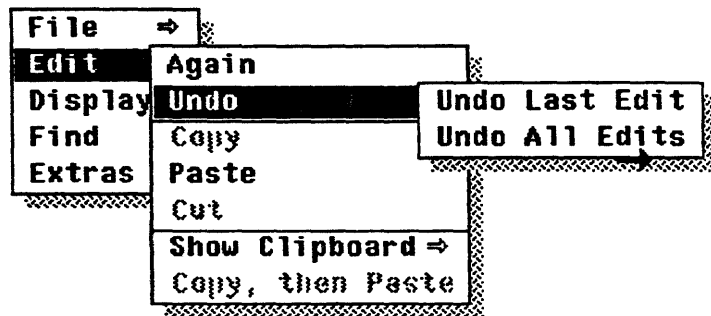
You can set the default style back to stacking menus by disabling *Walking\_Menus* in the *SunView* category of `defaultsedit(1)`; however, other menu defaults have no effect with old-style menus.

---

Since the default value for *SunView/Walking\_Menus* has changed in 4.0, `defaultsedit` will always write out your choice for *SunView/Walking\_Menus*. This forces tools that haven't been recompiled for 4.0 to pick up the

**New Text Menu**

The menu (shown below) in the text window has been expanded and reorganized. Many tools use the text window ( `textedit(1)`, `cmdtool(1)`, `mailtool(1)` and `dbxtool(1)`, for example), so you will notice the change in many areas.



**Text Menu Layout**

The new text menu is organized into several pull-right sub-menus, with "industry-standard" names; for example, 'File,' 'Edit,' 'Display,' and 'Find.' All basic editing operations such as 'Cut' are available from the new menu. The various flavors of 'Save' are available from the 'File => Finishing Up' pull-right menu.

**New Names**

The editing operations have been renamed with more industry standard terminology.

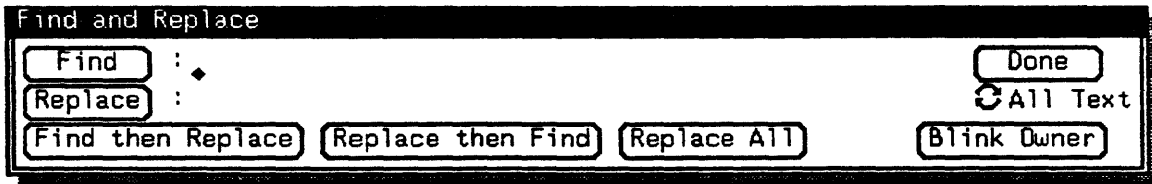
Table 1-1 *New Editing Terminology*

<i>New Name</i>	<i>Old Name</i>
Copy	Put
Paste	Get
Cut	Delete
Clipboard	Shelf
Copy then Paste	Put then Get

**Find and Replace Pop-Up Frame**

If you choose 'Find' or its first pull-right item, 'Find => Find and Replace', a small pop-up frame appears (shown below). You can type in the string to find, and an (optional) string to replace it. There are buttons for 'Find', 'Replace', 'Find then Replace', 'Replace then Find', 'Replace All', 'Done', and 'Blink Owner'; the latter helps to locate the owner of the frame, in case you have multiple Find and Replace frames on-screen.

default behavior.



**Delimiter Matching**

A new item in the 'Find' pull-right menu is 'Match delimiter'. If you select a delimiter in the text window, and then choose this menu item, the selection will be extended to the matching delimiter. For example, if you select the opening parenthesis in

```
Some text (stuff in parentheses).
```

and choose 'Find ⇒ Match delimiter' from the text menu, the entire text (stuff in parentheses) becomes the selection. You can also select the ending delimiter — ) in this example — and the selection will extend backwards to the starting delimiter. The matching skips over nested delimiters; for example, if you select the parenthesis before `setq` in the following LISP expression:

```
(let ((b 4)) (setq a (+ a b)))
```

and choose 'Find ⇒ Match delimiter', `(setq a (+ a b))` becomes the selection.

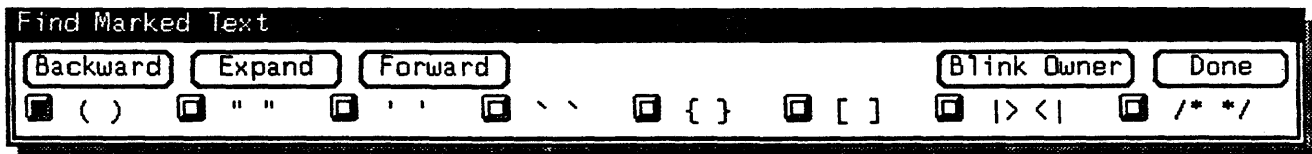
The text window understands the following delimiters:

Table 1-2 *Delimiters for Pattern Matching*

<i>Starting Delimiter</i>	<i>Ending Delimiter</i>	<i>Comments</i>
(	)	various styles of brackets
{	}	
[	]	
"	"	various quoting styles
'	'	
\	\	C language comments Note: C comments don't nest, so delimiter matching will not properly highlight a comment with a "nested" /* in it.
/*	*/	
>	<	field delimiters — see below

There is a new frame (shown below) that pops up when you choose 'Find

Marked Text' which lets you choose what delimiter to match, and the direction in which to search for it.



### Fields in Text windows

The `|>` and `<|` are special *field delimiters* in text windows. If you press **Control-Tab**, then the text between the next pair of field delimiters is selected and becomes pending-delete. Then, when you type something, the contents of the field are replaced. Thus if the text window displays the form

```
To: |>recipients<|
Subject:
Cc:
```

and you press **Control-Tab**, the `|>recipients<|` field will be highlighted, and as soon as you type anything, it will replace that field; you can “fill in the blanks” in the text window. This is, in fact, exactly what the new `mailtool` in Release 4.0 provides for (see *New mailtool* below).

You can also select a field from the menu, by placing the text caret between the delimiters and choosing ‘Find ⇒ Replace `|>field<|` ⇒ Expand’. This will make the field pending-delete. The other items in the ‘Find ⇒ Replace `|>field<|`’ pull-right menu, ‘Next’ and ‘Previous’, move the selection to the next or previous field.

### Word Wrap Mode

In previous versions of the text window, you could choose to have lines longer than the text window’s width clipped at the right edge or wrapped at a character. In Release 4.0, in the ‘Display ⇒ Change Line Wrap’ pull-right menu, there is a new option, ‘Wrap at Word’. If you select this, lines that you type will automatically be split at word boundaries as they get too long for the window, until you press **Return** to start a new paragraph. The split is purely visual — if the file is saved, the text up to **Return** is one long line, irrespective of how it was split into lines on the screen.

This is a useful option; however, much of UNIX is not equipped to deal with long lines. Most UNIX utilities, such as `grep(1)`, `wc(1)`, and even the C-shell and Bourne shell, have a maximum line length coded into them. If they are called upon to process lines longer than this, they will break. If you view files created in word-wrap mode on a terminal the long lines may just disappear off the end of the screen; if you copy them into a `shelltool` or `cmdtool`, they may exceed the shell’s input buffer size (256 characters), and may cause an alert to appear that warns you to flush the input buffer. The problem is especially acute with `mailtool(1)`, since the recipient may not be able to read the message you send on his or her display.

**NOTE** *In 4.0 BETA, a special filter program for formatting mail messages will be provided.*

If you use word wrap and you need to send the file to a UNIX utility or another user, you can use the `fmt(1)` command from the 'Extras' text menu (see 'Extras' Menu below), or from a function key assigned in your `~/textswrc` file; see the *SunView Beginner's Guide* for more information on using such a filter.

## 'Extras' Menu

Often you want to perform some operation on the contents of the selection in a text window and have its results appear in the text window; for example, you might want to capitalize a sequence of words, or insert quotation marks around a sentence. Before Release 4.0, you could do this by assigning the operation to a function key in your `.textswrc` file. In Release 4.0, this is still supported, but you can also operate on the selection from a new 'Extras' pull-right menu. The default Extras menu in `/usr/lib/.text_extras_menu` includes filters to 'Format', 'Capitalize', 'Shift Lines', 'Insert Brackets', 'Remove Brackets', and 'Pretty-print C'. These work the same way as FILTER keys in your `.textswrc` file: just select pending-delete the text to act on and choose the desired filter; the text will be replaced by the output of the filter.

You are encouraged to create your own 'Extras' menu just as you can create your own SunView "root" menu. You do this by creating a `~/text_extras_menu` of your own, using the same syntax as for the `~/rootmenu` file; then change *Extras\_menu\_filename* in the Text category of `defaultsed` to point to this file. The text extras file is re-read every time you bring up the text menu, so once you have created your own you can change it while running tools and the changes will take show up immediately in the 'Extras' pull-right menu.

## Keyboard Control of the Caret and Editing

"shift means reverse, control means motion, and meta means menu"

You can move the caret around the text window using the keyboard. In general, **Control**-key sequences move the caret, and **Meta**-key sequences are keyboard equivalents of menu actions such as editing, finding, etc. On the Sun-3 keyboard you can use the **Left** or **Right** key as the **Meta** key. On the new Sun keyboard, the **Meta** keys are marked with diamonds **◆**. In general, holding down the **Shift** key reverses the direction of an action. The following motions and actions are available:

Table 1-3 *Keyboard Motions and Accelerators*

<i>Action type</i>	<i>Key sequence</i>
<i>Erase actions (in 3.x also)</i>	
Erase_Char_Backward	<b>Delete</b>
Erase_Char_Forward	<b>Shift-Delete</b>
Erase_Word_Backward	<b>Control-W</b>
Erase_Word_Forward	<b>Shift-Control-W</b>
Erase_Line_Backward	<b>Control-U</b>
Erase_Line_End	<b>Shift-Control-U</b>
<i>Ways of moving the caret from the keyboard.</i>	



Table 1-3 Keyboard Motions and Accelerators—Continued

<i>Action type</i>	<i>Key sequence</i>
Go_Char_Backward	(Control-B) or (Shift-Control-F) or (R10)
Go_Char_Forward	(Control-F) or (Shift-Control-B) or (R12)
Go_Word_Backward	(Control-comma) or (Shift-Control-period) or (Shift-Control-slash)
Go_Word_End	(Control-period)
Go_Word_Forward	(Control-slash) or (Shift-Control-comma)
Go_Line_Backward	(Control-A) or (Shift-Control-E)
Go_Line_End	(Control-E) or (Shift-Control-A)
Go_Line_Forward	(Control-semicolon) or (R11)
Go_Column_Backward	(Control-P) or (Shift-Control-N) or (R18)
Go_Column_Forward	(Control-N) or (Shift-Control-P) or (R14)
Go_Document_Start	(Shift-Control-Return) or (R7)
Go_Document_End	(Control-Return) or (R13)
<i>Menu and function-key equivalents for various actions</i>	
Stop	(L1)
Again	(L2) or (Meta-A)
Props	(L3)
Undo	(L4) or (Meta-U)
Front	(L5)
Back	(Shift-L5)
Open	(L7)
Close	(Shift-L7)
Copy	(L6) or (Meta-C)

Table 1-3 Keyboard Motions and Accelerators—Continued

<i>Action type</i>	<i>Key sequence</i>
Go_Char_Backward	<b>(Control-B)</b> or <b>(Shift-Control-F)</b> or <b>(R10)</b>
Go_Char_Forward	<b>(Control-F)</b> or <b>(Shift-Control-B)</b> or <b>(R12)</b>
Go_Word_Backward	<b>(Control-comma)</b> or <b>(Shift-Control-period)</b> or <b>(Shift-Control-slash)</b>
Go_Word_End	<b>(Control-period)</b>
Go_Word_Forward	<b>(Control-slash)</b> or <b>(Shift-Control-comma)</b>
Go_Line_Backward	<b>(Control-A)</b> or <b>(Shift-Control-E)</b>
Go_Line_End	<b>(Control-E)</b> or <b>(Shift-Control-A)</b>
Go_Line_Forward	<b>(Control-semicolon)</b> or <b>(R11)</b>
Go_Column_Backward	<b>(Control-P)</b> or <b>(Shift-Control-N)</b> or <b>(R18)</b>
Go_Column_Forward	<b>(Control-N)</b> or <b>(Shift-Control-P)</b> or <b>(R14)</b>
Go_Document_Start	<b>(Shift-Control-Return)</b> or <b>(R7)</b>
Go_Document_End	<b>(Control-Return)</b> or <b>(R13)</b>
<i>Menu and function-key equivalents for various actions</i>	
Stop	<b>(L1)</b>
Again	<b>(L2)</b> or <b>(Meta-A)</b>
Props	<b>(L3)</b>
Undo	<b>(L4)</b> or <b>(Meta-U)</b>
Front	<b>(L5)</b>
Back	<b>(Shift-L5)</b>
Open	<b>(L7)</b>
Close	<b>(Shift-L7)</b>
Copy	<b>(L6)</b> or <b>(Meta-C)</b>

Table 1-3 Keyboard Motions and Accelerators—Continued

Action type	Key sequence
<i>Editing actions that have been remapped from their 3X keys.</i>	
Paste	<b>[L8]</b> or <b>[Meta-V]</b>
Cut	<b>[L10]</b> or <b>[Meta-X]</b>
Find_Backward	<b>[Shift-L9]</b> or <b>[Shift-Meta-F]</b>
Find_Forward	<b>[L9]</b> or <b>[Meta-F]</b>
Copy_Then_Paste	<b>[Meta-P]</b>
<i>Additional keyboard accelerators for actions.</i>	
Find_&_Replace	<b>[Control-L9]</b>
Select_Field_Backward	<b>[Shift-Control-Tab]</b>
Select_Field_Forward	<b>[Control-Tab]</b>
Store	<b>[Meta-S]</b>
Load	<b>[Meta-L]</b>
Do_It	<b>[Meta-Return]</b>
Help	<b>[Meta-?]</b> ( <b>[Meta-Shift-/]</b> )
Get_Filename	<b>[Escape]</b> , <b>[Shift-Escape]</b>

**NOTE** For the Get\_Filename action, the **[Escape]** key selects the first line in a text window, and uses it as the name of a file to load in. The key sequence **[Shift-Escape]** selects the current line (that the caret is on), and uses it as the name of the file to include.

For more sophisticated editing tasks, you can create sed scripts and add them to your 'Extras' text menu (see 'Extras' Menu above), or map them to function keys in your ~/.textswrc file.

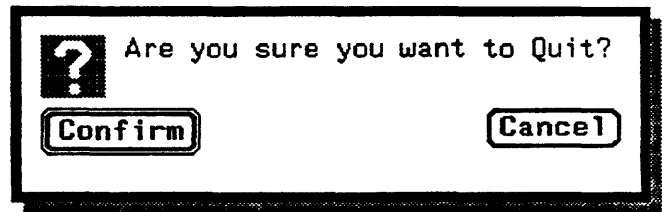
Keyboard control of the caret is integrated with the text window's **[Again]** key processing. This allows you to repeat fairly complex editing sequences. For example, say you want to get rid of the "> " inserted by diff(1) in front of a group of lines:

```
> New text new text new text.
> More new text more new text.
> Yet more new text.
> etc.
```

If you select the start of the first line and press **[Shift-Delete]**, **[Shift-Delete]**, then **[Control-N]**, this deletes the first "> " and moves down a line. Then you can just press **[Again]** repeatedly to alter the other lines.

## Alerts

Various errors, warnings and queries now appear in pop-up alert windows. An example of an alert is given below. In the alert, you push a button to 'Confirm', 'Cancel', 'Continue', etc. Alerts are intended to be self-explanatory; they replace the old (undocumented!) menu\_prompt() full-screen error messages.



Many alerts have a default button which is indicated by a double outline (as in the 'Confirm' button above). If an alert has a default button, then the pointer will jump to the button when the alert appears, so that a quick left click will take the default action. The pointer is moved back to its original position, when the alert goes away. You can disable *pointer jumping* by setting *SunView/IgnoreOptionalAlerts* to disabled.

#### Keyboard Accelerators for Alerts

If an alert has a default button, you can type **Return** to take the default action, even if the pointer is not in the alert. Also, many alerts have a 'Cancel' button. You can usually cancel an action by typing the **Stop** key (usually **L1**).

You can control the number of beeps when alerts are displayed by setting the *SunView/Alert\_Bell* default to your preference.

#### Shadowed Frames and Menus

The shadow under an alert which indicates that it is a transient window also appears under pop-up frames (frames with 'Done' in their menu instead of 'Close').

All shadows are now opaque.

#### File Size Limit on Editing Logs

In 3.x SunView, the temporary edit log for text windows (which records the changes you make to the file you are editing) was limited only by the space available in the filesystem in which your /tmp directory resides. This meant that if you were short of space in /tmp and you typed enough into `textedit`, or a lot of output appeared in `cmdtool`, you would run out of space.

In Release 4.0, you can set a bound on the size of the edit log in `cmdtool` by setting the defaults entry *Text\_wraparound\_size* in the *Tty* category of `defaultsed`. You can also set it using the command line option for `cmdtool -M maximum/minimum size`. (The equivalent text window attribute that client programs can set is `TEXTSW_EDIT_LOG_WRAPAROUND_SIZE`.) A value of 0 means no limit on the size of the edit log, other than space available in the filesystem.

If this limit is set and the contents of the text window grow by more than this amount without being saved to a file, the edit log "wraps around", re-using the beginning of the edit log, so that the log doesn't grow beyond the specified size. If you have simply been typing text into the text window, the effect is that characters are lost off the top of the edit log. If you scroll to an area where characters have been lost, you will see the words:

\*\*\* Text is lost because the maximum edit log size has been exceeded. \*\*\*

Since some text has been lost and the edit log is full at this point, you will encounter strange effects if you try to select or edit near points in the file where this message appears. Also, you can't 'Undo' editing operations in the part of the edit log that has been overwritten, nor can you repeat those actions with 'Again.'

*NOTE In practice, these special cases should not be a problem. Fixed-size edit logs are intended for use in situations where you want to display lots of text in a window and not be concerned about the edit log filling up your disk.*

#### Buffer Size Limit when Editing Files

If you are not editing a named file in a text window (`textedit` indicates this by displaying `(NONE)` as the file name in its frame header) then the edit log is kept in memory, not on the disk. You can set an upper limit on how large the edit buffer can be in this situation by setting `Text/Memory_Maximum` to some value in `defaultsedit`. The default is 20000 characters, the minimum is 1000, the maximum is 1000000. A value of 0 means to grow the buffer whenever it overflows.

If you do overflow the buffer size when editing in memory, the text window displays an alert warning you about this. You should respond to the alert by storing the file you are editing to some filename.

`mailtool` has a separate default for the buffer size for its text windows, `Mail/memorymaximum`.

#### New mailtool

Release 4.0 has a new version of `mailtool` with many enhancements. In order to provide backwards capability, you may change back to the previous version of `mailtool` by turning off the options in the `Mail` category of `defaultsedit` and restarting `mailtool`.

#### Simplified Control Panel

The layout of the control panel in `mailtool` has been simplified. However, if you set `panelstyle` in the `Mail` category of `defaultsedit` to `Old`, then `mailtool` will display much the same control panel as before, with some new buttons.

#### Multiple Composition Windows

You can compose a second message while in the middle of writing another. Each window will have its own reply control panel for delivering, or cancelling the message.

#### Separate Mail Composition Window

If you compose more than one message at once, or if you set `Mail/alwaysusepopup` to `Yes`, then `mailtool` will create a separate 'Reply to or Compose Mail' frame with a message composition window in which you can type messages. The 'Reply to or Compose Mail' frame has its own frame, so that you can 'Compose' a message even if `mailtool` is closed.

*NOTE mailtool may run out of file descriptors for its windows if you compose too many messages at once.*

## Reply Control Panel

Whether you compose messages in a separate frame or not, you will notice that when you 'Reply' or 'Compose' mail, a reply control panel appears above the composition window, with 'Include,' 'Deliver,' 'Clear,' and 'Re-address' buttons, and a 'Disappear'/'Stay Up'/'Close' cycle item. Note that the 'Deliver' button has a number of options on the menu behind the button.

'Include' lets you include any message while you are composing a message, not just when you select 'Reply.' You can choose the format in which the included message appears from a menu behind the button.

If your mail composition window is in a separate frame, you can use the 'Re-address' button to compose a message at any time, even if the main mail-tool frame is closed.

If you set the 'Reply to or Compose Mail' frame to 'Stay Up', then it will remain visible even after you 'Deliver' mail; if you set it to 'Close' then it will close to an icon after you deliver mail.

## Hierarchical Folders Menu

If you have a lot of mail folders, then the 'Folders' menu can be difficult to scan. In 4.0, mailtool allows you to have subdirectories in your folders directory. mailtool will convert any subdirectories in your folders directory into pull-right items in the folder menu. Moving the mouse to the right from these items will construct a pull-right menu of the folders in the chosen subdirectory, which can itself have subdirectories, and so on. To enable this feature, the defaults item *Mail/LISTER* is set to `ls -F` by default.

## Input Focus

Usually you want to type in the message composition window after you create it. If you use click-to-type mode (the default is that the keyboard focus follows the pointer), you may find it useful to set the default value of *Mail/moveinputfocus* in *defaultsedit* to *Yes*. This will cause mailtool to automatically place the caret in the message composition window whenever you click on 'Compose' or 'Reply,' regardless of whether or not the caret is currently in the tool. This allows you to click on 'Reply,' and then immediately start typing the body of the message. If you have fields enabled (see *Fields in Outgoing Messages* below), you can click on 'Compose,' and begin typing the recipients of the message without having to wait for the composition window to come up.

## Fields in Outgoing Messages

When you 'Compose' a message, a blank message header appears in the message composition window with the field next to *To*: highlighted.<sup>1</sup> You can then press **Control-Tab** to fill in the *Subject*: |body of message| fields; see *Fields in Text windows* above for more information on fields.

<sup>1</sup> You are only prompted for "Subject" and "Cc" if you set *Mail/asksub* and *askcc*, respectively, in *defaultsedit*.

**New Mail**

The new `mailtool` works with a new version of the `Mail(1)` program which can incorporate new mail into your mail file quickly, instead of committing your changes and then rereading the mail file. The default behavior you get when you push 'New Mail' is 'Incorporate New Mail'.

Keep in mind that you still need to commit your mail changes occasionally by choosing 'Done', 'Commit Changes and Retrieve New Mail', some variation on 'Commit', or quitting `mailtool`. Otherwise your mail file will still have all the messages you have already read in it; this may fill up your disk, and if `mailtool` should exit unexpectedly, all the mail you have read will show up again.

**Miscellaneous**

If the value of `Mail/editmessages` is *Unallowed*, the first time you attempt to edit a message in the message window, you will be asked to confirm the operation. (If you have set `Mail/expert` to *yes*, you won't see this at all.) The default value of `Mail/editmessages` is *Allowed* for backwards compatibility.

`mailtool` no longer changes the selection in the header window, thereby grabbing it from wherever else it was, when you do a `mailtool` operation. In particular, this means you can select a name in the header window, do a 'Find,' push 'Show,' then do a 'Find' again.

When you read new mail, or switch to a folder, `mailtool` will tell you the number of new, unread, and deleted messages.

Double-clicking on 'New Mail,' 'Done,' and 'Folder' (i.e., the time-consuming operations) will only cause one such invocation. Any additional events that occur before completion of the operation are ignored.

Specifying the font to be used in the tool via the `-font` or `-wt` command line argument now works correctly.

If you edit the header line of a message that you have received, the header window will be updated to show the new header when the message is saved, i.e., when you move to some other message.

**8-Bit Support in `shelltool` and `cmdtool`**

In conjunction with the new `tty` driver and Bourne shell in Release 4.0, `cmdtool` and the `tty` window package (used in `cmdtool` in 'Disable Scrolling' mode and in `shelltool`) support 8-bit characters, also known as extended character sets.

*NOTE* Most of the fonts in `/usr/lib/fonts/fixedwidthfonts` do not have characters defined above hexadecimal `0x7F`; you can use `fontedit(1)` to add another 128 glyphs to each.

**Underlining and Inverse in `shelltool`**

The `tty` window package used to use the 8th bit in its character memory to determine if a character was to be displayed in bold or not. The change to support 8-bit characters described above also allows the `tty` window code to support three graphic rendition modes:

Table 1-4 *Tty Display Modes*

<i>Mode</i>	<i>Escape Sequence</i>	termcap <i>Name</i>	<i>Description</i>
standout	Esc [ 7m	so	This is the same mode that the tty window supported before. It displays by inverting characters if <i>Tty/Standout_Mode</i> is enabled in <i>defaultsedit</i> .
underline	Esc [ 4m	us	This displays by underlining characters, if <i>Tty/Underline_Mode</i> is enabled in <i>defaultsedit</i> .
bold (extra-bright)	Esc [ 1m	md	This is the mode whose visual representation is controlled by the <i>Tty/Bold_style</i> setting in <i>defaultsedit</i> .
all other graphic rendition display modes	Esc [ nm	—	These display the same as bold(extra-bright)

The two new capabilities have been added to the *sun* entry in *termcap(5)*.

When there was only one graphic rendition mode, the tty window displayed everything in that mode — any kind of character highlighting would show up in your chosen *Bold\_style* (default inverse video). Now that there are three different modes, some things that used to display in your chosen bold style will now display inverted or underlined. In fact, “bold (extra-bright)” mode is rarely used, and this is the mode that you can change to many different styles by setting *Tty/Bold\_style* in *defaultsedit*.

You can get the old behavior by setting *Tty/Inverse\_mode* and *Tty/Underline\_mode* to *Same\_as\_bold* in *defaultsedit*. Also, if you need further control over what gets displayed in the different modes, you can modify *termcap(5)*.

## Frame Menu Changes

### 'Props' Item in the Menu

There is a 'Props' item in the frame menu, corresponding to the **[Props]** key (by default **[L3]**) on the keyboard. In applications that have a property sheet (for example, the optional *canvas\_demo* program) the 'Props' menu item will make the property sheet appear; in other applications, 'Props' will be grayed out. Unbundled applications and future tools use property sheets.



Other Name Changes	'Hide' is renamed 'Back' and 'Expose' is renamed 'Front.'
Unconstrained Move and Resize	In the frame menu, the default (top-most menu item in the pull-right menu) for 'Move' and 'Resize' is now 'Unconstrained.'
Other SunView Changes	These changes are of interest to current SunView users.
Files Renamed	The program you run to start SunView has been renamed <code>sunview</code> (although typing <code>suntools</code> still works). Similarly, the file in which you store your desired start-up tool positions has been renamed <code>.sunview</code> , but SunView will look for a <code>.suntools</code> file when it starts up, if there is no <code>.sunview</code> file.
New SunView "Root" Menu	The default "root" menu which you see when you bring up a menu over the background pattern has been changed. The old menu is still available on <code>/usr/lib/.rootmenu.old</code> . You can create your own SunView menu file, <code>~/ .rootmenu</code> , as before.
<b>SunView Changes Visible to the Programmer</b>	
Alerts	The new alert package used throughout the new SunView tools is documented in the 4.0 version of the <i>SunView Programmer's Guide</i> .
More File Descriptors	The 4.0 kernel supports more than twice as many file descriptors per process, so applications are less likely to run out of windows.
Lines in Menus	You can put lines in menus in Release 4.0 using the <code>MENU_LINE_AFTER_ITEM</code> attribute. This takes a value of either <code>MENU_HORIZONTAL_LINE</code> or <code>MENU_VERTICAL_LINE</code> . If you create an item with the <code>MENU_LINE_AFTER_ITEM</code> attribute, there will be a line between it and the next menu item; if you create a menu with <code>MENU_LINE_AFTER_ITEM</code> , then the entire menu has vertical or horizontal lines after items.
Props Attribute	You can use the new <code>FRAME_PROPS_ACTION_PROC</code> to specify a function to be called when the user chooses the 'Props' frame menu item, or hits the <code>Props</code> key.
Shadowed Frames	A new boolean frame attribute, <code>FRAME_SHADOW</code> , controls whether frames have shadows or not. You set this attribute at the time of creating the frame; thus it can be used in <code>window_create()</code> and <code>window_get()</code> , but not in <code>window_set()</code> .
"transient objects have shadows"	All subframes (frames owned by another frame, with 'Done' in their menu) have shadows by default.

**SunView Incompatibilities**

We think the new SunView features such as the new text menu are a dramatic improvement over their Release 3.X counterparts. However, many customers are affected by *any* change in the SunView user interface, usually because they have screendumps and instructions in documentation that assume the old SunView “look.” If you are such a customer, this section lists all the changed areas and the steps you can take to ensure visual fidelity with the past.

Table 1-5 *SunView User Interface Changes*

<i>Change</i>	<i>default</i> <i>sedit</i> <i>Work-Around</i>
Walking menus are the default	Set <i>SunView/Walking_Menu</i> to <i>Disabled</i>
New text menu	Set <i>Compatibility/New_Text_Menu</i> to <i>Disabled</i>
New frame menu	Set <i>Compatibility/New_Frame_Menu</i> to <i>Disabled</i>
Alerts replace “menu prompt”	Set <i>Compatibility/Alerts</i> to <i>Disabled</i>
New keyboard accelerators	Set <i>Compatibility/New_keyboard_accelerators</i> to <i>Disabled</i>
New root menu	Set <i>SunView/Rootmenu_filename</i> to <i>/usr/lib/rootmenu.old</i>
Many new mailtool features	Set <i>Compatibility/New_Mailtool_features</i> to <i>Disabled</i>
New tty menu	Set <i>Compatibility/New_Tty_Menu</i> to <i>Disabled</i>
Standout and Underline Modes	Set <i>Tty/Standout_Mode</i> to <i>Same_as_bold</i> Set <i>Tty/Underline_Mode</i> to <i>Same_as_bold</i>

## 1.10. Firmware and Diagnostics Changes and Upgrades

### 1.11. PROM Changes for Sun-4 Architecture

Due to Sun-4's new, RISC based architecture, the Boot PROM-based power-up self-tests are slightly different, as shown in the *Installation Notes for the Sun 4200 Board Set*, and in the *PROM User's Manual*. These differences show up only on the CPU Board LED display and on a dumb terminal attached to Serial Port A during a diagnostic boot-up.

Some PROM monitor commands were introduced to support the Sun-3/200 series during UNIX Release 3.2, and are also used to support Sun-4/xxx workstations. Commands such as `i`, `j` and `n` supported cache memory on Sun-3/2xx workstations and will now support Sun-4 cache memory.

#### Deleted Commands

The `a` and `t` commands are not present in the Sun-4 PROM monitor.

#### Changed Commands

The `d` command now dumps the state of the processor instead of opening a CPU data register.

The `h` (help) command now provides a more extensive user interface, described in the *PROM User's Manual*.

The `r` command, which previously displayed MC68020 registers, now displays SF9010 processor registers. Optional arguments are available for displaying floating point, global or special registers. You may also specify a register number to display a particular register. These registers may be observed after an unexpected trap or after a program or the user has aborted into the monitor.

The `s` command now sets Address Space Identifiers for the SF9010 processor, rather than Function Codes for the MC68020.

The `x` command provides, through a centralized diagnostic interpreter, a new user interface to the same extended tests that appeared in Sun-3 firmware.. This Extended Test System provides more comprehensive tests than the power-up self-tests, yet resides in the Boot PROMs. Rather than stepping through a menu hierarchy, you may now enter multiple commands from any menu to select tests and set test options. The *PROM User's Manual* describes the new command line options.

All other PROM monitor commands remain the same for this release.