

Credits and Trademarks

Sun Workstation® is a registered trademark of Sun Microsystems, Inc.

SunStation®, Sun Microsystems®, SunCore®, SunWindows®, DVMA®, and the combination of Sun with a numeric suffix are trademarks of Sun Microsystems, Inc.

UNIX, UNIX/32V, UNIX System III, and UNIX System V are trademarks of AT&T Bell Laboratories.

Intel® and Multibus® are registered trademarks of Intel Corporation.

DEC®, PDP®, VT®, and VAX® are registered trademarks of Digital Equipment Corporation.

Copyright © 1986 by Sun Microsystems.

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

Memory Requirements	3
1.1. System Software Changes and Upgrades	4
Filesystem Rearrangement	4
Support for Sun3 Stand-alone System of Specific Configuration	5
Support for Sun3/Sun2 Heterogeneous Client Support	5
New System Page Size	5
New a.out Format	5
New ptrace Interface	6
New Core File Format	6
New Disk Quota System	6
Sun3 Filesystems; newfs	6
NETWORK FILE SYSTEM Bug Fixes and Improvements	6
Network Yellow Pages Improvements	7
Improved Mail Features	8
1.2. Programming Environment Changes and Upgrades	8
Enhancements to Existing Facilities	9
New Packages	9
1.3. Windows and Window-Based Tools Changes	10
Must-Knows	10
Improvements	12
New Features and Applications	12
User Interface Bug Fixes	16
1.4. Programming Languages Changes and Upgrades	16
Compatibility Between Sun2 and Sun3	16

Upgrades to the Assembler	16
Changes to adb	19
Floating-Point Changes	20
Enhancements to dbx and dbxtool	22
Upgrades to FORTRAN	23
1.5. Graphics Upgrades	23
68020 Memory Pixrects	23
Graphics Processor Support (also included in release 2.1)	23
1.6. Firmware and Diagnostics Changes and Upgrades	23
1.7. Documentation Changes and Upgrades	25
Changes to -ms and -me Macro Packages	27
Font Changes	27
1.8. New Setup Interfaces	27
1.9. Support for New Peripherals	27
Driver Support for QIC-11 and QIC-24 Cartridge Tape Formats	27
1.10. New Games	28

Change Notes for Release 3.0

Change Notes for Release 3.0	3
1.1. System Software Changes and Upgrades	4
1.2. Programming Environment Changes and Upgrades	8
1.3. Windows and Window-Based Tools Changes	10
1.4. Programming Languages Changes and Upgrades	16
1.5. Graphics Upgrades	23
1.6. Firmware and Diagnostics Changes and Upgrades	23
1.7. Documentation Changes and Upgrades	25
1.8. New Setup Interfaces	27
1.9. Support for New Peripherals	27
1.10. New Games	28

Change Notes for Release 3.0

This document contains a description of changes to the system between Sun System Release 2.1 (May 1985) and Sun System Release 3.0 (January 1986). The highlights are:

- Across the board support for the MC68020, and for the hardware architecture of the new Sun3 line of machines.
- An upgraded window system, and with it a new family of window-based tools based upon the new mouse/window oriented `textedit` editor.
- An new FORTRAN optimizer that produces very fast code.

There are, in addition, numerous new features, bug fixes and rationalizations. These are described, in varying (and hopefully, appropriate) detail, below.

Memory Requirements

The 3.0 release is larger than the 2.0 release, and some 3.0 objects are larger than their corresponding 2.0 objects. Thus, you'll need more memory to run the 3.0 release effectively than was necessary with the 2.0 release — because of swapping overhead. Four Megabytes is ideal.

Furthermore, the 3.0 release occupies more disk space than did release 2.0. You can get an idea of its size by examining the following tables. They contain information on both Sun2 and Sun3 machines with 71 Megabyte formatted disks. No optional software — not even `suntools` — has been installed. Default values were used for most everything (swap size, disk labels, etc.).

Table 1-1 *Sun2 (Model 50; Micropolis Disk)*

Filesystem	Kbytes	Used	Available	Capacity	Mounted On
/dev/sd0a	7427	5240	1444	78%	/
/dev/sd0g	41771	18348	19245	49%	/usr
Total	49198	23588	20689		

Table 1-2 *Sun3 (Model 75; Fuji Disk)*

Filesystem	Kbytes	Used	Available	Capacity	Mounted On
/dev/sd0a	7402	5054	1607	76%	/
/dev/sd0g	42635	17936	20435	47%	/usr
Total	50037	22990	22042		

1.1. System Software Changes and Upgrades

Filesystem Rearrangement

/usr2 no longer exists in the 3.0 release. The home directory `xxx` will now be found in `/usr/server_name/xxx` rather than `/usr2/xxx`.

In order to conserve space on the root filesystem, a number of configuration changes have been made. In particular, a few programs formerly in `/etc` have been moved to, or symbolically linked to, `/usr/etc`: `/etc/gpconfig` has been moved to `/usr/etc/config`. Its microcode files were moved from `/etc` to `/usr/lib`.

- `skyr` has been moved from `/etc` to `/usr/etc`. Its microcode files were moved from `/etc` to `/usr/lib`.
- `showmount` has been moved from `/etc` to `/usr/etc`.
- `tunefs` has been moved from `/etc` to `/usr/etc`.
- `rwall` has been moved from `/etc` to `/usr/etc`.
- `/etc/config` is now a symbolic link to `/usr/etc/config`.
- `/etc/yp` is now a symbolic link to `/usr/etc/yp`.

There have also been a few important changes to the configuration of the standalone programs. In particular, the bulk of the contents of the `/stand` directory have been moved to `/usr/stand` on standalone machines (machines with disks) and to `/pub/stand` on NFS servers. This impacts the way in which most standalone programs are booted — the only standalone programs left in `/stand` are `diag` and `copy`.

For example, instead of booting

- `bboot_device (0, 0, 0) stand/mem`

the user will have to boot

- `bboot_device (0, 0, 6) stand/mem`

if the machine is a standalone machine with `/usr` in the 'g' partition (the normal situation), or

- `bboot_device (0, 0, 4) stand/mem`

if the machine is a Network File Server with `/pub` in the 'e' partition (a MC68010 CPU), or

□ `bboot_device(0, 0, 5) stand/mem`

if the machine is a Network File Server with a MC68020 CPU (in which case `/pub` will be on the 'f' partition.) Remember that 'dd' is here read as the name of the disk device.

Support for Sun3 Stand-alone System of Specific Configuration

Sun3 hardware support required changes in the kernel config file and the device driver interface.

- In the process of changing the config file, its syntax was cleaned up and made more understandable. See *Configuring the System Kernel* in the *Installing Unix* manual for details.
- New device drivers have been added (see `sd(4S)`, `st(4S)`, and `le(4S)`). The device driver interface was changed slightly to support the 32-bit VMEbus.

Support for Sun3/Sun2 Heterogeneous Client Support

A server may be either a Sun2 or a Sun3 machine and may support any mix of Sun2 and Sun3 clients. To support clients of a type different from the server requires additional disk space for the programs for the other machine type. See *Installing Unix on the Sun Workstation* for details of how to use the setup program to install such a server.

New System Page Size

The Sun3 release has a system page size of 8 Kilobytes (as opposed to 2 Kilobytes for Sun2 machines.) Note that this is the system page size and may well differ from the page size in the underlying machines.

New a.out Format

To support both Sun2's and Sun3's with a common a.out format, the a.out format was changed to include a machine type field. It indicates whether a program was compiled for a MC6810 or for a MC68020. (A value of 0 is taken to mean that the program was compiled for a version of UNIX† predating release 3.0).

Also, program text is now relocated to 8K (instead of 32K) and the data segment starts at the next 128K (instead of 32K) boundary. This is true on both Sun2's and Sun3's running release 3.0. Both Sun2's and Sun3's will understand the 2.0 release Sun2 a.out format when attempting to execute a program (see `sun3cvt(1)`). However, it may not always be meaningful to execute a 2.0 release program. For instance, if a program was compiled on 2.0 with the `-fsky` switch to use in-line Sky floating-point code, it will not be able to execute on a Sun3 machine, since Sun3's don't support Sky boards. Also, a program compiled on 2.0 will not take advantage of any of the hardware floating point units on a Sun3.

† UNIX is a trademark of AT&T Bell Laboratories.

Programs that depend on internal kernel data structures may not work on both Sun2's and Sun3's. Note — when executing old format `a.out`'s, the text space for these programs is NOT shared and they will NOT be demand loaded from the filesystem (instead they will be treated like non-shared (407) `a.out`'s).

Programs that depend on the format of `a.out` files must be recompiled in order to run on release 3.0.

New `ptrace` Interface

The `ptrace` interface has been extended so that debuggers no longer need to know details of internal kernel data structures (see `ptrace(2)`).

New Core File Format

The core file format has been changed so that it no longer contains internal kernel data structures (the "u" area) but instead contains structures that are now the defined interface between the kernel and debuggers, allowing the kernel to change without changing the debuggers. See `core(5)` for details.

New Disk Quota System

Release 3.0 contains a disk-quota system based upon the 4.2BSD quota system. The main difference is that the concept of login has been thrown out, along with much of the old quota code — counting logins doesn't work well in a network environment or with the NFS. The new quota system limits the amount of time a user can be over quota. The time limit can be adjusted on a per-file-system basis using `edquota(8)`.

Sun3 Filesystems; `newfs`

All Sun3 filesystems should be made with a block size of 8K. This is now the default for `newfs(8)`. Programs can not be demand loaded on Sun3's from filesystems that do not have an 8K block size.

NETWORK FILE SYSTEM Bug Fixes and Improvements

Bug Fixes and Speed-ups — All Completely Compatible with Previous Versions of NFS

`ls -lu` now works correctly for remote files.

`df` now does a better job when printing long mount-point path names.

When an NFS server is down, its name is given within the console notification message.

A client may mask-off `setuid` programs that come from mounted file systems. This feature is invoked via a new option to `mount`.

The `mount` command takes a large number of new options related to the mounting of NFS file systems.

The `mount` system calls have changed significantly. The `nfsmount(154)` and `mount(21)` systems calls are obsolete, and have been replaced by the new `mount(167)` system call. The `unmount` system call number has changed from 22 to 159.

`shutdown` and `rwall` work better:

- When a server is shutdown, the shutdown messages are sent asynchronously to all client machines in rapid succession.

- `rcp.rwalld` prints its message only to the console window.

The `find` command has a new option that prevents a `find` from looking at remote (NFS-mounted) files. Finds that search across NFS mounts *severely* impact network (and system) performance.

New Server Daemon

`/etc/rarpd` is a pair of user daemons which answer reverse address resolution protocol requests. They let a machine find its IP address when booting from the network. They should be run on any machines which runs `nd`. (See `rarpd(8c)`).

Sun Network Booting

The Sun3 PROMs use RARP and TFTP for the first part of the network bootstrap sequence. (In release 3.0, the machine actually TFTP boots a booter which uses `nd` to boot `vmunix`.) For a Sun3 to successfully boot from the ethernet:

- Its name and ethernet address must be in the server's `/etc/ethers` file.
- Its server must be running the `/etc/rarpd` daemon.
- Its server must have the Sun3's boot file in its `/tftpboot` directory. `setup` does this for you.
- Its kernel must be configured with the 'NIT' (Network Interface Tap) option enabled.

Network Yellow Pages Improvements

New Network Administration Programs — see Man Pages

- `sprayd(8)` -- see `spray.3r`
- `spray(8)` -- see `spray.3r`
- `rusers(1c)` -- see `rusers.3r`
- `rup(1c)` -- see `rstat.3r`
- `traffic(1)`
- `etherfind(8c)`

`traffic` and `etherfind` both use a new Yellow Pages daemon `maned rpc.ether` (See `etherd(3c)`). This daemon requires that the NIT facilities have been configured into the kernel.

New Yellow Pages Protocol

The 3.0 Yellow Pages are based upon an entirely new protocol, however 3.0 Yellow Pages servers and clients still support the old protocol, and are thus completely compatible with 2.0 machines.

Server Administration Interface

`yppush` is now synchronous — that is, it runs in the foreground —. It also reports to the user as machines update their Yellow Pages databases.

`yppull` is gone. It has been replaced by `ypxfer` which is also synchronous. It's recommended that slave servers use `crontab` to regularly update from master

servers via `ypxfer`. (See `cron(8)`).

`ypmatch` is a new user command (see `ypmatch(1)`). It's faster than `ypcat` | `grep`.

New Yellow Pages Maps

`ethers.byaddr` and `ethers.byname` map 48 bit ethernet addresses to host names and vice versa.

`netgroup.byuser` and `netgroup.byhost` are inverted forms of `netgroup.map`. Their existence makes it trivial to decide if a name is contained in a given `netgroup`.

The map `mail.aliases` can now be used for domain-wide mail aliases.

Performance Improvements

Due to protocol changes, `ypcat` and Yellow Pages database propagation are extremely fast.

Due to the new maps mentioned above, references to `netgroups` in `/etc/passwd` and `/etc/exports` now perform reasonably.

Improved Mail Features

In addition to the introduction of the new, window-based `mailtool` (described below), a number of changes have been made to the mail transportation system, `/usr/lib/sendmail`:

- Since the Yellow Pages map `mail.aliases` can now be used for domain-wide mail aliases, you should not have to remember hostnames when sending mail. `mail.aliases` will usually contain a copy of all the aliases known to a central mail machine. Refer to the *System Administration on the Sun Workstation* for details.
- Error messages have been improved, and several errors (like running out of swap space) that previously caused mail to be returned to the sender now just cause the delivery to be retried.
- Performance has been improved by consolidating processes. Frozen configuration files are now supported.
- The default configuration file will attempt direct delivery to any host in the `hosts.byname` map (or in `/etc/hosts` if the Yellow Pages are not enabled.) This eliminates the need for `/usr/lib/mailhosts`.

1.2. Programming Environment Changes and Upgrades

SunView is a new interface to the Sun window system. Actually, it's a pair of new interfaces — one for human users and one for software. The new software interface is based upon an abstraction of the relevant objects — windows, panels and the rest of it — and is guaranteed to remain stable even as the window system is upgraded. The SunView software interface is built upon the 2.0 release `suntool` interface, and that older interface will continue to work. However, the SunView level, where the 2.0 release tool structures are invisible, should be used for all future development. (See the *SunView Application Programmer's Guide* and the *SunView System Programmer's Guide* for more details.)

SunView is also the name of the new window environment, as it is experienced by human users.

Enhancements to Existing Facilities

There is a new, consistent, attribute-based client interface for the creation and manipulation of windows. It includes many enhancements — means for specifying subwindow layout, telling a window to consume or ignore events, get the state of the keyboard or mouse, etc.

A Virtual User Input Device (vuid) interface has been defined that is used by the kernel to interface with user input devices, e.g., the keyboard and the mouse. Devices that emulate the mouse driver, e.g., tablets, will no longer work with the new SunWindows kernel driver. To fix them, it will be necessary to modify the old device line discipline using the guidelines in the *Writing a Virtual User Input Device Driver* Appendix of the *SunView System Programmer's Guide*.

There is now a split keyboard/pick input focus (see the *Windows and Window-Based Tools Beginner's Guide* for details). To support it, there are now separate pick and keyboard masks, as well as routines to manipulate the keyboard focus, and new input events.

A number of enhancements have been made to Panels:

- They can be made scrollable by attaching scrollbars.
- Choice and toggle items can now have more than 32 choices.
- They support more flexible event handling.

The Option Subwindow package has been fully superceded by the Panel package, and has been dropped.

Icons now have an attribute-based interface.

Cursors now have an attribute-based interface and crosshair support.

Retained pixwins have been enhanced to support 'batching' for fast display updates. Overall, there is better retained pixwin support.

Better deadlock detection allows you to debug without setting lock override switches.

New Packages

There is now a Text Subwindow Package that supports various types of selections, multi-clicking capability, new-style walking menus, scroll bars, and the new window/mouse oriented editor `textedit`.

There is a Canvas Subwindow Package which supports drawing. Canvases can be retained or non-retained, and may have scrollbars built in.

The new Walking Menu Package provides an attribute-based interface, pull-aside submenus, etc.

There is a Scrollbar Package for viewing large objects in smaller windows. Scrollbars can be built into text, canvas and panel subwindows, and may also be used directly.

The new Notifier multiplexes sharing, within a user process, of process resources, e.g., signals, child process control, interval timers, the single thread of control. It enables multiple 'applications' to run within a single process, and supports the conversion of 'normal' straight-line-flow-of-control programs to the SunWindows environment.

There is a new Selection Package designed to allow applications to communicate concerning selections.

1.3. Windows and Window-Based Tools Changes

This section summarizes the changes made to the window system and the window-based tools for the 3.0 release. It explains three types of changes:

Must-Knows	Changes that may affect the current usage of the window system and its associated applications. Read this section before attempting to use the 3.0 window system.
Improvements	Changes that bring you up-to-date with the present state of the window and window application software.
New Features	New features that supplement the old window and window application software.

For more detail, see *Windows and Window-Based Tools: Beginner's Guide*.

Must-Knows

Definitely read this section so you can anticipate any difficulties that may occur in switching from the 2.0 release to the 3.0 release of the window system.

Further explanation of each of these items appears below:

- `suntools` attempts to assign new functions to certain keys, located on the far left or far right of your keyboard (when it has a problem doing so, you get the "ttypswrc cannot be mapped" error).
- You need to affix labels to your keys.
- When you use a Sun1 keyboard, you must run `setkeys`.
- When you want to use your mouse with your left hand, you must run `setkeys`.
- Four standard window applications have new names, as detailed below.
- Your windows may look different on the screen because the new default window font size affects the window attributes you set in your `.suntools` file.
- In fact, there is a new default `.suntools` file, so even if you don't customize your window attributes, you'll notice different windows appearing on your screen than before.

Sun2 and Sun3 Keyboards

By default, `suntools` assigns keys **[L1]** through **[L10]** on the Sun2 and Sun3 keyboards to operate with the new SunView text facility.

- Key Mapping Error** If you've already assigned some of these keys using a `.ttyswrc` or similar file, `sunttools` may produce a "ttyswrc cannot be mapped" error. Unassign these special function keys when you want to use the standard 3.0 release window system configuration.
- Text Facility and Key Labels** To find out about the text facility, read *Windows and Window-Based Tools: Beginner's Guide*. The labels are inserted within the back cover of that manual; instructions on how to affix the labels appear near the beginning of the chapter on the text facility.
- Sun1 Keyboard** If you have a Sun1 keyboard, associated with the 100U workstation, add:
- ```
setkeys Sun1
```
- to your `.login` file, so that `sunttools` assigns the special text facility keys to the right side of the keyboard. More information on `setkeys` appears in the chapter on the text facility.
- Left Hand Mouse Use** Having the keys on the right side is more convenient for people who want to use the mouse with their left hand. When you want `sunttools` to assign the special text facilities to the right side of the Sun2 or Sun3 keyboard, add:
- ```
setkeys lefty
```
- to your `.login` file. More information on `setkeys` appears in the chapter on the text facility.
- New Application Names** This chart shows the new names and the old names of the four window applications whose names have changed (although `edittool` wasn't distributed with the standard 2.0 release):
- | <i>3.0 Name</i> | <i>2.0 Name</i> |
|-----------------|-----------------|
| clock | clocktool |
| textedit | edittool |
| fontedit | fonttool |
| iconedit | icontool |
- New Default Font Size** The new default font is an 11 point font, one point size smaller than the 12 point font used until the 3.0 release. The window attributes you've set within your `.sunttools` and customized root menu files may not cause windows to appear on the screen the same as before.
- You can change your default window font size using the new `defaultsedit` application, or you can use the new `toolplaces` command (described in the **New Features** section below) to properly store new window attributes.

Improvements

Improvements to the existing window system and associated applications include:

- The rectangular cursor becomes a *shadow cursor* when you focus attention by using the cursor to choose another window.
- Text facility windows contain a triangular cursor. It turns grey when you focus attention with the cursor on another window.
- The default root menu has been altered to make more functions easily accessible.
- The name of the Tool Mgr menu to the Frame menu has been changed.
- The Stretch item on the Frame menu has been renamed Resize, to convey shrinking as well as stretching.
- The border of a window now darkens when you move the cursor into the window, for example, to type a command. When using the "click-to-type" attention focus described in the New Features section, the border darkens when you click on a window instead.
- There is now a split keyboard/pick input focus. It allows you can now set the focus for keyboard input to a specific window, and yet move the mouse freely over other windows to scroll, manipulate windows and make selections. It is enabled by the '-S' option to suntools.
- You can now mouse and type ahead reliably by giving the '-b' option to suntools to enable input synchronization. If you enable the split keyboard/pick input focus, you also get input synchronization.
- The tty subwindow has a new 'fast display' option, settable via `defaultsedit`
- There are new accelerators, which allow users to manipulate windows with a minimum number of mouse clicks.
- There is an improved icon layout policy.
- There is an improved user interface for stretching and moving windows and subwindows, including a fullscreen zoom.
- `lockscreen` has been enhanced, and can now run arbitrary programs. Two possibilities are `/usr/games/lifescreeen` and `/usr/demo/maze`.

New Features and Applications

- There is a variety of new accelerators, or window command abbreviations. They are described at length in the accelerator and quick reference guide sections of *Windows and Window-Based Tools: Beginner's Guide*.
- There is now a distinction between constrained and unconstrained window movement and resizing.
- It is now possible to move and resize subwindow borders within applications that contain them using the same accelerators as you use for moving and resizing window borders.

- There is a new zoom operation, which extends a window to full screen length.
- The `toolplaces` command has been enhanced so that its output can be read by `suntools`.
- There is a new text editor, `textedit`, which is built upon the Text Subwindow Package.
- `Dbxtool` has been enhanced with the Text Window Subpackage, and now supports source code editing.
- There is a new shelltool, `cmdtool`, which combines the shell interpreter capabilities of the old `shelltool` with the text editing capabilities of the new text facility. (`cmdtool` doesn't yet work well with programs which modify the `stty` state of their window or use control characters to interact with users).
- There is a new window-based program, `defaultsedit`, which centralizes customization information (for `mail`, `scrollbars`, `textedit`, etc) into a Defaults Database.
- There is a "Click-to-type," alternative to the "move-to-type" method of focusing the keyboard on a particular window.
- There is a new window-oriented mail program, `mailtool`.
- There is a new program, `click`, which allows the user to toggle the Sun3 keyboard clicking on/off.

toolplaces

The `toolplaces` command appeared in release 2.2. `toolplaces` generates position, size, label, and program attributes for the windows running on a window system screen at the time when you run the command. It is useful for creating a customized `.suntools` file, to reuse your favorite window attributes, and for customizing your root menu.

Selections

With `textedit` and `cmdtool`, you can select a portion of text using the left mouse button to start, the middle mouse button to adjust, just as you could with `shelltool`. However, cut-and-paste operates differently than the `shelltool`.

To copy text, you can select a portion of text, and *put* it on the *shelf*, using the **PUT** function key (see the *Windows and Window-Based Tools: Beginner's Guide* chapter on the text facility to identify the keys). Then, move the cursor to where you want to insert the selected text, and press the **GET** function key.

When you wish to delete the selection, then insert it elsewhere, simply select it, press the **DELETE** function key, move the insertion point, and press the **GET** function key.

To copy text from one `shelltool` window to another `shelltool` window, you can use the selection and stuff procedure as with 2.0.

Or you can use the new procedure for copying text between any two text facility tools. To copy text from one tool to another, select the text in the first tool, press the **[PUT]** function key, move the insertion point to the appropriate spot in the second tool, then press the **[GET]** function key to complete the copy operation.

Multi-Clicking

When you want to select a whole word, press the left mouse button twice in rapid succession. When you want to select a whole line, press the left mouse button three times in rapid succession.

Secondary Selections

You can make a temporary selection that lasts only as long as you hold down a function key. When you release the function key, the function operates on the secondary selection you made while holding the function key down. The secondary selection appears underlined. Useful with the **[FIND]** function key for searching.

Pending Delete Selections

You can make a selection that will be deleted by a subsequent insertion or copy by pressing the **[CTRL]** key while making the selection. A pending delete selection appears in gray.

New-Style Walking Menus

Walking menus are a new style of menu that allows you to “walk” off the right edge of the menu by moving the cursor so that a sub-menu appears. In this way, one walking menu can present several sub-menus for a variety of added functions.

Walking menu selections in `textedit` and `cmdtool` include:

- `Reset` — remove the current file from the window
- `Load named file` — select the name of the file you want to load before choosing this item, use the `Scratch area...` when you don't want to select within the main window. Later saves will save to this filename.
- `Save` — which includes a sub-menu for more specific menu item choices. **Remember to save your work regularly.**
- `Split this view` — allows you to look at the file in multiple windows, insertion point held consistent
- `Destroy this view` — the complement to `Split this view` for getting rid of multiple windows.
- `Line break` — to changes line wrapping attribute.
- `cd` — to connect to the directory of the name which you have selected.
- `Insertion to top` — repositions the window so that the line containing the caret is at the top of the window.
- `Show numbered line` — show the line with the line number you have selected

- Line number? — provide the initial line number for the current selection
- Stuff from file — insert contents of file with name you have selected. Later saves ignore this filename, see Load named file above.

Scroll Bars

`textedit` and `cmdtool` have *scroll bars* along the left side of the frame. The scroll bars control movement with a left click moving you down and a right click moving you up. A middle click moves you to the section of the file represented by the height at which you click along the scroll bar.

You can modify attributes of the scroll bars with `defaultsedit`.

defaultsedit

`defaultsedit` permits many changes to your defaults, for example getting rid of an audible bell. It provides help information when you click the left mouse button over any relevant screen portion of the application. The following categories of defaults are available:

- Defaults
- Indent
- Mail
- Scrollbar
- SunView
- Text

Focusing Attention: Click-to-type

Click-to-type is an alternative way of choosing windows to the typical move-cursor-and-type model. Instead of moving the cursor and typing, click-to-type means click while the cursor is in a particular window and the attention will remain focused on that window. In other words, everything you type will appear in that window.

You can start click-to-type using `defaultsedit`.

mailtool

To learn about `mailtool`, the new window and mouse mail program, see *Mail and Messages: Beginner's Guide*. Here are a couple of tips though:

- `mailtool` pops up help information when you press the right mouse button over any of the panel buttons. A panel button is a button that appears within a window tool on the screen.
- You may want to set `mailtool` to hold mail in your spool file (`/usr/spool/username`), rather than dumping it into your `~/mbox` directory. Do this by setting the `Set/hold` attribute within the `Mail` category in `defaultsedit` to the `yes` setting.

User Interface Bug Fixes

Lock timeouts are now computed in process virtual time. This means that windows shouldn't be killed when the machine is loaded.

Exiting suntools is now more reliable. It used to occasionally hang the keyboard, repeat keys or print a lot of annoying messages.

The 'Ghost' color images which preceded the painting of a window on a color display have been replaced by simple clearing.

The keyboard bell now works in `shelltool`.

1.4. Programming Languages Changes and Upgrades

Compatibility Between Sun2 and Sun3

Most programs that run on a Sun2 machine will also run on a Sun3 machine, unless the `fsky` option (in-line Sky floating-point code) is used. Programs compiled on a Sun3 machine use the extended instruction set of the Motorola MC68020 and will not run on Sun2 machines. (The MC68020 instruction set improves performance and reduces code size in program constructs like array subscripting, bounds checking, and integer division and multiplication.)

If at all possible, we strongly recommend that programs be recompiled. If that is not possible, we strongly recommend that they be relinked. Relinking (which may not always work because of library changes) will at least allow the text segment of the program to be shared and will allow the program to be demand loaded from the filesystem.

Cross compilation (in either direction) between Sun2 machines and Sun3 machines is not supported in the 3.0 release.

Most programs that run on 2.0 and earlier releases will also run without modification on machines running the 3.0 release.

Preliminary versions of release 3.0 included a utility program (`sun3cvt`) for converting old Sun2 programs into the new release 3.0 object file format. This program is no longer required.

Upgrades to the Assembler

First of all, a repair — `pc@` addressing has been 'fixed' and hopefully improved.

Register Syntax

The floating-point registers are `fp0`, `fp1`,...`fp7`; `fpc`, `fps`, `fpi`. Motorola calls the last three CONTROL, STATUS, and IADDR.

Operand Types

The known operand types are `b`, `w`, `l`, `s`, `d`, `x`, and `p`. `x` and `p` are not currently supported as constants or immediate data.

Floating Constant Declarations

```
.single      e1, e2, e3, ...  
.double     e1, e2, e3, ...
```

`e1`, `e2`, etc. can be either floating constants or integer expressions.

Floating Constant Syntax

Floating-point constants and immediates may be integer expressions or constants beginning with 0r or 0R. Thus to declare a single-precision constant '1', use any of the following:

```
one:    .single    1
one:    .single    0r1
one:    .single    0R1.0
```

To move '1' to fp0, use any of the following:

```
fmoveb    #1, fp0
fmoves    #0r1, fp0
fmoved    #0R1.0, fp0
fmoves    one, fp0
fmovecr   #0x32, fp0
```

Coprocessor Identification

The coprocessor id field of a single instruction is indicated like this:

```
fopT@C    operands
```

where C is an expression that evaluates to a coprocessor id in the range 0 to 7. If the @C is missing then the default is used. The default starts out as 1 but may be changed with the assembler directive

```
.fcpid    C
```

Instruction Mnemonics

In fopT, f is fixed, op is the op name, and T is the type letter. In general the mnemonic is the same as the Motorola mnemonic, but in lower case and without the dot. This similarity allows coding from the Motorola MC68881 manual with little reference to the Sun manual for opcode names.

Supported Standard Format Operations

Almost all the common operations are supported in the following syntaxes:

```
fopx    fpm, fpn
fopT    <ea>, fpn    T = b, w, l, s, d, x, p
```

The supported operations include abs, acos, add, asin, atan, atanh, cmp, cos, cosh, div, etox, etoxml, getexp, getman, int, intrz, log10, log2, logn, lognp1, mod, move (in), mul, neg, rem, scale, sglmul, sgldiv, sin, sinh, sqrt, sub, tan, tanh, tentox, twotox. The supported moves are those with an fpn destination.

Additional Supported Operations

fmoveT	fpm, <ea>	T = b, w, l, s, d, x
fmovecr	#<ccc>, fpn	
ftstT	<ea>	T = b, w, l, s, d, x, p
ftstx	fpm	
fmove1	<ea>, fpc	
fmove1	fpc, <ea>	
fmove1	<ea>, fps	
fmove1	fps, <ea>	
fmove1	<ea>, fpi	
fmove1	fpi, <ea>	
fmovem	#<imm>, <ea>	
fmovem	<ea>, #<imm>	
fmovem	<freglist>, <ea>	
fmovem	<ea>, <freglist>	
fmovem	<ea>, dn	
fmovem	dn, <ea>	
fmovem	<ea>, <fcreglist>	
fmovem	<fcreglist>, <ea>	
fnop		
fbcc	<label>	
fbcc1	<label>	
fjcc	<label>	
fdbcc	dn, <label>	
fsccl	<ea>	
fttrapcc		
fttrapccT	#<xxx>	T = w, l
frestore	<ea>	
fsave	<ea>	

Register Lists

movemT and fmovem instructions may use register lists instead of immediate constants. Register lists are based on Motorola syntax with ranges indicated by '-' and disjoint registers by ','. Register lists are:

movemT	a0-a7/d0-d7
fmovem	fp0-fp7
fmovem	fpc-fpi

The old form with an immediate constant is accepted for convenience, except that fmovem of control registers cannot be indicated with immediate constants.

New Integer Instruction Mnemonics

To reduce the number of confusing differences between the syntax specified in the MC68020 manual and the syntax accepted by Sun's assembler, the following new mnemonics have been added:

```

illegal
chkw   =      chk
rtd    =      rts
moveb  =      movb
movew  =      movw
movel  =      movl
movec  =      movc
movesb =      movsb
movesw =      movsw
movesl =      movsl

```

Software Floating Point Conditional Mnemonics

jfcc and sfcc mnemonics have been added for the following condition codes to simplify code generation from higher-level languages for software floating point. Note that proper floating-point condition codes do not correspond to integer condition codes, since the latter do not allow for unordered comparison.

jf	sf	implemented as:	
eq	ne	eq	ne
lt	nlt	cs	cc
le	nle	ls	hi
gt	ngt	gt	le
ge	nge	ge	lt

PC-Relative Addressing Modes

The assembler now supports addresses of the form `pc@(d)` where `d` is absolute, relocatable, external, or external +- absolute. Addresses of the form `pc@(d,m:w)` are supported where `d` is absolute or relocatable. The following declarations are supported:

```

.word   external+-absolute
.long   external+-absolute
.word   external-relocatable
.long   external-relocatable

```

Note that the current linker may generate word displacement overflow error messages for the `.word` form. Some of these are spurious; the output must be inspected for correctness.

Changes to adb

MC68881 instructions are disassembled. Unrecognized instructions are disassembled as `'word'` rather than as `'???'`.

Absolute and pc-relative displacement formatting has been changed to move toward the conflicting goals of correctness, legibility, and the ability to reassemble to the original object code. The results are inherently controversial.

The F (double) print format has been fixed to work better with registers. `<r=F` prints the double found at the core image address `r`. Thus `<d0=F` prints out the combination of `d0,d1` as a double, the way you probably thought it worked all

along. Actually, the way it used to work was bizarre by normal standards — the address from which the second half was fetched was the value in d0 plus four. You no longer get ‘address wraparound’ if d0 contains -1.

New Registers

The floating-point data registers are fp0, fp1, fp2, fp3, fp4, fp5, fp6, fp7. These are treated as 32-bit registers just like all other adb registers. The image of the MC68881’s FP0 is actually mapped into three 32-bit chunks whose names are fp0, fq0, and fr0. You can read these 32 bit registers in the usual way, e.g. <fq0=X.

Other 32-bit floating-point registers are fpc (control), fps (status), and fpi (address). Two other registers are defined from the reg.h mapping: fpg = fp_code, for signals, and fpf = fps_flags, the MC68881 state: 0 for unused, 1 for idle, 2 for busy.

New Output Format

‘e’ or ‘E’ can be used to print registers or memory as extended values. Thus the usual way to look at fp0 would be <fp0=e, which prints out a decimal string just as <d0=f or <d0=F does. Extended format can also be used to print out the contents of memory in 12-byte chunks. Extended format only works when a physical MC68881 is present, and the conversion is limited by the capabilities of the MC68881: the range of exponents is limited to +- 999, and significands are limited to 17 digits. If no MC68881 is present, extended format prints an error message.

New Registers Command

‘\$x’ or ‘\$X’ can be used to print out all the floating-point registers in hex, with the data registers printed in ASCII. If no MC68881 is present, an error message is printed.

Non-changes

It has never been possible to assign a floating-point string value in the fashion ‘3.14159>d0’. It still isn’t. However, floating-point constants can still be manipulated in hex chunks:

3f800000>d0	– assign a single-precision 1.0 to d0
<d0=f	– print it
3ff00000>d0	– assign a double-precision 1.0 to d0
0>d1	– and to d1
<d0=F	– print it

Floating-Point Changes

Release 3.0 floating-point software resides in /lib, libc.a, and libm.a. It has been reorganized to support floating-point code generation via software, the Sky board or a MC68881, and to allow the run-time selection between these options.

Compiler Switches

New options have been added to the cc, pc, and f77 compilers, to specify the type of code to be generated for floating-point arithmetic expressions. The

options are:

`-fsoft`

Software floating-point library calls. This is the default, intended for use in programs that make only incidental use of floating point and machines with no floating-point hardware.

`-fsky`

In-line code for Sky floating-point processor (Only supported on Sun2 machines.)

`-f68881`

In-line code for Motorola MC68881 floating-point processor (only supported on Sun3 machines.)

`-fswitch`

Run-time switched floating-point library calls (for floating-point-intensive programs that must run on any of the above hardware configurations.)

Floating-point code generation may also be specified by the environment variable `FLOAT_OPTION`. If none of the above options is specified, the compiler will check `FLOAT_OPTION`. Legal values of `FLOAT_OPTION` are 'fsoft', 'fsky', 'f68881', and 'fswitch'.

If a specific floating-point device is selected at compile time (`fsky` or `f68881`) the compiler will generate in-line code that calls that device. If `fsoft` is invoked, the compiler will call pure software implementations of the floating-point operations. If `fswitch` is invoked, the decision about how to implement floating-point operations will be deferred to run time. At run time, the first floating-point operation encountered will select a device (depending on the configuration of the machine.) Subsequent floating-point operations will use that same device.

Almost all floating-point routines in `libc.a` and almost all elementary transcendental functions in `libm.a` are sensitive to the floating-point options. Be consistent in declaring your floating-point choice! Give it on all compiler invocations, including the linking step — otherwise you'll get software square roots, etc.

Switching Between Devices at Run Time

If desired, code compiled with `fswitch` can be switched to run on specific devices at run time. The FORTRAN calls

```
call fswitchfp
call sswitchfp
call mswitchfp
```

can be used to force switched calls to be performed in software, with the Sky board, and with the MC68881, respectively. Code compiled with options other than `fswitch` is not affected by these calls. These calls are primarily for testing and may not be supported in future software releases.

Initialization

Programs are initialized by code in `/lib` that is loaded to satisfy the external 'start'. The compiler drivers instruct the linker to load file `/lib/crt0.o`,

`/lib/gcrt0.o`, or `/lib/mcrt0.o`, according to whether profiling is requested. Each of these modules requests an entry point `'start_float'`. If no floating-point device was specified, then `start_float` is satisfied by a no-op routine in `libc`. If a specific device *D* was specified, then the compiler driver requests an external `'Dused'` and instructs the linker to load file `/lib/Dcrt1.o`. `Dcrt1.o` defines `Dused` and `start_float`, so that only one kind of specific device can be requested in one program. At run time `start_float` initializes the specified device and terminates with an error message if the device is not there.

New Condition Codes

The results of floating-point comparisons have been altered to better meet the IEEE arithmetic standard. Now, if *x* or *y* is not a number (NaN), even if *x* and *y* are identical NaNs, then the following will be true:

`x .ne. y`

while the following will be false:

`x .eq. y`

`x .lt. y`

`x .le. y`

`x .gt. y`

`x .ge. y`

Enhancements to `dbx` and `dbxtool`

- The release 3.0 version of `dbxtool` uses the text sunwindow facility. This will allow editing of source code in a source window.
- Arbitrary process debugging is now supported via the `detach` command.
- Kernel debugging is now supported via the `proc` command.
- Programs consisting of multiple processes can now be debugged.
- Initialization is much faster.
- The MC68020 is fully supported.
- The MC68881 is supported by the addition of the register symbols `$fp0 – $fp7` and the `set81` command.
- Literal strings may now be passed as arguments to functions.
- Two new attributes, `case` and `speed`, have been added to the `dbxenv` command.
- A `rerun` command has been added.
- The `'?:'` operator is supported.
- The `'^^'` operator is supported as a bitwise-XOR, and no longer supported as post-fix indirection.
- The `clear` command with no arguments clears the current breakpoint.
- The `cont` command with a signal number has different syntax.

Upgrades to FORTRAN

Test Coverage for FORTRAN

There is a new flag, '-a', for f77, the FORTRAN compiler. It's invocation causes the compiler to insert code into the program which counts how many times each basic block is executed. The `tcov` utility can then be run on the source file to generate a statement-by-statement profile of the program.

New Optimizer

The '-O' flag of f77 now invokes `/usr/lib/irop`, a separate code transformation utility, which produces substantially better code than earlier releases of the compiler. When '-O' is specified, the following transformations are performed where appropriate:

1. Loop-Invariant Code Motion
2. Strength Reduction for Induction Variables
3. Common Subexpression Elimination
4. Dead Code Elimination
5. Constant Folding
6. Short-Circuit Evaluation of Boolean Expressions
7. Copy Propagation
8. Global Register Allocation

For very large procedures, the processing time required by the optimizer may exceed the time required for optimized compilation with the release 2.0 version of f77. A new f77 flag, '-P', requests partial optimization. When -P is set, transformations 1-4 in the above list will not be attempted and optimization time will diminish substantially.

1.5. Graphics Upgrades

68020 Memory Pixrects

Memory `pixrects` code has been rewritten to take better advantage of the 68020. No changes to the client interface were made.

Graphics Processor Support (also included in release 2.1)

Release 3.0 of course supports the Graphics Processor and Graphics Buffer options first available (for the Model 160 Color Workstation) with release 2.1.

1.6. Firmware and Diagnostics Changes and Upgrades

Release 3.0 firmware has changed in several major ways:

- The diagnostics have been split from the software releases and are being treated as unbundled programs. Further, to allow for the testing of the disks themselves (and to save disk space), the diagnostics have been made tape bootable. The option to install the diagnostics on the disk is still available (for use with a server or for performance reasons.)
- The Sun3 hardware contains an EEPROM which describes the configuration of the system and allows for the customization of a variety of system characteristics (e.g. the logo). It's important that the EEPROM be updated

whenever the system configuration is changed, (see monitor(8S)), for the diagnostics depend upon it in determining what to test. For more details on the new EEPROM, see the *Sun3/160 Preliminary Field Service Manual*.

- Tape Boot Support** The Sun3 PROMs support the Xylogics 472 controller as well as the Tape Master controller common in Sun2 systems.
- SCSI Tape Boot in PROMs** Under release 3.0, Sun2 and Sun3 both support the booting of the diagnostics directly from tape. (This will work for either the Sysgen or Emulex controllers).
- Support for Large Main Memories** During startup, the Sun3 PROM's will size and initialize memory up to 24 Megabytes (default is 4Megabytes.)
- Terminal Selection Made in EEPROMs** A Sun3 system will boot using as its terminal:
- A black/white terminal,
 - A color terminal,
 - Either TTYA or TTYB.
- The EEPROM controls which device is to be the primary terminal, and must therefore be properly configured prior to booting if more than one of these devices exists. (This can be done with the 'q' command to the monitor — see below).
- Diagnostic Switch Vectoring** A diagnostic switch, used to direct the Boot PROM during power up, has been installed on all Sun3 products. It has two positions:
- In the diagnostic position the PROM performs a self test. At the completion of the test it offers an extended test menu.
 - In the the boot position the PROM performs a short, modified version of the self test. At the completion of the test it searches for and attempts to boot UNIX. If unable to do so it exits to the monitor.
- Watchdog Reset** Release 3.0 PROMs contain a watchdog reset. It is automatically invoked by a double bus error, and can be manually activated by a user reset switch. In either case the monitor portion of the PROM takes over, and if so instructed by values stored in the EEPROMs, does a power reset boot.
- Sun3 Keyboard Support** The new PROMs support both Sun2 and Sun3 keyboards.
- New Sun3 PROM Commands** Five new commands have been added to the monitor:
- **f** = fill, allows for the filling of memory with a hexadecimal pattern.
 - **h** = help — lists PROM commands and their meanings.

- **q** = reads/writes the CPU board EEPROM, thus allowing its configuration parameters to be changed.
- **v** = view, allows for the viewing of memory
- **x** = extended test menu, provides an extended test menu that permits the testing of memory, video, keyboard, serial ports, and boot paths. The boot path testing includes all boot devices (SCSI, disc, tape) except ethernet, for which a loopback test is provided.

Object Code compatibility in Stand Alone Libraries

The stand-alone libraries have been converted so that both Sun2 and Sun3 run the same object code.

TFTP Booting Support in PROMs for Sun3

The only way to boot a Sun3 over the Ethernet is by using the TFTP protocol — ND is no longer supported in the PROM.

1.7. Documentation Changes and Upgrades

The New Look

As a part of the continuing process of rationalizing both the organization and content on the Sun manuals, they are now being formatted in a 'new look.' We are using a typewriter font like this throughout for examples of code, user input, and system response. What the user types at the system is shown in **bold typewriter font like this**.

Half Title Pages

In general, Sun Technical Manuals now contain what are known as 'half-title pages' — there is a mini table of contents at the start of each chapter.

Indexes

In general, Sun Technical Manuals now contain an index at the back. In particular, the two volumes of manual pages, *Commands Reference Manual* (sections 1, 6, 7, and 8) and the *System Interface Manual* (sections 2, 3, 4, and 5) now contain real indexes. The old UNIX style permuted index is defunct.

System Internals Manual Deleted

The *System Internals Manual* has disappeared entirely. The section *Using adb to Debug the UNIX Kernel* has been incorporated into a new document — *Debugging Tools for the Sun Workstation*. The section called the *CPU PROM Monitor* has been folded into *Writing Device Drivers for the Sun Workstation*. The section of the file system has been moved into *System Administration for the Sun Workstation*.

Programming Utilities for the Sun Workstation

There is a new chapter on *Tools for the C Programming Language*. The previous chapter on `lint` has been folded into this chapter.

The chapter on `make` has been updated to reflect the System V version of `make`.

There is a new chapter on *Performance Analysis Tools*.

The old part of the manual that dealt with the low level SCCS command interface has been moved to an appendix. The individual SCCS command

descriptions from the section (1) manual pages have also been moved to this manual.

All acknowledgements and references were moved to an appendix.

Writing Device Drivers for the Sun Workstation

Writing Device Drivers for the Sun Workstation has been updated to reflect Sun3 architecture. New material has been added to various sections to enhance the 'folklore' in this manual. The chapter on *General Hardware and Software Topics* now contains a description of how to use the CPU PROM monitor.

System Overview for the Sun Workstation

System Overview for the Sun Workstation is a new high-level overview of 'what's available' and where to look for more information. This document is still evolving.

System Interface Overview for the Sun Workstation

System Interface Overview for the Sun Workstation is a somewhat new manual — it was derived from the *System Interface Overview* that was previously stuck in the front of the *System Interface Manual*. The *System Interface Manual* now contains only manual pages and is completely reference oriented.

The System Administration Manual

The *System Administration Manual* contains new sections on the disk quota system and netgroups. There have also been numerous updates and clarifications. In particular, the section on disk maintenance has been rewritten and expanded.

The Pixrects Reference Manual

The Pixrects Reference Manual is a new programmer's guide to pixrects, the low-level device-independent graphics interface for Sun Workstations.

SunCGI Reference Manual

The *SunCGI Reference Manual* has been substantially corrected and clarified.

Beginner's Guide to Unix

A new beginner's guide for the complete novice.

Doing More With Unix

Doing More With Unix is intended for UNIX beginners who have read and understood the *Beginner's Guide to Unix* and would like to learn more.

Program Debugging Tools for the Sun Workstation

This new manual describes three debuggers available on Sun Workstations: `dbx`, `dbxtool`, `adb`, as well as the related tool `adbgen`.

Changes to the Manual Pages

There have been hundreds of changes to the man pages:

- Numerous bugs in them have been fixed.
- The `cc` and `adb` man pages have been updated to reflect Sun3 architecture and the presence of the 68881 floating-point coprocessor.
- Many other man pages in sections 3 and 4 have been updated to reflect the Sun3 architecture.
- `Intro (4)` has had a section added to document `ioctl` calls on files.
- The indexes are more comprehensive. In particular, the section 2 error codes and the `ioctl` calls are now indexed.

- There is a new section — 3R — which documents remote procedure call service functions.
- Some man pages, like `ptrace(2)` and `config(8)`, have been extensively rewritten to reflect major changes in code and machine architecture.
- Other man pages, which correspond to Release 3.0 utilities derived from System V (e.g. `make`, `mail`, and `sh`) are completely new and reflect the System V versions.

Changes to -ms and -me Macro Packages

The “cut marks” have been removed from the -ms and -me macro packages. “Cut marks” are small marks placed between pages on a typesetter film to indicate where the film should be cut to produce pages. They are useful only for typesetters, and with most people now using some form of laser printer they range from being a nuisance on some printers to producing incorrect page breaks on others.

Customers wishing to reinstate the cut marks should edit the source files for the appropriate macro package — `/usr/lib/tmac/tmac.s` for the -ms macro package, and `/usr/lib/tmac/tmac.e` for the -me macro package. Go to the end of the appropriate file and look for the construct `‘.rm CM’` (in the -ms macro package), or `‘.rm @m’` (in the -me macro package). Either remove the line altogether or comment it out.

Font Changes

`fonttool` has been renamed and is now called `fontedit`.

New Fonts

- This release includes a new set of serif fonts, loosely based on Prestige typewriter, in 10, 11, 12, 14, and 16 point.
- This release also includes a new set of Courier typewriter fonts, both Roman and Bold, in 10, 12, and 14 point.

1.8. New Setup Interfaces

System setup procedures and procedures for installing UNIX on the Sun have been entirely revised. `setup` now has both a `TTY` (character oriented) and a bitmap display (window-based) interface. It resides in the mini-Unix file system, and contains provisions for the installation of the root filesystem, the installation of optional software, network configuration, disk layout and diskless client partitioning, etc. For further details see *Installing Unix on the Sun Workstation*.

1.9. Support for New Peripherals

Driver Support for QIC-11 and QIC-24 Cartridge Tape Formats

The SCSI tape driver has been modified to support both the QIC-11 format and the QIC-24 format. The QIC-11 format is used on Sun2 machines by default (and all except the very oldest also provide hardware support for QIC-24). Sun3 machines support both QIC-11 and QIC-24. QIC-11 format is accessed through device `/dev/rst0`. QIC-24 format is accessed through device `/dev/rst8` (if the hardware supports that format). For more info, see `st(4S)`.

1.10. New Games

Several new games, boggletool and hack are included in release 3.0.

Revision History

Revision	Date	Comments
50 A	October 7, 1985 January 8, 1986	Change Notes for Release 3.0 Change Notes for Release 3.0

Notes

Notes

Notes

Notes

Notes

Notes

Notes