



Release 3.5 Manual for the Sun Workstation[®]

Sun Workstation, Sun-1, and Sun-2, Sun-3 are trademarks of Sun Microsystems, Incorporated.

Multibus is a trademark of Intel Corporation.

UNIX is a trademark of AT&T Bell Laboratories.

VMEbus is a trademark of VMEbus Manufacturers Group.

Copyright © 1987 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

Chapter 1 Introduction	3
1.1. Supporting Documentation	4
1.2. Documentation Conventions	4
Chapter 2 Release 3.5 Upgrade Installation	9
2.1. Introduction	9
Terminology	10
Requirements For Each Configuration	10
Booting a Special Kernel	10
Standalone System with Local Tape Drive	11
Standalone System with Remote Tape Drive	11
Homogeneous Server with Local Tape Drive	11
Homogeneous Server with Remote Tape Drive	12
Heterogeneous Server with Local Tape Drive	12
Heterogeneous Server with Remote Tape Drive	13
Tables for the Upgrade	13
2.2. Upgrade Overview	17
2.3. Example Upgrade Walkthroughs	25
Upgrade A Standalone System with a Local Tape Drive	25
Upgrade A Standalone System With Remote Tape Drive	27
Upgrade A Homogeneous Server With Local Tape Drive	29
Upgrade A Homogeneous Server With Remote Tape Drive	31
Upgrade A Heterogeneous Server With Local Tape Drive	33
Upgrade A Heterogeneous Server With Remote Tape Drive	36

2.4. How To Upgrade A Client Partition To 3.5	39
The Steps to Upgrade Client2 to 3.5	40
2.5. How To Restart The Upgrade Procedures	41
Chapter 3 Reconfiguring Your Kernel	47
3.1. Procedures for Reconfiguring the Kernel	47
Kernel Reconfiguration for Standalone Systems	47
Kernel Reconfiguration for Servers	48
Kernel Reconfiguration for Clients	49
3.2. Kernel Reconfiguration — an Annotated Copy of a <i>GENERIC</i> File	51
3.3. Sun-2 <i>GENERIC</i> Configuration File	58
3.4. Sun-3 <i>GENERIC</i> Configuration File	60
Chapter 4 Changes To Hardware and Software in Release 3.5	67
4.1. New Hardware Functionality	67
Sun-3 Eurocard	67
Sun-3/60	67
ALM-2	67
380MB SCSI Drive	68
1GB SMD Drive	68
4.2. Graphics Enhancements	68
4.3. SunView Enhancements	68
Double-Buffering Support in Release 3.5	68
How Double-Buffering Works	68
Prototypical Application Code	69
Pixwin Interface	69
pw_dbl_access()	69
pw_lock()	69
pw_unlock()	70
pw_dbl_flip()	70
pw_dbl_release()	70
pw_dbl_set()	70

pw_dbl_get ()	71
Pixrect Interface	71
pr_dbl_get ()	71
pr_dbl_set ()	72
Chapter 5 Bugfixes Since Release 3.4	75
5.1. Graphics Bugs Fixed	75
<i>SunCore</i> Bug Fixes	75
Text not Clipping	75
<i>Pixrect</i> Bugs Fixed	75
pr_polypoint on GP may Mess Up the First Point	75
pr_load Misbehaves in Case of Error	75
<i>SunCGI</i> Bugs Fixed	76
Bad Name Argument	76
close_cgi_pw() Fails	76
Visible Mouse Cursor	76
Compile CGI Program	76
<i>Graphics Processor</i> Bug Fixes	76
PR_PGON_TEX does not Clip Texture to Screen Correctly	77
Incorrect GP1_PR_ROP_TEX Semantics	77
GP1_PR_PGON_TEX Problem	77
5.2. SunOS Bugs Fixed	77
Utility Bugs Fixed	77
2.59 sysdiag Disables Disk Testing in Systems with GP/GB	77
rasfilter8to1.1g Has Wrong Filename Suffix	77
sysdiag Assumes You are Always on the Console	77
sysdiag sptest Fails	77
make does not Handle Square Bracket Characters	77
look Dumps Core on Long Lines	77
SCCS deledit Duplicates Random Lines in a File	78
SCCS Uses Delta Times for Diffs	78
Problem with Sendmail	78

Core Dump	78
Duplicate Lines	78
Commands Ignored	78
SunPro Files not Installed	78
make did not Build	78
Printer Daemon Disappears	78
Undocumented <code>fsck</code> Error Message	79
Termcap Entry Breaks	79
5.3. Kernel Bugs Fixed	79
Kernel Bus Error on Sirius	79
Shell Script Crashes 3/200 Series Kernel	79
Colormap Updates Fail	79
System Flooded with <code>zs</code> Interrupts	79
Sysdiag Core Dump	80
Sysdiag Fills Root Partition	80
Physio Bug Causes <code>writev(2V)</code> Failure	80
Physio Breaks <code>readv</code>	80
SCSI MD21 (141 MB) Errors Cause System Hang	80
System Panics When Using <code>ttya</code> with SCSI-3	80
SCSI Disk Driver Hangs	80
UNIX Hangs When Booting if Tape Drive On-line	80
Demo Programs Fail with a Core Dump	81
diag Causes Error Message	81
<code>ptrace</code> Interaction with Interrupting Slow System Calls	81
Misspelled Error Message	81
<code>cgtwo</code> Driver Watchdogs	81
Booting from 1/2 Inch Tape Hangs 3/260 System	81
Device Driver Panic	81
Setting Raw Mode Panics System	81
Error Handling Inconsistent	82
<code>mti</code> Error is Cryptic	82
TOD Chip Failure	82
<code>sys pt too small message</code>	82

Incorrect Comment	82
AF_UNIX SOCK_STREAM OOB	82
-ffpa Kernel Bus Error	82
Sun-2 SCSI Driver Bug	83
Redebugging Crashes System	83
5.4. Network Bugs Fixed	83
Network Routing Daemon in.routed Dies Periodically	83
Routing Daemon in.routed Sometimes Uses Wrong Interface	83
rcp Broadcast Address Computes Incorrectly	83
Bad ICMP Address Mask Reply	83
TCP/IP File Transfer Hangs	83
NFS Attribute Cache Functions Incorrectly	83
flock() Very Slow	83
if.c Changes to Support MCP and Subnets	84
TCP Performance Problem	84
RSTAT (3R) Returns 0	84
Incorrect ping Response	84
Wrong Loopback Address	84
rexid Race Condition	84
5.5. System Administration Bugs Fixed	84
rpc.lockd Fails to Free	84
Symbolic Links not Installed	84
5.6. SunView Bugs Fixed	84
pw_line() Draws in Window Space Instead of Canvas Space	84
pw_line() Draws Incorrectly When pw_batch_on()	85
Problem with pw_line and pw_polyline	85
tektool: Menu Missing Items and no Longer in Walking Style	85
tektool Key Does Not Work	85
tektool Could Core Dump	85
Crosshair Cursors not Working	85

Screenblank Too Effective	85
5.7. Compiler Bugs Fixed	86
Optimizer Substitution Error	86
dbx Shows Segmentation Violation	86
Recursive #includes Confuse dbx	86
ld -A Produces Bogus Symbol Tables	86
printf Padding Strings with Leading Zero's are Broken	86
dbx Failure	86
Chapter 6 3.4 Enhancements	89
6.1. Graphics Enhancements	89
Pixrect Enhancements	89
Unstructured Text Functions	89
Graphics Utilities	90
Ordered Dithering with <code>rasfilter8to1(1)</code>	90
Single Binary for Graphics Utilities	90
<code>screendump(1)</code>	90
<code>clear_colormap(1)</code>	90
SunCGI Enhancements	90
Signal Handling	90
Set VDC Extent	91
Additional CGI Error Messages	91
Open a CGI Canvas	92
6.2. SunView Enhancements	93
Improvements to <code>cmdtool</code>	94
Scroll Mode	94
Tty Mode	94
How Tty Mode Works	94
Compatibility of Applications with <code>cmdtool</code>	95
Compatibility of SunOS Releases with <code>cmdtool</code>	95
Changing Modes Manually	96
Tips and Tricks	96
Things to Watch For	97

Menu Enhancements	98
Stay-up Menus	98
Row-Column Order in Menus	99
Menu Compatibility	99
Corresponding New Menu Attributes	99
Text Subwindow Enhancements	99
‘‘Quadruple-clicking’’ Now Selects the Entire Document	99
Positive Confirmation When ‘Find’ Fails	99
The Upper Context is Set Correctly After ‘Find’	99
Text Subwindow’s Temporary File Name Shortened	99
Specifying Different Defaults	99
Miscellaneous Enhancements Visible to the User	100
New Tty Subwindow Attributes	100
TTY_TTY_FD	101
TTY_PID	101
TTY_ARGV_DO_NOT_FORK	101
Sample Uses of New Tty Attributes	101
Example Using TTY_TTY_FD to Read/Write Standard Input/Output	101
Example Using TTY_PID to Fork a New Process on Child Death	103
SunView Library Changes	104
lint Libraries	104
Profiled Libraries	104
Improved Attribute-Value List Error Handling	105
6.3. Kernel Enhancements	105
6.4. Networking Enhancements	105
Name Resolver Library	105
New IP, TCP, and ICMP from 4.3BSD	105
RPC Programs Read Broadcast Address	105
6.5. File Formats	105
Chapter 7 3.4 Bug Fixes	109

7.1. Language-related Bugs Fixed	109
as Bugs	109
68020 Addressing Bug	109
Assembler Long Jump Bug	109
C-related Bugs	110
op REG Compiler Error	110
Long Filenames Crashed cc	110
C Compiler Error with -f68881 Set	110
Casting Error Bug	110
Optimizer Failure	110
Optimizer Lost Address Register	110
cc Mishandled ',' Operator	111
3.1 Optimizer Bug	111
cc -f68881 Bug	111
Division by Powers of 2 Bug	111
doscan Bug	111
Coercion Bug	111
Unneeded File Bug	111
cc Conversion Bug	112
Type-casting Bug	112
Integral-conversion Bug	112
No Overwrite Warning Bug	112
Peephole Optimizer Bug	112
Optimizer Register Use Bug	112
Register Variable Bug	112
Debugger-related Bugs	113
dbx #include Bug	113
dbx Printing Bug	113
dbx Stack Trace Bug	113
dbx Function Pointer Bug	113
dbx Casting Bug	113
Incorrect Help Bug	114
dbx Symbol Table Bug	114

dbx Error Message Bug	114
dbx Prefix Bug	114
where Command Bug	114
catch and ignore Report Bug	114
dbx clear Command Bug	114
dbx “No Program” Bug	114
dbx Array Index Bug	114
Private Procedure Bug	115
Symbolic Link Bug	115
Double Negative Bug	115
Numeric Conversion Bug	115
FORTRAN-related Bugs	115
Complex Number Compare Bug	115
INCLUDE Bug	115
f77 ‘op REG’ Error Bug	115
Number Scaling Error Bug	116
f77 read Bug	116
FORTRAN Uppercase Bug	116
f77 read Bug	116
Bug in Assembly Phase of Inline Expansion	116
Inline Expansion Bug	116
f77 -a Bug	117
f77 Compiler Bug	117
Incorrect Handling of Backslash Characters	117
Problems with Profiling Options	117
Redundant I/O-List Parentheses Bug	117
system and fork Error Bug (1003162)	118
Pascal-related Bugs	118
Pascal Filename Bug	118
Library Bugs Fixed	118
prof.h Failure	118
Missing Library Routine	118
System V putpwent Bug	118

System V <code>ioctl</code> Bug	118
System V-related <code>curses</code> Bug	118
Duplicate Library Routine	118
Utility Bugs Fixed	118
<code>tcov</code> Failed on SIGCHLD	119
<code>lint</code> Error Output	119
Miscellaneous Bugs	119
<code>pxp</code> Bug	119
Compiler “Invalid” Options Bug	119
Double-Precision Transcendental Bug	119
7.2. Graphics Bug Fixes	119
Pixrect Bug Fixes	119
File I/O	119
Pixrect File I/O Rewrite	119
Type Declarations Added	120
<code>pr_load</code>	120
RMT_RAW Colormaps	120
Writing to a Rasterfile	120
RT_BYTE_ENCODED Rasterfiles	120
Rasterfile Filters	120
<code>lint</code> Library	120
Macros	121
<code>rop_fastloop</code> Macro	121
Memory Pixrects	121
Private Data Format for Memory Pixrects	121
8, 16, and 32-bit Memory Pixrect Support	122
<code>pr_vector</code>	122
<code>pr_get</code>	122
<code>pr_put</code>	123
<code>pr_rop</code>	123
Text	123
Memory Leaks with <code>pf_open</code>	123
<code>pr_close</code>	123

Opening vfont (5) Fonts	123
pf_ttext String Lengths	124
Miscellaneous	124
pr_getcolormap with bwone Pixrects	124
pr_batchrop	124
Global Definitions Removed from pr_traprop	124
pr_vector	124
pr_polypoint	124
pr_polygon_2	124
pr_replrop	124
pr_rop	125
pr_stencil	125
pr_line	125
Frame Buffer Device Drivers	125
gpone (4S)	125
FBIIOGVIDEO ioctl	125
cgfour 4S Support for the Sun-3/60 Frame Buffer	125
Graphics Utilities	125
rasfilter8to1 (1)	125
rastrepl (1) and RT_BYTE_ENCODED Rasterfiles	126
SunCGI Bug Fixes	126
Resizing Viewports	126
NOCLIP	126
Clear View Surface	126
Cvwsurf	126
Quitting a View Surface Tool	126
Pixwin Output	126
CGIPW and SunView	126
Request Input	126
cfmksizespecmode	127
GP View Surfaces	127
Rectangle Perimeters	127
Arc	127

Freed Memory with Await Event	127
Multiple Character Keyboard Events in CGIPW	127
Inquire Device Identification	127
View Surface Table	127
Obsolete Code Removed	127
Close View Surface	127
Deactivate View Surface	127
CGIPW Validity Check	127
CGIPW Set VDC Extent	128
Scaling	128
Inquire Text Extent	128
Transparent Text	128
FORTTRAN Input	128
VALUATOR Input with Await Event	128
Ignored Input with Await Event	128
Bad Data Structure Error Code with Await Event	128
SunCore Bug Fixes	128
Fat Vectors	128
raster Structure	128
put_raster	128
Clipped Vectors with GP View Surfaces	129
Polygon Vertex Limit	129
Backspace Key	129
7.3. SunView Bug Fixes	129
Defaults, *.rc Files, and Filters	129
.textswrc Processes Shell Metacharacters	129
Fixes to Input Filters	129
.ttyswrc String Changes	130
Blank lines in ~/.ttyswrc	130
Mail Aliases in defaultsedit	131
.suntools Can Accept Blank Lines	131
Other Bug Fixes Visible to the User	131
Cursor Correct on Sun-3/110	131

lockscreen -e on the Sun-3/110LC Color Display	131
‘UnZoom’ when Iconic	131
Characters Lost on Multi-character Input	131
Bug Fixes in cmdtool	131
Bug Fixes in textedit and the Text Subwindow	132
Programs Exiting in shelltool and cmdtool	133
Not Enough Swap Space in shelltool and cmdtool	133
Fixes to Tty Subwindows	133
Security Hole in lockscreen Fixed	134
Miscellaneous Fixes to Tools	134
User Interface Cleanup	134
Bug Fixes Visible to the Programmer	134
Window Geometry Fixes	134
WIN_ROWS When There is No Frame Header	134
Panel Choices Too High	135
‘Next’ Panel Item Created Too Low	135
Choice Item Positioning	135
Other Fixes Visible to the Programmer	135
Miscellaneous Text Subwindow Fixes	135
Pop Up Subframe Restrictions	136
WIN_FONT Now Works for Tty Subwindows	136
event_set_time()	136
KEY_BOTTOMFIRST and KEY_BOTTOMLAST Defined	136
Diagonal pw_copy() Fix	136
pw_line() in a Canvas	136
Click_to_Type in Panels	137
Cursor Positioning Escape Sequence in Tty Subwindows	137
Odd-Sized Icons	137
Internal Name Changes	137
Example Program filer	137
Other Fixes to Example Programs	137
7.4. SunOS Bug Fixes	138
Installation and Configuration-related Bugs	138

Missing Sun-3/110 Configuration Files	138
/sys/sundev/mti_conf.c File Missing	138
Remote Install Problems	138
Sun-3/50-4 Tape Cartridge Hangs	138
bzero Routine Could Not Handle Large MAXUSERS	138
Sun-3 Kernel Could Not Be Built Without Sun-3/260 Defined	139
Sun-3s with Early PROMs Had Rebooting Problems	139
3.3UPGRADE Did Not Work on Systems with 3.2EXPORT	139
Standalone Programs Did Not Exit Correctly	139
Kernel-related Bugs	139
Cursor Motion Not Coordinated with Mouse Motion	139
Memory Device Driver Bug	139
Bug in kadb	140
Error Reporting Problems with SCSI Tape Drives	140
VPC-2200 Parallel Printer/Plotter Interface Problems	140
Daisy Chained Shoe Boxes on Sun-3/50 Sometimes Hung	140
setrlimit Sometimes Did Not Return Error Codes	140
Occasional Spurious User Bus Errors Around Page Boundaries	140
New Version of the ie Ethernet Driver	140
Incompatible Routine in System V curses Library	140
Daylight Savings Times Were Incorrectly Calculated	141
MAKEDEV Did Not Create a vpc Node	141
Kernel Did Not Recognize Pseudo-terminals	141
Hard Links to Directories Caused System Panic	141
readlink Bug	141
Problems with kmem	141
indent-related Bug Fixes	141
4.3 BSD Bug Fixes	141
indent Problems with typedefs	141
Comment Blocks Were Incorrectly Formatted	141
Missing Semicolons Confused indent	142

indent Misformats Certain Combinations	142
Formatting Problems with Negative Floating-point Numbers	142
indent Put Spaces Around "\$"	142
Utilities-related Bug Fixes	143
Processes Failed to Reset Terminal Modes	143
Using sccs with a setuid Front End	143
Bug in dkinfo	143
Reply Sometimes Garbled Addresses	143
vgrind Omitted "?" Identifier	143
Numeric Fields Failed in tbl	143
.TH Macro Bug	143
Bug in maze Demonstration	144
Bug in chesstool	144
Complex Makefiles Caused Hash Table Overflow	144
indxbib Dropped Core	144
Shell-related Bug Fixes	144
Bug in login Command	144
time Memory Statistics Printed in Pages	144
7.5. Networking	144
Problems Using ifconfig for Diskless Clients	144
Bugs in TCP/IP	145
Net Booting on Non-zero Subnets	145
Telnet Server End-of-line Conventions	145
Frozen Configuration Files Ignored Domain Information	145
Lance Ethernet Driver Bug	145
ping Sent Bad Packets	145
rpcinfo Did Not Work on Non-networked Standalones	145
NFS Clients Sometimes Wrote Garbage	145
Software Loopback lo0 Address Could Not Be Set	146
FBIOGVIDEO Was Sometimes Unreliable	146
7.6. System Administration	146
Diskless Clients Can Now Reboot After Changing Ethernet Address with ifconfig	146

/etc/dkinfo	146
NFS Server Port Checking	146
Wrong termcap Entry for TVI-925	146
Errata and Addenda for 3.x Manuals	147
Chapter 8 Language Manuals	151
8.1. Language-related Addenda	151
dbx setenv Documentation Bug	151
as Addressing Modes Documentation	151
Language-related Errata	152
as Documentation Bug	152
Debugging Documentation Bug	152
lint Line Count Bug	152
adb Signal Bug	152
Chapter 9 File I/O Facilities for Pixrects	155
9.1. Writing and Reading Raster Files	155
Write Raster File	155
Read Raster File	157
9.2. Details of the Raster File Format	158
9.3. Writing Parts of a Raster File	159
Write Header to Raster File	159
Initialize Raster File Header	160
Write Image Data to Raster File	160
9.4. Reading Parts of a Raster File	160
Read Header from Raster File	160
Read Colormap from Raster File	161
Read Image from Raster File	161
Read Standard Raster File	161
Chapter 10 Windows and Window Based Tools: Beginner's Guide	165
10.1. Errata and Addenda for Windows and Window Based Tools Beginner's Guide	165

cmdtool Improved	165
“Quadruple-Clicking” in Text Subwindows	165
Chapter 11 Working with SunView	169
11.1. Working With SunView	169
11.2. General Tips	169
Other Sources of Information	169
Where are the Function Keys?	169
11.3. Common defaultsedit Changes	170
Walking Menus	170
SunView Default Font	170
Input Preferences	171
Miscellaneous defaultsedit Settings	171
11.4. Working with Selections	171
Pending-Delete Selections	171
Making Selections	172
Useful Operations on the Selection Using <code>get_selection</code>	172
Problems with <code>get_selection</code>	172
11.5. Secondary Get	173
The Basic Way to Copy Text	173
The Fast Way	173
Move the Cursor Back	173
<i>Click_to_Type</i> Mode	173
11.6. Other Common Text Subwindow Solutions	174
How Do I Read in a File Using <code>mailtool</code> ?	174
Scanning Digests	174
Putting <code>></code> in Front of Included Text (quick search and replace)	175
11.7. Souping Up Your Root Menu	176
Add Local Applications	176
Operations on the Selection	176
Standard Tools	177
User Interface	178

11.8. The Power of <code>.textswrc</code>	178
Getting Started with <code>.textswrc</code> Filters	178
How Filters Work	178
Pending-Delete or Not	179
Useful <code>.textswrc</code> Filters	179
Multiple Means to the Same End	179
Keyboard “Macros”	179
Short Macros	179
Long Macros	179
A Scrap/Test Filter	180
Running Any Command	180
From the Bourne Shell	180
From the C-Shell	180
Suggested Uses	181
Running the Selection through Any Command	181
Running the Selection through a “Run a Command” Filter	182
Summary (an Example)	182
11.9. Using <code>cmdtool</code>	183
Function Key Mappings	184
Macros with <code>mapi</code>	184
Escape Sequences Using <code>mapo</code>	184
Sample <code>~/cshrc</code>	184
11.10. Performance	186
Reduce CPU Overhead	186
Schedule Fewer Tasks for the CPU	186
Don’t Run <code>comsat</code>	187
Don’t Make Other Tools Repaint	187
Optimize Memory Usage	187
Use System Fonts	187
Merging Applications	187
Reconfigure Your Kernel	188
Improve Interactive Response	188
Use Retained Subwindows	188

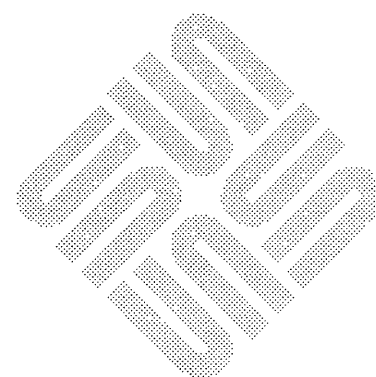
Learn Keyboard Accelerators	188
Increase Mouse “Gain”	188
11.11. Miscellaneous	188
Macintosh Users	188
Editing Operations	189
Unpacking Shell Archives	189
Chapter 12 SunView Programmer’s Guide	193
12.1. Errata and Addenda for the SunView Programmer’s Guide	193
Maximum Attribute-Value List Size	193
Fitting Frames Around Subwindows	193
Restrictions on Pop Up Windows	193
Window Positioning	194
WIN_EXTEND_TO_EDGE Invalid for Frames	194
Zero WIN_X	194
Changing Subwindow Size	194
Attribute Ordering	194
File Descriptor Usage Table	194
Canvas Resizing	195
LOC_TRAJECTORY Events	195
TEXTSW_INSERT_MAKES_VISIBLE Does Not Affect textsw_insert()	196
Cautions for Default Panel Item Layout	196
Menu Generate Proc Attributes	196
Nonexistent WIN_ICON Attribute	197
Creating Scrollbars	197
Differences Between notify_set_signal_func() and signal(3)	197
Correct PANEL_ITEM_Y_GAP Value	198
Undocumented PANEL_MARK_IMAGE and PANEL_NOMARK_IMAGE Attributes	198
Correct TEXTSW_BLINK_CARET Default	199
Revised typein.c Example Program	199

Chapter 13 SunView System Programmer's Guide	205
13.1. Errata and Addenda for the SunView System Programmer's Guide	205
Scrollbar Chapter	205
 Chapter 14 System Administration Manual	 209
14.1. System Administration Manual Addenda	209
/etc/ttys File	209
Run <code>catman</code> to Access Online Manual	210
Hooking Up a Serial ASCII Printer	210
Hooking Up a Printer to a VPC-2200 Multibus Board	210
Major Changes in Disk and Directory Layout in Release 3.0 and Later Releases	211
Preparing a Previously-Used Client Partition	211
<code>dump</code> Cannot Be Used to Dump an NFS-mounted File System	212
How to Use <code>yppasswd</code> Versus <code>passwd</code>	212
Print Filter Hook	212
<code>fstab</code> Entry for a Sun-2/3Com Diskless Client Mounting from a Sun-3 Server	214
How to Add a New <code>yp</code> Slave Server Not in the Original Set	214
<code>uucp L.cmds</code> File	215
 Appendix A Reference Manual Pages	 219
 Appendix B Installing UNIX on Tapeless Workstations	 225
B.1. Overview of the Installation Procedure	225
B.2. Configuring the Remote Host as a Network Disk	226
B.3. Configuring the Remote Host as a Network Disk Server	226
B.4. Configuring the Remote Host	227
B.5. Determining Network Information	230
B.6. Loading the Mini UNIX System on the Remote Host	231
B.7. Using <code>diag</code> to Format and Label the Target Machine's Disk	232
B.8. Loading the Mini UNIX System	232
B.9. Booting the Mini UNIX System	233

Appendix C Installing SunPro	237
C.1. Installation Instructions	237
Appendix D Files to Be Saved	241
D.1. Standalone Systems	241
D.2. Servers	241
D.3. Diskless Clients	242
D.4. YP Masters and Non YP Machines	242
Appendix E Release 3.3 Change Notes	245
E.1. disconnect/reconnect	245
Product Description	245
Advantages to the User	247
E.2. Subnets	247
The Sun System Release 3.3 Implementation	247
Enabling Subnets	248
Setting Up a Subnet	249
Examples of Subnets	249
E.3. Extensions to the Pascal Compiler	250
char Data Type	250
Public or Private Declarations	250
GETFILE Function	250
E.4. Bug Fixes	250
Locking Bugs	251
tbl Layout Processor Bug	251
Sun-3 ECC Error Message Bug	251
Problems with 1/2 Inch Tapes on Sun 3/280s	251
MT02 Tape Appending Problems	252
MT02 and VPC-2000 Tape Hangs	252
Interphase Controller Reference	252

Figures

Figure 6-1 CGI Canvas Example Program	92
Figure 6-2 A Split <code>cmdtool</code>	96
Figure 6-3 Cursor Indicating “Stay Up” Menu Imminent	98
Figure 6-4 Sample Centered Menu	98
Figure 7-1 Memory Pixrect Program Example	122
Figure 9-1 Example Program with <code>pr_dump</code>	157
Figure 9-2 Example Program with <code>pr_load</code>	158
Figure E-1 SCSI Bus Usage with and without disconnect/reconnect	246



Tables

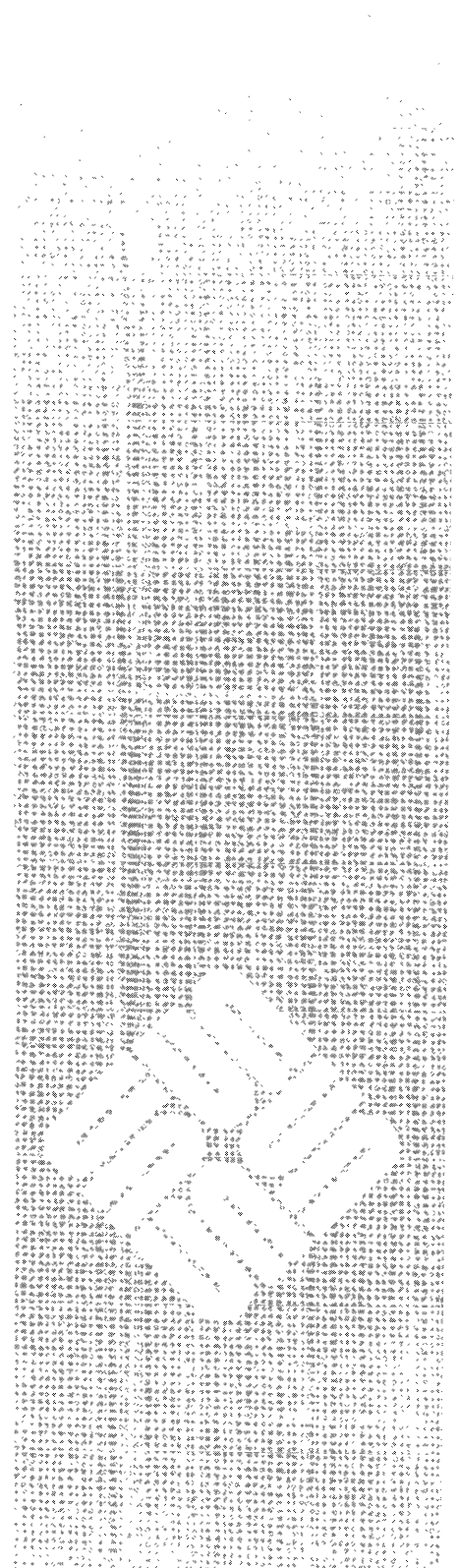
Table 2-1 Default Addresses for Disk Controllers	9
Table 2-2 System Sizes without Optional Software	14
Table 2-3 Table of Contents for the 1/2-Inch Distribution Tape	15
Table 2-4 Table of Contents for the 1/4-Inch Distribution Tape	16
Table 2-5 Optional Software Space Requirements	18
Table 2-6 Tape Devices	19
Table 2-7 Disk Devices	19
Table 2-8 Ethernet Types	20
Table 2-9 Optional Software	26
Table 2-10 Optional Software	28
Table 4-1 Pixwin-Level set Attributes	71
Table 4-2 Pixwin-Level get Attributes	71
Table 4-3 Pixrect-Level get Attributes	72
Table 4-4 Pixrect-Level set Attributes	72
Table 6-1 New Tty Subwindow Attributes	100
Table 11-1 SunView Function Key Mnemonics	170
Table A-1 Reprinted Reference Manual Pages	219
Table A-2 Corrected On-Line Reference Pages	221

Introduction

Introduction 3

1.1. Supporting Documentation 4

1.2. Documentation Conventions 4



Introduction

This manual explains how to install the new SunOS™ Release 3.5 software and describes the many features and bug fixes included in the release. The text is organized as follows:

- Chapter 1 gives an overview of the document and the conventions it uses.
- Chapter 2 contains the installation instructions for upgrading to Release 3.5.
- Chapter 3 explains how to reconfigure the kernel for Release 3.5.
- Chapter 4 explains new software support for the new hardware features.
- Chapter 5 describes the bug fixes made for Release 3.5.
- Chapter 6 contains the enhancements included in Release 3.4.
- Chapter 7 contains the bug fixes included in Release 3.4.
- Chapter 8 is an addendum to languages documentation.
- Chapter 9 is an addendum to file I/O facilities for pixrects.
- Chapter 10 provides errata and addenda for previous releases of the *Windows and Window Based Tools: Beginner's Guide*.
- Chapter 11 is an addendum to the *Beginner's Guide*, which describes various aspects of the SunView Environment.
- Chapter 12 provides errata and addenda for the SunOS 3.2 version of the *SunView Programmer's Guide*.
- Chapter 13 provides errata and addenda for SunOS 3.2 version of the *SunView System Programmer's Guide*.
- Chapter 14 provides errata and addenda for SunOS 3.0 version of the *System Administration for the Sun Workstation* manual.
- Appendix A contains a table of all additions and changes made to both the online and printed manual pages.
- Appendix B explains how to finish installing the SunPro optional software.

- Appendix C describes how to install on tapeless workstations.
- Appendix D tells you which files are automatically saved when installing Release 3.5.

Release 3.5 is completely compatible with SunOS Releases 3.2, 3.3 and 3.4. Any program developed to run under these previous releases runs under 3.5. However, you must recompile to take advantage of the new features.

1.1. Supporting Documentation

The following manuals may be needed for further reference and to update your existing documentation if you are upgrading.

- *Installing UNIX on the Sun Workstation for Release 3.2*
- *Reference Manual Insertion Pages for Release 3.5*
- *Make User's Guide*
- *UNIX Interface Reference Manual*
- *Commands Reference Manual for the Sun Workstation*
- *Sun System Diagnostic Manual*
- *System Administration Manual*
- *System V Overview*

1.2. Documentation Conventions

The following conventions are used in the procedures and examples throughout this document:

- Prompts from the system are printed in *listing font* like *this*.
- Information that you type in response to the system is shown in **bold-face listing font like this**. Type everything shown in boldface exactly as it appears.
- Where parts of a command are shown in *italic text like this*, they refer to a variable that you have to substitute from a selection; it is up to you to make the proper substitution.
- Dialogues between you and the system are enclosed in gray boxes (except chapter 2) like the following:

```
host% ls personnel.rec
amina          azhar          bb             cameron
ernest         farrasha      gregorio      jim
linda         michael       stefania      susan
tony          turia
```

- Sections of program code are enclosed in clear boxes like the following:

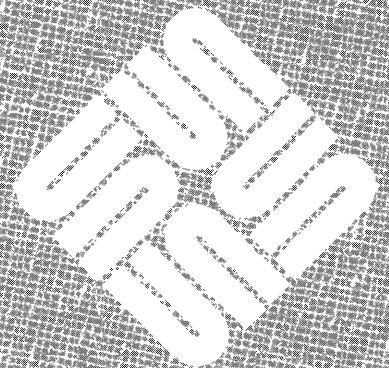
```
int test[100];

main()
{
    register int a, b, c, d, e, f;

    test[a] = b & test[c & 0x1] & test[d & 0x1];
}
```


Release 3.5 Upgrade Installation

Release 3.5 Upgrade Installation	9
2.1. Introduction	9
Terminology	10
Requirements For Each Configuration	10
Booting a Special Kernel	10
Standalone System with Local Tape Drive	11
Standalone System with Remote Tape Drive	11
Homogeneous Server with Local Tape Drive	11
Homogeneous Server with Remote Tape Drive	12
Heterogeneous Server with Local Tape Drive	12
Heterogeneous Server with Remote Tape Drive	13
Tables for the Upgrade	13
2.2. Upgrade Overview	17
2.3. Example Upgrade Walkthroughs	25
Upgrade A Standalone System with a Local Tape Drive	25
Upgrade A Standalone System With Remote Tape Drive	27
Upgrade A Homogeneous Server With Local Tape Drive	29
Upgrade A Homogeneous Server With Remote Tape Drive	31
Upgrade A Heterogeneous Server With Local Tape Drive	33
Upgrade A Heterogeneous Server With Remote Tape Drive	36
2.4. How To Upgrade A Client Partition To 3.5	39
The Steps to Upgrade Client2 to 3.5	40
2.5. How To Restart The Upgrade Procedures	41



Release 3.5 Upgrade Installation

2.1. Introduction

This chapter guides you through the SunOS™ Release 3.5 upgrade. The upgrade procedures in this chapter are designed to upgrade systems currently running SunOS Release 3.2, 3.3, 3.4, or 3.5Beta. Release 3.5 is completely compatible with these previous releases. However, you must recompile to take advantage the new features offered with this release.

NOTE *If you are receiving SunOS 3.5 with a new Sun system, or you are currently running a version of the operating system before 3.2, you must perform a full install of the the SunOS 3.5 release following the procedures in Installing Unix on the Sun Workstations for complete details. Also, if you are running SunOS 3.2 or better, and wish to change your current system defaults, you must also perform a full install following the procedures in Installing Unix.*

If you are installing a 3/E or a 3/60, the following table shows the default controller addresses:

Table 2-1 *Default Addresses for Disk Controllers*

<i>Controller Type</i>	<i>Address (hex)</i>	
	<i>1st Controller</i>	<i>2nd Controller</i>
Xylogics 440/450/451	ee40	ee48
SCSI for Multibus System	80000	84000
SCSI for VMEbus System	200000	
SCSI for the Sun-3/50	140000	
SCSI for the Sun-3/60	140000	
SCSI for the Sun-3/E	310000	

NOTE *There are two ways to install Release 3.5. If you have a new Sun workstation, or if you want to change your current system's defaults, you can use Setup to install Release 3.5. (See Installing Unix on the Sun Workstation.) Use the 3.5 upgrade script if you are running SunOS Release 3.2, 3.3, 3.4, or 3.5Beta and you want your defaults to remain the same. Setup supports new client cards for the Sun-3 Eurocard and the Sun-3/60, 3/260, and 3/280 workstations.*

This upgrade procedure supports the following configurations:

- Standalone machines with a local tape drive and disk(s).
- Standalone machines with a disk(s) but without a local tape drive.
- Homogeneous servers with a local tape drive and a disk(s).
- Homogeneous servers with a disk(s) but without a local tape drive.
- Heterogeneous servers with a local tape drive and a disk(s).
- Heterogeneous servers with a disk(s) but without a local tape drive.

Please read through this chapter a few times to familiarize yourself with the procedures before beginning the upgrade.

Terminology

Below are some of the specific terms used in this chapter. It assumes that you have some experience in installing SunOS on a Sun workstation. If you do not understand some of the following instructions or terms, refer to *Installing UNIX on the Sun Workstation* and the *System Administration Manual for the Sun Workstation* for more details.

Standalone	A system with a disk that has its own root <code>/usr</code> file system, does not require a file server to boot, and may or may not be on the network.
Tapehost	The system with the tape drive.
Homogeneous server	A server that supports clients of its own architecture.
Heterogeneous server	A server that supports both Sun-2 (MC68010) and Sun-3 (MC68020) architectures.

Requirements For Each Configuration

This section lists the requirements of each configuration that the upgrade procedures support. Read the information in the section pertaining to your particular system configuration very carefully. Your system **MUST** meet all the requirements listed before you can properly upgrade it.

Upgrading with a remote tape drive is slower than upgrading with a local tape drive. Unless you have no other choice, upgrading with a local tape drive is highly recommended. Client partitions that are commented out in `/etc/nd.local` on a server will not be upgraded. Refer to section 2.4 of this chapter for instructions on how to upgrade these partitions.

Booting a Special Kernel

A client that boots a special kernel from its root partition is upgraded with the 3.5 GENERIC kernel in the root partition after the upgrade. If you want that client to boot the kernel that it ran before the upgrade, follow these steps:

1. Make sure there are no compatibility problems.
2. Copy the kernel to a directory that will not be affected by the upgrade.

3. Move the kernel back to the client root partition after the upgrade and before you bring up the client system.

If you wish the client to boot a special 3.5 kernel, configure a new kernel after the upgrade, then move it to the client root partition before you bring up the client system.

This chapter describes the following upgrade walkthroughs:

NOTE *When performing a remote tape upgrade, whether or not your system uses Yellow Pages, your tapehost's Internet address must be added to `/etc/hosts` on the existing (3.2, 3.3 3.4, or 3.5Beta) system before taking your system down in order to upgrade to 3.5.*

Standalone System with Local Tape Drive

Upgrading a standalone system with a local tape drive.

- This configuration is a standalone system running SunOS Release 3.2, 3.3, 3.4, or 3.5Beta with a local tape drive.
- The release tape(s) should be appropriate for the standalone system's architecture.

Standalone System with Remote Tape Drive

Upgrading a standalone system with a remote tape drive.

- This configuration is a standalone system running SunOS Release 3.2, 3.3, 3.4, or 3.5Beta.
- The release tape(s) should be appropriate for the standalone system's architecture and the tapehost's tape drive type.
- The tapehost must be reachable from the standalone system through the Ethernet. The tapehost and the standalone system must be on the same network.
- The `/etc/hosts` on a standalone system to be upgraded from a remote tape must contain the Internet address of the tapehost.
- `/etc/hosts` on the tapehost must contain the Internet address of the standalone system if YP is not running. If YP is running, the `/etc/hosts` file on the master YP server must contain the Internet address of the standalone system.
- `.rhosts` on the tapehost must contain the name of the standalone system.

Homogeneous Server with Local Tape Drive

Upgrading a homogeneous server with a local tape drive.

- This configuration is a homogeneous server running SunOS Release 3.2, 3.3, 3.4, or 3.5Beta with a local tape drive.
- The release tape(s) should be appropriate for the system's architecture and tape drive type.

- Make sure that `/etc/hosts` and `/etc/ethers` on the homogeneous server contain correct Internet and Ethernet addresses of all clients supported by the server.

Homogeneous Server with Remote Tape Drive

Upgrading a homogeneous server with a remote tape drive.

- This configuration is a homogeneous server running SunOS Release 3.2, 3.3, 3.4 or 3.5Beta.
- The release tape(s) should be appropriate for the system's architecture and the tapehost's tape drive type.
- Make sure that `/etc/hosts` and `/etc/ethers` on the homogeneous server contain correct Internet and Ethernet addresses of all clients supported by the server.
- The tapehost must be reachable from the homogeneous server through the Ethernet. The tapehost and the homogeneous server must be on the same network.
- The `/etc/hosts` on the homogeneous server to be upgraded from a remote tape must contain the Internet address of the tapehost.
- `/etc/hosts` on the tapehost must contain the Internet address of the homogeneous server if YP is not running. If YP is running, the `/etc/hosts` file on the master YP server must contain the Internet address of the standalone system.
- `.rhosts` on the tapehost must contain the name of the homogeneous server.

Heterogeneous Server with Local Tape Drive

Upgrading a heterogeneous server with a local tape drive.

- This configuration is a heterogeneous server running SunOS Release 3.2, 3.3, 3.4, or 3.5Beta with a local tape drive.
- The release tapes must be appropriate for both architectures and suitable for the local tape drive. For example, you must have suitable Sun-2 and Sun-3 tapes, and tapes must both be 1/4 or 1/2 inch. You cannot mix media types.
- Make sure that `/etc/hosts` and `/etc/ethers` on the heterogeneous server contain correct Internet and Ethernet addresses of all clients supported by the server.
- Make sure that each client mounts a `usr` file system of its own architecture from the server. An MC68010 client uses the server's `/usr.MC68010` file. An MC68020 client uses the server's `/usr.MC68020` file. Make sure that the `/etc/fstab` file for each client partition contains the correct information.

Heterogeneous Server with Remote Tape Drive

Upgrading a heterogeneous server with a remote tape drive.

- This configuration is a heterogeneous server running SunOS Release 3.2, 3.3, 3.4, or 3.5Beta.
- The release tapes must be appropriate for both architectures and suitable for the remote tape drive. For example, you must have suitable Sun-2 and Sun-3 tapes, and tapes must both be 1/4 or 1/2 inch. You cannot mix media types.
- Make sure that `/etc/hosts` and `/etc/ethers` on the heterogeneous server contain correct Internet and Ethernet addresses of all clients supported by the server if it is not running YP. If it is running YP, the `/etc/hosts` and `/etc/ethers` files on the master YP server must contain this information.
- The tapehost must be reachable from the heterogeneous server through the Ethernet. The tapehost and the heterogeneous server must be on the same network.
- The `/etc/hosts` on the heterogeneous server to be upgraded from a remote tape must contain the Internet address of the tapehost.
- `/etc/hosts` on the tapehost must contain the Internet address of the heterogeneous server if YP is not running. If YP is running, the `/etc/hosts` file on the master YP server must contain the Internet address of the standalone system.
- `.rhosts` on the tapehost must contain the name of the heterogeneous server.
- Make sure that each client mounts a `usr` file system of its own architecture from the server. An MC68010 client uses the server's `/usr.MC68010` file. An MC68020 client uses the server's `/usr.MC68020` file. Make sure that the `/etc/fstab` file for each client partition contains the correct information.

Note that when you upgrade heterogeneous systems, the server will try to upgrade its own architecture first, then the other serving architectures. For example, if the heterogeneous server is a Sun-2 serving both Sun-2 and Sun-3 clients, the Sun-2 clients are updated before the Sun-3 clients.

Tables for the Upgrade

The next section provides a table for system sizes, and table of contents for 1/2-inch and 1/4-inch tapes.

Table 2-2 *System Sizes without Optional Software*

<i>System</i>	<i>Size</i>
<i>Standalone System</i>	
/	5652K
/usr	14215K
<i>Homogeneous Server</i>	
/	2750K
/pub	3836K
/usr	14192K
<i>Heterogeneous Server</i>	
/	2865K
/pub.MC68010	3929K
/pub.MC68020	3836K
/usr.MC68010	13607K
/usr.MC68020	14192K

WARNING *Each client root partition requires at least 1661 Kbytes.*

The following tables give the contents of the distribution tapes and the order in which the tape files appear.

Table 2-3 *Table of Contents for the 1/2-Inch Distribution Tape*

Tape	File	Contents
1	1	tape boot block
1	2	toc
1	3	copyright
1	4	diag
1	5	copy
1	6	mini-root
1	7	root
1	8	pub
1	9	client image
1	10	Sys
1	11	Networking tools and programs
1	12	Debugging tools
1	13	usr
1	14	copyright
2	1	copyright
2	2	toc
2	3	SunView User Programs
2	4	SunView Programmers Files
2	5	SunView and Demo Program source
2	6	Text Processing tools
2	7	Setup tools
2	8	Standalone Diagnostics
2	9	Fortran Compiler & Libraries
2	10	User Level Diagnostics
2	11	SunCore & CGI Libraries
2	12	Pascal Interpreter & Compiler
2	13	Profiled Libraries
2	14	uucp programs
2	15	copyright
3	1	copyright
3	2	toc
3	3	System V programs and libraries
3	4	Manual Pages
3	5	Demonstration Programs
3	6	Games
3	7	SunPro
3	8	Versatec Printer Software
3	9	copyright

Table 2-4 *Table of Contents for the 1/4-Inch Distribution Tape*

Tape	File	Contents
1	1	tape boot block
1	2	toc
1	3	copyright
1	4	diag
1	5	copy
1	6	mini-root
1	7	root
1	8	pub
1	9	client image
1	10	copyright
2	1	copyright
2	2	toc
2	3	Sys
2	4	Networking tools and programs
2	5	Debugging tools
2	6	usr
2	7	copyright
3	1	copyright
3	2	toc
3	3	SunView User Programs
3	4	SunView Programmers Files
3	5	SunView and Demo Program source
3	6	Text Processing tools
3	7	Setup tools
3	8	Standalone Diagnostics
3	9	Fortran Compiler & Libraries
3	10	User Level Diagnostics
3	11	SunCore & CGI Libraries
3	12	Pascal Interpreter & Compiler
3	13	Profiled Libraries
3	14	uucp programs
3	15	copyright
4	1	copyright
4	2	toc
4	3	System V programs and libraries
4	4	Manual Pages
4	5	Demonstration Programs
4	6	Games
4	7	SunPro
4	8	copyright
5	1	copyright
5	2	toc
5	3	Versatec Printer Software
5	4	copyright

2.2. Upgrade Overview

NOTE *Full backups are strongly recommended before you start the upgrade. If the system is a server, remember to halt all clients before the backups. Your home directories on the disk will not be touched by the upgrade procedure. Therefore, you can copy the files you want to save on the disk to your home directory and restore the files after the upgrade.*

1. Log in to the system and use the `df(1)` command to display information about the space available in each file system. All optional software currently existing on the disk(s) automatically loads with the latest version. The upgrade procedure then asks if you want to load any optional software that does not exist on the disk(s).
2. If you want to load additional software, you **MUST** make sure the file systems have enough space available **BEFORE** you start the upgrade. Use the table below (Table 2-2) to figure out the space requirements on your disks. If there is not enough space to load additional software, it is recommended that you run *Setup* to reinstall your system with adequate disk space. See *Installing UNIX on the Sun Workstation*.

Table 2-5 *Optional Software Space Requirements*

<i>Optional Software</i>	<i>Size</i>
Networking	2568 /usr
Debugging	1168K /usr
Suntools_users	2180K /usr
Suntools_programmers	2568K /usr
Suntools_source	574K /usr
Text_processing	874K /usr
Setup	1071K /usr
Stand_diag	4K /usr 4K /pub
Fortran	1157K /usr
Usr_diag	1988K /usr
Graphics	3125K /usr
Pascal	1141K /usr
Profiled	2681K /usr
Uucp	536K /usr
System V	3820K /usr
Man	4899K /usr
Demo	2477K /usr
Games	3977K /usr
SunPro	1087K /usr
Vtroll	6680K /usr

NOTE While running the upgrade, if you choose SunPro as optional software, there is a separate procedure to finish installing SunPro. See Appendix B for the instructions.

The software program Profiled includes the SunView Profile Library. This library consists of:

```
/usr/lib/libsuntool_p.a
/usr/lib/libsunwindow_p.a
/usr/lib/libtoolmerge_p.a
```

The SunView Profile Library takes up about 1.5 MB of space.

NOTE You must halt all your clients before halting the system if your system is a server.

3. Become superuser and halt your system by entering:

```

host% su
password:
host# sync
host# /etc/halt

```

4. If you are upgrading the system with a remote tape drive, skip this step and go to the next step. If you are upgrading the system with a local tape drive, boot the general purpose bootstrap program from the tape by typing **b**. Then enter a two character device abbreviation for your tape drive type, and open and closed parentheses.

```
>b tape ()
```

Table 2-6 *Tape Devices*

<i>Devices</i>	<i>Description</i>
<i>ar</i>	Archive quarter-inch tape cartridge (Sun-2 only)
<i>mt</i>	Nine-track magnetic 1/2" tape-Tapemaster controller
<i>st</i>	SCSI tape controller cartridge
<i>xt</i>	Nine-track magnetic 1/2" tape-Xylogics 472 controller

5. Load the minimal subset of the SunOS called the miniroot onto your disk. All upgrade software resides in the miniroot.
6. If you are upgrading with a local tape drive, enter the following:

```

Boot: tape (0, 0, 4)
Standalone Copy
From: tape (0, 0, 5)
To: disk (0, 0, 1)

```

Table 2-7 *Disk Devices*

<i>Devices</i>	<i>Description</i>
<i>xy</i>	Xylogics 440/450/451 SMD disk controller
<i>sd</i>	SCSI disk controller
<i>ip</i>	Interphase disk controller (Sun-2 only)

NOTE Before you can begin upgrading with a remote tape drive, you must copy the miniroot from tape onto the tapehost's disk. See Appendix C, Installing UNIX

on Tapeless Workstations, sections C.2-C.7.

7. If you are upgrading with a remote tape drive, do the following:

```
>b ethernet(0,serverhost)stand/copy
From: ethernet(0,serverhost,pub#)miniroot
To: disk(0,0,1)
```

Table 2-8 Ethernet Types

Type	Description
ec	3COM Ethernet controller (Sun-2 only)
ie	Sun Ethernet controller
le	Sun 3/50, 3/60 Ethernet controller

serverhost is the host number of tapehost, which has the bootable version of the miniroot, in hexadecimal representation. Refer to the section *Requirements for Each Configuration* in the beginning of this chapter.

The *pub#* is 0 if the system you are upgrading is Sun-2 and 1 if the system you are upgrading is Sun-3.

8. Boot the miniroot from your disk.

```
Boot: disk(0,0,1)vmunix -as
.
.
.
root device ? disk0*
.
.
.
```

9. When the system is up and displays a # prompt, make sure the date is correct. Now change your working directory to `/usr/etc/upgrade`.

```
# cd /usr/etc/upgrade
```

10. Start the upgrade procedure by typing the following command:

```
# UPGRADE
```

11. Specify the type of the system.

```
Enter system type ? [standalone | server]:
```

12. If the system is a server, you need to specify whether it is a homogeneous server or a heterogeneous server.

```
Enter server type ? [homo | heter]:
```

13. Specify whether the upgrade will be done with a local tape drive or remote tape drive. If you are upgrading the system with a remote tape drive, make sure you meet all the requirements specified in the last section.

```
Enter tape drive type ? [local | remote]:
```

14. If you are upgrading the system with a remote tape drive, you also need to specify the name of the tapehost and the Ethernet type of your system.

```
Enter host of remote drive ?
```

```
Enter Ethernet type of this system ? [ec | ie | le]:
```

```
Using Ethernet 0 or Ethernet 1 ? [0 | 1]
```

See *Table 2-5* for Ethernet type information.

15. Specify abbreviation of tape device.

```
Enter tape type ? [ar | st | mt | xt]:
```

See *Table 2-3* for device abbreviation.

16. Specify whether or not the system is running Yellow Pages.

```
Enter yp type of machine? [master | slave | client |
none]:
```

17. Specify the disk partition where the root file system resides.

```
Enter root disk partition for the MC680x0 architecture
(e.g. xy0a)?
```

18. Specify the list of optional software to be loaded. If the system is a heterogeneous server, you need to specify the list of optional software to be loaded for both MC68010 and MC68020 architectures. Optional software currently existing on the disk will be automatically loaded. Optional software currently not existing on the disk will be prompted for your attention.

```
Select optional software for the MC680x0 architecture :
```

```
Do you want to install "Networking"? [y/n]:
Do you want to install "Debugging"? [y/n]:
Do you want to install "Suntools_users"? [y/n]:
Do you want to install "Suntools_programmers"? [y/n]:
Do you want to install "Suntools_source"? [y/n]:
Do you want to install "Text_processing"? [y/n]:
Do you want to install "Setup"? [y/n]:
Do you want to install "Stand_diag"? [y/n]:
Do you want to install "Fortran"? [y/n]:
Do you want to install "Usr_diag"? [y/n]:
Do you want to install "Graphics"? [y/n]:
Do you want to install "Pascal"? [y/n]:
Do you want to install "Profiled"? [y/n]:
Do you want to install "Uucp"? [y/n]:
Do you want to install "System_V"? [y/n]:
Do you want to install "Man"? [y/n]:
Do you want to install "Demo"? [y/n]:
Do you want to install "Games"? [y/n]:
Do you want to install "SunPro"? [y/n]:
Do you want to install "Vtrotff"? [y/n]:
```

19. The UPGRADE saves some of your administrative files from your current release. After the UPGRADE is completed, these files remain on your disk. (See Appendix D for the file names of the files saved.) UPGRADE overwrites ALL other administrative files on your disk. If you have administrative files you want to save, you must rename them. For example, if you want to save your `/usr/lib/uucp/USERFILE`, you could enter

```
mv /usr/lib/uucp/USERFILE /usr/lib/uucp/USERFILE.OLD
```

After UPGRADE is completed, you can then rename it back to its original file name.

20. The upgrade procedure is about to begin. If you entered information incorrectly, you can answer `n` and restart by going back to step 7. If you are ready to start the upgrade, answer `y` and you will be prompted for attention only when tape needs to be changed.

```
Are you ready to start the upgrade ? [y/n] :
```

NOTE Below is the upgrade procedure for a Sun-3 standalone system called *godzilla* that was running 3.# before the upgrade. All optional software is chosen to be loaded in this example.

```
Beginning 3.# to 3.5 upgrade for the MC68020 architecture.
```

Changing directory to "/".

Extracting "root" files from "/dev/nrmt0" release tape.

[This takes approximately 2 1/2 minutes with mt/xt and approximately 2 1/2 minutes with ar/st.]

Extracting "pub" files from "/dev/nrmt0" release tape.

[This takes approximately 3 minutes with mt/xt and approximately 7 minutes with ar/st.]

Changing directory to "/usr".

Extracting "sys" files from "/dev/nrmt0" release tape.

[This takes approximately 8 minutes with mt/xt and approximately 12 minutes with ar/st.]

Extracting "user" files from "/dev/nrmt0" release tape.

[This takes approximately 14 minutes with mt/xt and approximately 15 minutes with ar/st.]

Extracting "Networking" files from "/dev/nrmt0" release tape.

[This takes approximately 5 minutes with mt/xt and approximately 4 minutes with ar/st.]

Extracting "Debugging" files from "/dev/nrmt0" release tape.

[This takes approximately 3 minutes with mt/xt and approximately 3 1/3 minutes with ar/st.]

Extracting "Suntools_users" files from "/dev/nrmt0" release tape.

[This takes approximately 4 1/2 minutes with mt/xt and approximately 3 5/6 minutes with ar/st.]

Extracting "Suntools_programmers" files from "/dev/nrmt0" release tape.

[This takes approximately 2 1/2 minute with mt/xt and approximately 4 1/4 minutes with ar/st.]

Extracting "Suntools_source" files from "/dev/nrmt0" release tape.

[This takes approximately 2 1/2 minutes with mt/xt and approximately 3 minutes with ar/st.]

Extracting "Text_processing" files from "/dev/nrmt0" release tape.

[This takes approximately 2 minute with mt/xt and

approximately 3 minutes with ar/st.]

Extracting "Setup" files from "/dev/nrmt0" release tape.

[This takes approximately 1 1/2 minutes with mt/xt and approximately 3 1/2 minutes with ar/st.]

Extracting "Stand_diag" files from "/dev/nrmt0" release tape.

[This takes approximately 1 1/2 minutes with mt/xt and approximately 3 minutes with ar/st.]

Extracting "Fortran" files from "/dev/nrmt0" release tape.

[This takes approximately 2 minute with mt/xt and approximately 3 minutes with ar/st.]

Extracting "Usr_diag" files from "/dev/nrmt0" release tape.

[This takes approximately 3 minute with mt/xt and approximately 5 minutes with ar/st.]

Extracting "Graphics" files from "/dev/nrmt0" release tape.

[This takes approximately 4 minutes with mt/xt and approximately 5 minutes with ar/st.]

Extracting "Pascal" files from "/dev/nrmt0" release tape.

[This takes approximately 4 minutes with mt/xt and approximately 4 minutes with ar/st.]

Extracting "Profiled" files from "/dev/nrmt0" release tape.

[This takes approximately 2 minutes with mt/xt and approximately 5 minutes with ar/st.]

Extracting "Uucp" files from "/dev/nrmt0" release tape.

[This takes approximately 2 minute with mt/xt and approximately 5 minutes with ar/st.]

Extracting "System_V" files from "/dev/nrmt0" release tape.

[This takes approximately 5 minutes with mt/xt and approximately 6 minutes with ar/st.]

Extracting "Man" files from "/dev/nrmt0" release tape.

[This takes approximately 15 minutes with mt/xt and approximately 17 minutes with ar/st.]

Extracting "Demo" files from "/dev/nrmt0" release tape.


```
[ This takes approximately 2 minute with mt/xt and
approximately 4 minutes with ar/st. ]

Extracting "Games" files from "/dev/nrmt0" release tape.

[ This takes approximately 2 minutes with mt/xt and
approximately 5 minutes with ar/st. ]

Extracting "SunPro" files from "dev/nrmt0" release tape.

Extracting "Vtroff" files from "/dev/nrmt0" release tape.

[ This takes approximately 4 minutes with mt/xt and
approximately 8 minutes with ar/st. ]

Checking filesystems :
/dev/rxy0a: 516 files, 2575 used, 4896 free (16 frags, 610 blocks)
/dev/rxy0g: 1678 files, 18810 used, 39733 free (69 frags, 4958 blocks)
3.# to 3.5 upgrade completed.
Reboot your system and configure a kernel for your system.
#
```

21. Abort the system by pressing **(LI)** and **(a)** at the same time, (like pressing **(control) (c)**), or by pressing **(Break)** if you are using a terminal as a console.

```
>b
```

22. Reconfigure a kernel for your system. Refer to *Chapter 3* for details.

2.3. Example Upgrade Walkthroughs

This section contains example upgrade walkthroughs for the following system configurations:

- Standalone System with a Local Tape Drive
- Standalone System with a Remote Tape Drive
- Homogeneous Server with a Local Tape Drive
- Homogeneous Server with a Remote Tape Drive
- Heterogeneous Server with a Local Tape Drive
- Heterogeneous Server with a Remote Tape Drive

Upgrade A Standalone System with a Local Tape Drive

Assume the standalone system has a local 1/2" tape drive (mt) and the system is running 3.4. Below is a list of the optional software that currently exists on the disk:

Table 2-9 *Optional Software*

<i>Optional Software</i>
<i>Suntools_users</i>
<i>Suntools_source</i>
<i>Fortran</i>
<i>Pascal</i>
<i>Profiled</i>
<i>Man</i>
<i>Suntools_programmers</i>

```

>b mt ()
Boot: mt (, , 4)
From: mt (, , 5)
To: xy (, , 1)
Boot: xy (, , 1) vmunix -as
.
.
.
root device ? xy0*
.
.
.
#
# cd /usr/etc/upgrade
# UPGRADE

Enter system type ? [standalone | server]: standalone

Enter tape drive type ? [local | remote]: local

Enter tape type ? [ar | st | mt | xt]: mt

Enter yp type for machine? [master | slave | client | none]: client

Enter root disk partition for the MC68020 architecture (e.g. xy0a) ? xy0
/dev/rxy0a: 516 files, 2575 used, 4896 free (16 frags, 610 blocks)
/dev/rxy0g: 1678 files, 18810 used, 39733 free (69 frags, 4958 blocks)

Select optional software for the MC68020 architecture :
Do you want to install "Debugging"? [y/n]: y
Do you want to install "Text_processing"? [y/n]: y
Do you want to install "Setup"? [y/n]: y
Do you want to install "Stand_diag"? [y/n]: y
Do you want to install "Usr_diag"? [y/n]: y
Do you want to install "Graphics"? [y/n]: y
Do you want to install "Uucp"? [y/n]: y
Do you want to install "System_V"? [y/n]: y
Do you want to install "Demo"? [y/n]: y

```

```

Do you want to install "Games"? [y/n]: y
Do you want to install "SunPro"? [y/n]: y
Do you want to install "Vtproff"? [y/n]: y

Are you ready to start the upgrade ? [y/n] : y

Beginning 3.4 to 3.5 upgrade for the MC68020 architecture.

Changing directory to "/".

Extracting "root" files from "/dev/nrmt0" release tape.

Extracting "pub" files from "/dev/nrmt0" release tape.

Changing directory to "/usr".

Extracting "sys" files from "/dev/nrmt0" release tape.

Extracting "user" files from "/dev/nrmt0" release tape.

Extracting "Networking" files from "/dev/nrmt0" release tape.

.
.
.

Extracting "SunPro" files from "/dev/nrmt0" release tape.
Extracting "Vtproff" files from "/dev/nrmt0" release tape.

Checking filesystems :
/dev/rxy0a: 516 files, 2575 used, 4896 free (16 frags, 610 blocks)
/dev/rxy0g: 1678 files, 18810 used, 39733 free (69 frags, 4958 blocks)

3.4 to 3.5 upgrade completed.
Reboot your system and configure a kernel for your system.

```

Upgrade A Standalone System With Remote Tape Drive

Assume the host number of the system we are booting from is 114. 114 in decimal is equal to 72 in hexadecimal. We have a Sun-3 machine and it is running 3.4. Therefore, the pub number is 1. Below is a list of the optional software that currently exists on the disk:

Table 2-10 *Optional Software*

<i>Optional Software</i>
<i>Suntools_users</i>
<i>Suntools_source</i>
<i>Fortran</i>
<i>Pascal</i>
<i>Profiled</i>
<i>Man</i>
<i>Suntools_programmers</i>

```

>b ie(,72)boot -a
Boot: ie(,72)stand/copy
From: ie(,72,1)miniroot
To: xy(,,1)
Boot: xy(,,1)vmunix -as
.
.
.

root device ? xy0*

.
.
.

#
# cd /usr/etc/upgrade
# UPGRADE

Enter system type ? [standalone | server]: standalone

Enter tape drive type ? [local | remote]: remote

Enter host of remote drive ? pebbles

Enter Ethernet type of this system ? [ec | ie | le] : ie

Enter tape type ? [ar | st | mt | xt]: st

Enter yp type for machine? [master | slave | client | none]: none

Enter root disk partition for MC68020 architecture (e.g xy0a) ? xy0a

/dev/rxy0a: 516 files, 2575 used, 4896 free (16 frags, 610 blocks)
/dev/rxy0g: 1678 files, 18810 used, 39733 free (69 frags, 4958 blocks)

Select optional software for the MC68020 architecture :
Do you want to install "Stand_diag"? [y/n]: y
Do you want to install "Usr_diag"? [y/n]: y

```

```

Do you want to install "Graphics"? [y/n]: y
Do you want to install "Uucp"? [y/n]: y
Do you want to install "System_V"? [y/n]: y
Do you want to install "Demo"? [y/n]: y
Do you want to install "Games"? [y/n]: y
Do you want to install "SunPro"? [y/n]: y
Do you want to install "Vttruff"? [y/n]: y

Are you ready to start the upgrade ? [y/n] : y

Changing directory to "/".

Extracting "root" files from "/dev/nrmt0" release tape.

Extracting "pub" files from "/dev/nrmt0" release tape.

Changing directory to "/usr".

Extracting "sys" files from "/dev/nrmt0" release tape.

Extracting "user," files from "/dev/nrmt0" release tape.

Extracting "Networking" files from "/dev/nrmt0" release tape.

Extracting "Debugging" files from "/dev/nrmt0" release tape.

.
.
.

Extracting "SunPro" files from "/dev/nrmt0" release tape.
Extracting "Vttruff" files from "/dev/nrmt0" release tape.

Checking filesystems :
/dev/rxy0a: 516 files, 2575 used, 4896 free (16 frags, 610 blocks)
/dev/rxy0g: 1678 files, 18810 used, 39733 free (69 frags, 4958 blocks)

3.4 to 3.5 upgrade completed.
Reboot your system and configure a kernel for your system.

```

Upgrade A Homogeneous Server With Local Tape Drive

Assume we have a server called godzilla with three Sun-3 clients : frodo, grendel and sofia. In this case, all 3.5 optional software exists on the disk.

```

>b mt ()
Boot: mt (, , 4)
From: mt (, , 5)
To: xy (, , 1)
Boot: xy (, , 1) vmunix -as
.

```

```
.
.
root device ? xy0*

.
.
.

#
# cd /usr/etc/upgrade
# UPGRADE

Enter system type ? [standalone | server]: server

Enter server type ? [homo | heter]: homo

Enter tape drive type ? [local | remote]: local

Enter tape type ? [ar | st | mt | xt]: mt

Enter yp type for machine? [master | slave | client | none]: slave

Enter root disk partition for MC68020 architecture (e.g. xy0a) ? xy0a

/dev/rxy0a: 516 files, 2575 used, 4896 free (16 frags, 610 blocks)
/dev/rxy0h: 1678 files, 18810 used, 39733 free (69 frags, 4958 blocks)
/dev/rxy0f: 83 files, 3495 used, 840 free (18 frags, 104 blocks)

Select optional software for the MC68020 architecture:
Do you want to install "System V"? [y/n]: y

Are you ready to start the upgrade ? [y/n] : y

Beginning 3.4 to 3.5 for the MC68020 architecture.

Changing directory to "/".

Extracting "root" files from "/dev/nrmt0" release tape.

Changing directory to "/pub".

Extracting "pub" files from "/dev/nrmt0" release tape.

Beginning 3.4 to 3.5 upgrade on MC68020 diskless clients.

Beginning 3.4 to 3.5 upgrade on client frodo.

Completed 3.4 to 3.5 upgrade on client frodo.

Beginning 3.4 to 3.5 upgrade on client grendel.

Completed 3.4 to 3.5 upgrade on client grendel.
```

```

Beginning 3.4 to 3.5 upgrade on client sofia.

Completed 3.4 to 3.5 upgrade on client sofia.

Changing directory to "/usr".

Extracting "sys" files from "/dev/nrmt0" release tape.

Extracting "user" files from "/dev/nrmt0" release tape.

Extracting "Networking" files from "/dev/nrmt0" release tape.

Extracting "Debugging" files from "/dev/nrmt0" release tape.
.
.
.
.
Extracting "SunPro" files from "/dev/nrmt0" release tape.
Extracting "Vtrotff" files from "/dev/nrmt0" release tape.

Checking filesystems :
/dev/rxy0a: 516 files, 2575 used, 4896 free (16 frags, 610 blocks)
/dev/rxy0h: 1678 files, 18810 used, 39733 free (69 frags, 4958 blocks)
/dev/rxy0f: 83 files, 3495 used, 840 free (18 frags, 104 blocks)

3.4 to 3.5 upgrade completed.
Reboot your system and configure a kernel for your system.
#

```

Upgrade A Homogeneous Server With Remote Tape Drive

Assume the host number of the system we are booting from is 114. 114 in decimal is equal to 72 in hexadecimal. We have a Sun-3 machine and the system is running 3.5. Therefore, the pub number is 1. In this case, none of the optional software exists on the disk.

```

>b ie(,72)boot -a
Boot: ie(,72)stand/copy
From: ie(,72,1)miniroot
To: xy(,,1)
Boot: xy(,,1)vmunix -as

.
.
.
root device ? xy0*

.
.
.
#

```

```
# cd /usr/etc/upgrade
# UPGRADE

Enter system type ? [standalone | server]: server

Enter server type ? [homo | heter]: homo

Enter tape drive type ? [local | remote]: remote

Enter host of remote drive ? pebbles

Enter Ethernet type of this system ? [ec | ie | le] : ie

Enter tape type ? [ar | st | mt | xt]: mt

Enter yp type for machine? [master | slave | client | none]: client

Enter root disk partition for MC68020 architecture (e.g. xy0a) ? xy0a

/dev/rxy0a: 516 files, 2575 used, 4896 free (16 frags, 610 blocks)
/dev/rxy0h: 1678 files, 18810 used, 39733 free (69 frags, 4958 blocks)
/dev/rxy0f: 83 files, 3495 used, 840 free (18 frags, 104 blocks)

Select optional software for the MC68020 architecture :
Do you want to install "Suntools_users"? [y/n]: y
Do you want to install "Suntools_programmers"? [y/n]: y
Do you want to install "Suntools_source"? [y/n]: y
Do you want to install "Stand_diag"? [y/n]: y
Do you want to install "Fortran"? [y/n]: y
Do you want to install "Usr_diag"? [y/n]: y
Do you want to install "Graphics"? [y/n]: y
Do you want to install "Pascal"? [y/n]: y
Do you want to install "Profiled"? [y/n]: y
Do you want to install "Uucp"? [y/n]: y
Do you want to install "System_V"? [y/n]: y
Do you want to install "Man"? [y/n]: y
Do you want to install "Demo"? [y/n]: y
Do you want to install "Games"? [y/n]: y
Do you want to install "SunPro"? [y/n]: y
Do you want to install "Vtroff"? [y/n]: y
Are you ready to start the upgrade ? [y/n] : y

Beginning 3.4 to 3.5 upgrade for the MC68020 architecture.

Changing directory to "/".

Extracting "root" files from "/dev/nrmt0" release tape.

Changing directory to "/pub".

Extracting "pub" files from "/dev/nrmt0" release tape.

Beginning 3.4 to 3.5 upgrade on MC68020 diskless clients.
```



```

Beginning 3.4 to 3.5 upgrade on client frodo.

Completed 3.4 to 3.5 upgrade on client frodo.

Beginning 3.4 to 3.5 upgrade on client grendel.

Completed 3.4 to 3.5 upgrade on client grendel.

Beginning 3.4 to 3.5 upgrade on client sofia.

Completed 3.4 to 3.5 upgrade on client sofia.

Changing directory to "/usr".

Extracting "sys" files from "/dev/nrmt0" release tape.

Extracting "user" files from "/dev/nrmt0" release tape.

Extracting "Networking" files from "/dev/nrmt0" release tape.

Extracting "Debugging" files from "/dev/nrmt0" release tape.

.
.
.

Extracting "SunPro" files from "/dev/nrmt0" release tape.
Extracting "Vtrotff" files from "/dev/nrmt0" release tape.

Checking filesystems :
/dev/rxy0a: 516 files, 2575 used, 4896 free (16 frags, 610 blocks)
/dev/rxy0h: 1678 files, 18810 used, 39733 free (69 frags, 4958 blocks)
/dev/rxy0f: 83 files, 3495 used, 840 free (18 frags, 104 blocks)

3.4 to 3.5 upgrade completed.
Reboot your system and configure a kernel for your system.
#

```

Upgrade A Heterogeneous Server With Local Tape Drive

Assume we have a server called godzilla with two Sun-3 clients (frodo and grendel) and one Sun-2 client (sofia) and the system is running 3.5. In this case we assume that none of the optional software from both architectures exists on the disk.

```

>b mt ()
Boot: mt (, , 4)
From: mt (, , 5)
To: xy (, , 1)
Boot: xy (, , 1) vmunix -as
.

```

```

.
.
root device ? xy0*

.
.
.

#
# cd /usr/etc/upgrade
# UPGRADE

Enter system type ? [standalone | server]: server

Enter server type ? [homo | heter]: heter

Enter tape drive type ? [local | remote]: local

Enter tape type ? [ar | st | mt | xt]: mt

Enter yp type for machine? [master | slave | client | none]: slave

Enter root disk partition for the MC68020 architecture (e.g. xy0a) ? xy0a

/dev/rxy0a: 516 files, 2575 used, 4896 free (16 frags, 610 blocks)
/dev/rxy0h: 1678 files, 18810 used, 39733 free (69 frags, 4958 blocks)
/dev/rxy0f: 83 files, 3495 used, 840 free (18 frags, 104 blocks)
/dev/rxy0g: 1163 files, 18150 used, 40393 free (73 frags, 5040 blocks)
/dev/rxy0e: 83 files, 3574 used, 905 free (17 frags, 111 blocks)

Select optional software for the MC68020 architecture :
Do you want to install "Suntools_users"? [y/n]: y
Do you want to install "Suntools_programmers"? [y/n]: y
Do you want to install "Suntools_source"? [y/n]: y
Do you want to install "Stand_diag"? [y/n]: y
Do you want to install "Fortran"? [y/n]: y
Do you want to install "Usr_diag"? [y/n]: y
Do you want to install "Graphics"? [y/n]: y
Do you want to install "Pascal"? [y/n]: y
Do you want to install "Profiled"? [y/n]: y
Do you want to install "Uucp"? [y/n]: y
Do you want to install "System_V"? [y/n]: y
Do you want to install "Man"? [y/n]: y
Do you want to install "Demo"? [y/n]: y
Do you want to install "Games"? [y/n]: y
Do you want to install "SunPro"? [y/n]: y
Do you want to install "Vtrotff"? [y/n]: y

Select optional software for the MC68010 architecture :
Do you want to install "Suntools_users"? [y/n]: y
Do you want to install "Suntools_programmers"? [y/n]: y
Do you want to install "Suntools_source"? [y/n]: y

```

```
Do you want to install "Stand_diag"? [y/n]: y
Do you want to install "Fortran"? [y/n]: y
Do you want to install "Usr_diag"? [y/n]: y
Do you want to install "Graphics"? [y/n]: y
Do you want to install "Pascal"? [y/n]: y
Do you want to install "Profiled"? [y/n]: y
Do you want to install "Uucp"? [y/n]: y
Do you want to install "System_V"? [y/n]: y
Do you want to install "Man"? [y/n]: y
Do you want to install "Demo"? [y/n]: y
Do you want to install "Games"? [y/n]: y
Do you want to install "SunPro"? [y/n]: y
Do you want to install "Vtroll"? [y/n]: y

Are you ready to start the upgrade ? [y/n] : y

Beginning 3.4 to 3.5 upgrade for the MC68020 architecture.

Changing directory to "/".

Extracting "root" files from "/dev/nrmt0" release tape.

Changing directory to "/pub".

Extracting "pub" files from "/dev/nrmt0" release tape.

Beginning 3.4 to 3.5 upgrade on MC68020 diskless clients.

Beginning 3.4 to 3.5 upgrade on client frodo.

Completed 3.4 to 3.5 upgrade on client frodo.

Beginning 3.4 to 3.5 upgrade on client grendel.

Completed 3.4 to 3.5 upgrade on client grendel.

Changing directory to "/usr".

Extracting "sys" files from "/dev/nrmt0" release tape.

Extracting "user" files from "/dev/nrmt0" release tape.

Extracting "Networking" files from "/dev/nrmt0" release tape.

Extracting "Debugging" files from "/dev/nrmt0" release tape.

.
.
.

Extracting "SunPro" files from "/dev/nrmt0" release tape.
Extracting "Vtroll" files from "/dev/nrmt0" release tape.
```

```

Beginning 3.4 to 3.5 upgrade for the MC68010 architecture.

Changing directory to "/pub.MC68010".

Extracting "pub" files from "/dev/nrmt0" release tape.

Beginning 3.4 to 3.5 upgrade on MC68010 diskless clients.

Beginning 3.4 to 3.5 upgrade on client sofia.

Completed 3.4 to 3.5 upgrade on client sofia.

Changing directory to "/usr.MC68010".

Extracting "sys" files from "/dev/nrmt0" release tape.

Extracting "user" files from "/dev/nrmt0" release tape.

Extracting "Networking" files from "/dev/nrmt0" release tape.

Extracting "Debugging" files from "/dev/nrmt0" release tape.

.
.
.

Extracting "SunPro" files from "/dev/nrmt0" release tape.
Extracting "Vtrotff" files from "/dev/nrmt0" release tape.

Checking filesystems :
/dev/rxy0a: 516 files, 2575 used, 4896 free (16 frags, 610 blocks)
/dev/rxy0h: 1678 files, 18810 used, 39733 free (69 frags, 4958 blocks)
/dev/rxy0f: 83 files, 3495 used, 840 free (18 frags, 104 blocks)
/dev/rxy0g: 1163 files, 18150 used, 40393 free (73 frags, 5040 blocks)
/dev/rxy0e: 83 files, 3574 used, 905 free (17 frags, 111 blocks)

3.4 to 3.5 upgrade completed.
Reboot your system and configure a kernel for your system.
#

```

Upgrade A Heterogeneous Server With Remote Tape Drive

Assume the host number of the system we are booting from is 114. 114 in decimal is equal to 72 in hexadecimal. We have a Sun-3 machine and the system is running 3.5. Therefore, the pub number is 1. In this case, we assume none of the optional software from both architectures exists on the disk.

```

>b ie(,72)boot -a
Boot: ie(,72)stand/copy
From: ie(,72,1)miniroot
To: xy(,,1)
Boot: xy(,,1)vmunix -as

```

```

.
.
.

root device ? xy0*

.
.
.

#
# cd /usr/etc/upgrade
# UPGRADE

Enter system type ? [standalone | server]: server

Enter server type ? [homo | heter]: heter

Enter tape drive type ? [local | remote]: remote

Enter host of remote drive ? pebbles

Enter Ethernet type of this system ? [ec | ie | le] : ie

Enter tape type ? [ar | st | mt | xt]: st

Enter yp type for machine? [master | slave | client | none]: slave

Enter root disk partition for the MC68020 architecture (e.g. xy0a) ? xy0a)

/dev/rxy0a: 516 files, 2575 used, 4896 free (16 frags, 610 blocks)
/dev/rxy0h: 1678 files, 18810 used, 39733 free (69 frags, 4958 blocks)
/dev/rxy0f: 83 files, 3495 used, 840 free (18 frags, 104 blocks)
/dev/rxy0g: 1163 files, 18150 used, 40393 free (73 frags, 5040 blocks)
/dev/rxy0e: 83 files, 3574 used, 905 free (17 frags, 111 blocks)

Select optional software for the MC68020 architecture :
Do you want to install "Suntools_users"? [y/n]: y
Do you want to install "Suntools_programmers"? [y/n]: y
Do you want to install "Suntools_source"? [y/n]: y
Do you want to install "Stand_diag"? [y/n]: y
Do you want to install "Fortran"? [y/n]: y
Do you want to install "Usr_diag"? [y/n]: y
Do you want to install "Graphics"? [y/n]: y
Do you want to install "Pascal"? [y/n]: y
Do you want to install "Profiled"? [y/n]: y
Do you want to install "Uucp"? [y/n]: y
Do you want to install "System_v"? [y/n]: y
Do you want to install "Man"? [y/n]: y
Do you want to install "Demo"? [y/n]: y
Do you want to install "Games"? [y/n]: y
Do you want to install "SunPro"? [y/n]: y
Do you want to install "Vtloff"? [y/n]: y

```

```
Select optional software for the MC68010 architecture :
Do you want to install "Suntools_users"? [y/n]: y
Do you want to install "Suntools_programmers"? [y/n]: y
Do you want to install "Suntools_source"? [y/n]: y
Do you want to install "Stand_diag"? [y/n]: y
Do you want to install "Fortran"? [y/n]: y
Do you want to install "Usr_diag"? [y/n]: y
Do you want to install "Graphics"? [y/n]: y
Do you want to install "Pascal"? [y/n]: y
Do you want to install "Profiled"? [y/n]: y
Do you want to install "Uucp"? [y/n]: y
Do you want to install "System_V"? [y/n]: y
Do you want to install "Man"? [y/n]: y
Do you want to install "Demo"? [y/n]: y
Do you want to install "Games"? [y/n]: y
Do you want to install "SunPro"? [y/n]: y
Do you want to install "Vtrotff"? [y/n]: y

Are you ready to start the upgrade ? [y/n] : y

Beginning 3.4 to 3.5 upgrade for the MC68020 architecture.

Changing directory to "/".

Extracting "root" files from "/dev/nrst0" release tape.

Changing directory to "/pub".

Extracting "pub" files from "/dev/nrst0" release tape.

Beginning 3.4 to 3.5 upgrade on MC68020 diskless clients.

Beginning 3.4 to 3.5 upgrade on client frodo.

Completed 3.4 to 3.5 upgrade on client frodo.

Beginning 3.4 to 3.5 upgrade on client grendel.

Completed 3.4 to 3.5 upgrade on client grendel.

Changing directory to "/usr".

Extracting "sys" files from "/dev/nrst0" release tape.

Extracting "user" files from "/dev/nrst0" release tape.

Extracting "Networking" files from "/dev/nrst0" release tape.

Extracting "Debugging" files from "/dev/nrst0" release tape.

.
.
.
```

```

Extracting "Vtproff" files from "/dev/nrst0" release tape.
Extracting "SunPro" files from "/dev/nrst0" release tape.

Beginning 3.4 to 3.5 upgrade for the MC68010 architecture.

Changing directory to "/pub.MC68010".

Extracting "pub" files from "/dev/nrst0" release tape.

Beginning 3.4 to 3.5 upgrade on MC68010 diskless clients.

Beginning 3.4 to 3.5 upgrade on client sofia.

Completed 3.4 to 3.5 upgrade on client sofia.

Changing directory to "/usr.MC68010".

Extracting "sys" files from "/dev/nrst0" release tape.

Extracting "user" files from "/dev/nrst0" release tape.

Extracting "Networking" files from "/dev/nrst0" release tape.

Extracting "Debugging" files from "/dev/nrst0" release tape.

.
.
.

Extracting "SunPro" files from "/dev/nrst0" release tape.
Extracting "Vtproff" files from "/dev/nrst0" release tape.

Checking filesystems :
/dev/rxy0a: 516 files, 2575 used, 4896 free (16 frags, 610 blocks)
/dev/rxy0h: 1678 files, 18810 used, 39733 free (69 frags, 4958 blocks)
/dev/rxy0f: 83 files, 3495 used, 840 free (18 frags, 104 blocks)
/dev/rxy0g: 1163 files, 18150 used, 40393 free (73 frags, 5040 blocks)
/dev/rxy0e: 83 files, 3574 used, 905 free (17 frags, 111 blocks)

3.4 to 3.5 upgrade completed.
Reboot your system and configure a kernel for your system.
#

```

2.4. How To Upgrade A Client Partition To 3.5

This section explains how you can upgrade a partition to 3.5 software that was not upgraded to 3.5 during the upgrade of the server. Here is a sample `/etc/nd.local` file:

```

#
# These lines added by the Sun Setup Program
#
clear
version 1
user 0 1 /dev/xy0f 0 11960 -1
user client1 0 /dev/xy0c 50600 10120 0
user client1 1 /dev/xy0c 60720 39560 -1
#user client2 0 /dev/xy0c 100280 10120 1
#user client2 1 /dev/xy0c 110400 39560 -1
user client3 0 /dev/xy0c 149960 10120 2
user client3 1 /dev/xy0c 160080 39560 -1
son
#
# End of lines added by the Sun Setup Program
#

```

The Steps to Upgrade Client2 to 3.5

Follow these steps to upgrade client2 to 3.5.

1. Make sure `/etc/hosts` on the server contains the Internet address of client2. You need to make sure `/etc/hosts` on the YP master contains the Internet address of client2 if you are running Yellow Pages.
2. Make sure `/etc/ethers` on the server contains the Ethernet address of client2. You need to make sure `/etc/ethers` on the YP master contains the Ethernet address of client2 if you are running Yellow Pages.
3. Fix `/etc/nd.local` by removing '#' from these two lines.

```

#user client2 0 /dev/xy0c 100280 10120 1
#user client2 1 /dev/xy0c 110400 39560 -1

```

4. Run `nd`.

```
server# /etc/nd < /etc/nd.local
```

5. Mount the client partition.

```
server# mount /dev/nd11 /mnt
```

6. Go to the client partition.

```
server# cd /mnt
```

7. Save all administrative files from the current release.


```

server# cp .login .login.save
server# cp .cshrc .cshrc.save
server# cp .rhosts .rhosts.save
server# cp etc/passwd etc/passwd.save
server# cp etc/printcap etc/printcap.save
server# cp etc/fstab etc/fstab.save
server# cp etc/hosts.equiv etc/hosts.equiv.save
server# cp etc/hosts etc/hosts.save
server# cp etc/rc.local etc/rc.local.save
server# cp etc/rc.boot etc/rc.boot.save
server# cp private/usr/lib/crontab private/usr/lib/crontab.save

```

8. Mount tape #1 of the release tape to the tape drive.
9. Position tape to the right file on the release tape.

```
server# mt -f /dev/nrtape0 fsf 8
```

Refer to *Table 2-3* for tape devices.

10. Extract files from the tape.

```

server# tar xvpf /dev/nrtape0
server# mt -f /dev/nrtape0 rew

```

11. Restore all administrative files by comparing the files saved and the new files and merge them by hand.
12. Get out of the client partition.

```
server# cd /
```

13. Unmount the client partition.

```
server# umount /dev/nd11
```

2.5. How To Restart The Upgrade Procedures

If for any reason UPGRADE terminated before it was completed, you can restart the upgrade procedure depending on the state of the system. Here are some suggestions. Please do not try these suggestions unless you fully understand the instructions.

1. If UPGRADE terminated before you saw this message on the console:

```
Extracting "root" files from "/dev/nrtape" release tape.
```

On the console, do the following:

2. Make sure you are still in the miniroot.

3. Make sure no file systems are mounted.

```
# cd /usr/etc/upgrade
```

```
# UPGRADE
```

4. If UPGRADE terminated after you saw this message

```
Extracting "root" files from "/dev/nrtape" release tape.
```

Do the following on the console:

5. Make sure you are still in the miniroot.
6. Make sure root and user file systems are mounted to /a and /a/usr.
7. Go through all files under /a, /a/etc and /a/private/usr/lib ending with ".save" and make sure the files contain correct information. If the files do not contain correct information and you do not know how to recover the files, you will not be able to continue with the upgrade procedure and you will have to use *Setup* to rebuild your system from scratch.
8. Make sure /a/usr/sys/conf/RELEASE contains the release number of the system before you started the upgrade (SunOS 3.2, 3.3, 3.4, 3.5BETA).
9. If your system is a server, you need to go through each client partition to make sure all files contain correct information for the partition.
10. If the files are correct, run these commands:

```
# cd usr/etc/upgrade
# adm_tool restore name machinetype yptype release
```

Usage: adm_tool op name machinetype yptype release
where:

op = save, restore or clean
name = name to be performed the operation
machinetype = standalone, server or diskless
yptype = master, slave, client or none
release = 3.4 or 3.5BETA

11. If the system is a server, go through each partition and do the following:

```
# mkdir /a/client
# mount /dev/nd1# /a/client
# cd /a/client
# /usr/etc/upgrade/adm_tool restore client_name diskless yptype release
# cd /
# umount /dev/nd1#
```

Replace '#' with a positive integer.. This step must be repeated for each client. See your /etc/nd.local file on your server.

12. Unmount root and user file systems.

13. Make sure no file systems are mounted, then enter:

```
# cd /usr/etc/upgrade
```

```
# UPGRADE
```

Reconfiguring Your Kernel

Reconfiguring Your Kernel	47
3.1. Procedures for Reconfiguring the Kernel	47
Kernel Reconfiguration for Standalone Systems	47
Kernel Reconfiguration for Servers	48
Kernel Reconfiguration for Clients	49
3.2. Kernel Reconfiguration — an Annotated Copy of a <i>GENERIC</i> File	51
3.3. Sun-2 <i>GENERIC</i> Configuration File	58
3.4. Sun-3 <i>GENERIC</i> Configuration File	60

Reconfiguring Your Kernel

This chapter explains how to reconfigure the kernel. You do not have to reconfigure your kernel because a *GENERIC* kernel, which supports every Sun configuration and peripheral, is provided on the upgrade tape. However, it is recommended that you reconfigure in order to save space, customize the kernel to recognize your hardware, or change *MAXUSERS* to a larger size to accommodate *Suntools*. You must reboot your system after completing the 3.5 upgrade, whether or not you reconfigure.

If you are doing kernel configuration for the first time, refer to the procedures in *Installing UNIX on the Sun Workstation* for Release 3.2.

If you have previously configured a kernel, you can use the next sections to guide you through reconfiguration. The first section gives reconfiguration procedures for all architectures. The second section contains an annotated copy of the new *GENERIC* kernel configuration file; read it carefully to make sure that you are including the correct device description lines for your system. The last two sections of this chapter contain specific *GENERIC* files for a Sun-2 or Sun-3 machine.

Note: See "General and Specific System Description Lines" sections of *Installing Unix on the Sun Workstation* for more detailed information concerning the annotated *GENERIC* file.

3.1. Procedures for Reconfiguring the Kernel

Kernel Reconfiguration for Standalone Systems

This section contains kernel reconfiguration procedures for all architecture types running Release 3.5.

For standalone machines, proceed as follows.

1. Change the current directory to `/usr/sys/conf`.

```
# cd /usr/sys/conf
```

2. Create a kernel configuration file.

Copy the file *GENERIC*. In the following examples, the new kernel configuration file is called *SYS_NAME* (representing the name of the system).

```
# cp GENERIC SYS_NAME
# chmod +w SYS_NAME
```

3. Edit `/usr/sys/conf/SYS_NAME` to reflect your system configuration and comment out those lines that do not apply. Use the annotated copy of

GENERIC provided in the following section for an explanation of entries in the configuration file. Make sure that you are including the proper device description lines for your system.

4. While still in the `/usr/sys/conf` directory, run `/etc/config` for the new kernel configuration file. Then change to the new configuration directory created by `/etc/config`, as shown below.

```
# /etc/config SYS_NAME
# cd ../SYS_NAME
# make
[ lots of output ]
```

5. Now you can save your old kernel and install the new one as follows:

```
# mv /vmunix /vmunix.old
# cp /usr/sys/conf/SYS_NAME/vmunix /vmunix
# /etc/shutdown -h now
    The system goes through the halt sequence, then
    the monitor displays its prompt, at which point you
    can boot the new kernel:
>b vmunix
```

6. If the system appears to work, this completes the upgrade procedure. If the new kernel does not seem to function properly, boot `/vmunix.old`, as shown below, copy it back to `/vmunix`, and go about fixing your new kernel.

Note: It is very useful to save the old kernel until the new one is working smoothly, and in case you need it for diagnostic purposes.

```
# /etc/shutdown -h now
>b vmunix.old -s [ boot the system in single user mode ]
# mv /vmunix /vmunix.oops
# mv /vmunix.old /vmunix
# ^D [ Brings the system up multiuser ]
```

Kernel Reconfiguration for Servers

For server machines, proceed as follows.

1. Change the current directory to `/usr/sys/conf`.

```
# cd /usr/sys/conf
```

2. Create a kernel configuration file.

Copy the file *GENERIC*. The new kernel configuration file is called *SYS_NAME* (representing the name of the system). For example,

```
# cp GENERIC SYS_NAME
# chmod +w SYS_NAME
```


3. Edit `/usr/sys/conf/SYS_NAME` to reflect your system configuration commenting out the lines that do not apply. Use the annotated copy of *GENERIC* provided in the next section for an explanation. Be sure to include the proper device description lines for your system.
4. Still in the `/usr/sys/conf` directory, run `/etc/config` for the new kernel configuration file. Then, change to the new configuration directory, and make the new system.

```
# /etc/config SYS_NAME
# cd ../SYS_NAME
# make
[ lots of output ]
```

5. Now you can go to the directory containing the server's kernel, save your server's old kernel, install your new one, and try everything out as follows:

```
# cd /usr/sys/SYS_NAME
# mv /vmunix /vmunix.old
# cp /usr/sys/SYS_NAME vmunix /vmunix
```

6. Since at this point normal system performance is a highly, but not absolutely, certain indicator of a trouble-free kernel, if your system(s) appears to work, you may proceed with some confidence. You have successfully completed installation.

If, on the other hand, any new kernels do not seem to function properly, halt all systems, and boot from the original kernel. Then, move the faulty kernel away, and reinstall the original in its place. Once you are booted up on the original, you can try to fix the faulty kernel. For example, try the following on the server:

```
# /etc/shutdown
>b vmunix.old -s
# cd /
# mv vmunix vmunix.bad
# mv vmunix.old vmunix
# ^D [ Brings the system up multi-user ]
```

Kernel Reconfiguration for Clients

1. Now, prepare a kernel for your clients in the same way. When editing the kernel configuration file (*CLIENT_SYS_NAME* in the following example), remember to include the entire set of devices used by client machines. For example, substitute 68010 (for Sun-2s) or 68020 (for Sun-3s) for *client_arch* in the commands below.

```
# cd /usr.MCclient_arch/sys/conf
# cp GENERIC_CLIENT_SYS_NAME
# chmod +w CLIENT_SYS_NAME
[ Edit CLIENT_SYS_NAME to reflect all clients' systems.
  Be especially careful with the device description lines. ]
# /etc/config CLIENT_SYS_NAME
# cd ../CLIENT_SYS_NAME
# make
[ lots of output ]
```

- Next, install the appropriate client kernel in /pub for the architecture. To install the clients' kernel, **make sure all the clients are halted**; save the original kernel (if there is one); install the new kernel image in the appropriate /pub; then test it out by booting up one of the clients.

```
# cd /usr.MCclient_arch/sys/CLIENT_SYS_NAME
# cp vmunix /pub.client_arch/vmunix
```

On the client machine, enter

```
>b vmunix
```

- If, at this point, any of the new kernels do not seem to be functioning properly, halt all the clients on the server. You have to correct the problem from the server.

Go back to the server and enter the following:

```
# cd /pub.MCclient_arch
# mv vmunix vmunix.bad
# mv vmunix.old vmunix
```

You can now boot up the clients and allow them to run while, or until, a new client kernel is made and ready to install. Or, if the clients can remain down, build and install a new client kernel now.

3.2. Kernel Reconfiguration — an Annotated Copy of a GENERIC File

Note: For the specific GENERIC files for a Sun-2 or Sun-3 machine, see Section 3.3 (Sun-2) or Section 3.4 (Sun-3) of this chapter.

The next page shows the *GENERIC* configuration file to help you identify the lines you need to include in your own system configuration file.

The comments explain the device and pseudo-device lines, and may also refer you to the reference manual entry that covers the device in question. If the comments say the line is **mandatory**, the line *must* be included in every system configuration file, either exactly as it stands, or, if commentary indicates variables, with the variables adjusted to fit your system.

A number of parameters relating to the System V Inter-Process Communication (IPC) extensions may also be tuned in the configuration file. These parameters do not appear in the GENERIC file but are documented in the System V Overview.

<i>Configuration Line</i>	<i>Comments</i>	<i>Description</i>
#		
# GENERIC SUN-3		
#		
machine "sun3"	mandatory	Identifies the specific machine.
cpu "SUN3_160"	mandatory*	Identifies CPU type (Sun-3/160, Sun-3/180, Sun-3/140, Sun-3/75).
cpu "SUN3_50"	mandatory*	Identifies the specific CPU type.
cpu "SUN3_260"	mandatory*	Identifies the specific CPU type (Sun-3/260 or Sun-3/280).
cpu "SUN3_110"	mandatory*	Identifies the specific CPU type (Sun-3/110).
cpu "SUN3_60"	mandatory*	Identifies the specific CPU type (Sun-3/60).
cpu "SUN3_E"	mandatory*	Identifies the specific CPU type (Sun-3/E)
ident GENERIC	mandatory	See <i>General</i> and <i>Specific System Description Lines</i> for information. Finally, if <i>SYS_NAME</i> contains both alpha and numeric characters (for example, SDST120), you must enclose the name in double quotes ("SDST120"), or you will get a syntax error when you run <code>/etc/config</code> .
timezone 8 dst	mandatory	Specifies your time zone and adjusts values accordingly. It can also use half hour designations.
maxusers 4	mandatory	Number may vary. For most systems, "4" is the proper value for maxusers. See the section, <i>General System Description Lines</i> , for information.
options INET	mandatory	Controls inclusion of Internet code — see <code>inet</code> (4). You must also include the "pseudo-device loop" lines below.
options SYSACCT	<i>optional</i>	Controls inclusion of code to do process accounting — see <code>acct</code> (2) and <code>acct</code> (5).
options QUOTA	<i>optional</i>	Controls the disk quota checking system.
options NFS	<i>optional</i>	Inclusion of NFS code.
options NIT	<i>optional</i>	Inclusion of network interface tap code.
options IPCMESSAGE	<i>optional</i>	Controls inclusion of code for SystemV IPC Message Facility.
options IPCSEMAPHORE	<i>optional</i>	Controls inclusion of code for SystemV IPC Semaphore Facility.
options IPCSHMEM	<i>optional</i>	Controls inclusion of code for SystemV IPC Shared-Memory Facility.
config vmunix swap generic	mandatory	Specifies kernel name and configuration clauses. Please see <i>Specific System Description Lines</i> for information.
pseudo-device ptty	<i>optional</i>	Pseudo-tty's. These are needed for network or window system.
pseudo-device bk	<i>optional</i>	Berknet line discipline for high speed tty input — see <code>bk</code> (4).
pseudo-device ether	<i>optional</i>	ARP code. Must include if using Ethernet — see <code>arp</code> (4).
pseudo-device loop	mandatory	Software loop back network device driver — see <code>lo</code> (4). Must include with 'options INET'.
pseudo-device nd	<i>optional</i>	Network disk. Necessary for servers and diskless clients, and for machines serving as remote hosts for remote installation — see <code>nd</code> (4).
pseudo-device win128	<i>optional</i>	Window system. Number indicates maximum number of windows. If you include this line, you must also include the "pseudo-device dtop," "ms," and "kb" lines just below.
pseudo-device dtop4	<i>optional</i>	Maximum number of screens ("desktops"). Required for window system.
pseudo-device ms3	<i>optional</i>	Maximum number of mice. Required for window system — see <code>ms</code> (4).
pseudo-device kb3	<i>optional</i>	Maximum number of Sun keyboards. Required if using any Sun keyboard, and for the window system.
pseudo-device mcpa64	<i>optional</i>	

Maximum number of asynchronous serial lines supported by the Sun ALM-2. If you include this line, you must also include the `mcp n` lines below.

** You need not include all machine types, only the machine type(s) that you may be running.*

The following are connections for machine types. These connections, in conjunction with controllers, devices, and disks for a structure, enable your system to recognize various hardware and software attached to it. For each device or controller on a bus, you need to have the bus type it is connected to listed under connections for machine type. It easiest to leave all lines for machine types that way. As you add controllers and devices, the connections are already in place and will be recognized by your system.

connections for machine type 1 (SUN3_160)

```
controller    virtual 1 at nexus ?
controller    obmem 1 at nexus ?
controller    obio 1 at nexus ?
controller    vme16d16 1 at nexus ?
controller    vme24d16 1 at nexus ?
controller    vme32d16 1 at nexus ?
controller    vme16d32 1 at nexus ?
controller    vme24d32 1 at nexus ?
controller    vme32d32 1 at nexus ?
```

connections for machine type 2 (SUN3_50)

```
controller    virtual 2 at nexus ?
controller    obmem 2 at nexus ?
controller    obio 2 at nexus ?
```

connections for machine type 3 (SUN3_260)

```
controller    virtual 3 at nexus ?
controller    obmem 3 at nexus ?
controller    obio 3 at nexus ?
controller    vme16d16 3 at nexus ?
controller    vme24d16 3 at nexus ?
controller    vme32d16 3 at nexus ?
controller    vme16d32 3 at nexus ?
controller    vme24d32 3 at nexus ?
controller    vme32d32 3 at nexus ?
```

connections for machine type 4 (SUN3_110)

```
controller    virtual 4 at nexus ?
```

```

controller      obmem 4 at nexus ?
controller      obio 4 at nexus ?
controller      vme16d16 4 at nexus ?
controller      vme24d16 4 at nexus ?
controller      vme32d16 4 at nexus ?
controller      vme16d32 4 at nexus ?
controller      vme24d32 4 at nexus ?
controller      vme32d32 4 at nexus ?

```

connections for machine type 7 (SUN3_60)

```

controller      virtual 7 at nexus ?
controller      obmem 7 at nexus ?
controller      obio 7 at nexus ?

```

connections for machine type 8 (SUN3_E)

```

controller      virtual 8 at nexus ?
controller      obmen 8 at nexus ?
controller      obio 8 at nexus ?
controller      vme16d16 8 at nexus ?
controller      vme24d16 8 at nexus ?
controller      vme32d16 8 at nexus ?
controller      vme16d32 8 at nexus ?
controller      vme24d32 8 at nexus ?
controller      vme32d32 8 at nexus ?

```

The following are controllers and devices (devices, disks, and tapes) that connect to bus types. Bus types and devices must hang off the appropriate controller, which, in turn, hangs off another controller until a configuration is formed that gets you to a bus type that hangs off a nexus. On Sun workstations, all bus types are considered to hang off a nexus. For example, disk

```
xy0 at xyc0 drive 0
```

hangs off controller

```
xyc0 at vme16d16 ? csr 0xee40 priority 2 vector xyintr 0x48
```

which hangs off bus type

```
controller      vme16d16 1 at nexus ?
```

In order to determine and note what devices are present on your machine, boot the GENERIC kernel after you have executed the 3.5UPGRADE. If you want, you can delete those lines that pertain to devices not on your machine. Or, you can configure your file with devices on other machines that may want to boot from the same kernel.

Do not remove the `zs` lines, which represent UARTS, from the configuration file. If they are removed, the system does not recognize the presence of the keyboard, mouse, or serial ports.

`xy` is the Xylogics 450/451 SMD disk controller. `xy` refers to any SMD disk attached to this controller.

```
controller  xyc0 at vme16d16 ? csr 0xee40 priority 2 vector xyintr 0x48
controller  xyc1 at vme16d16 ? csr 0xee48 priority 2 vector xyintr 0x49
disk        xy0 at xyc0 drive 0
disk        xy1 at xyc0 drive 1
disk        xy2 at xyc1 drive 0
disk        xy3 at xyc1 drive 1
```

`sc` is the SCSI-2 host adaptor. `sd` refers to any SCSI hard disk, and `st` refers to any SCSI cartridge tape. `sf` refers to any SCSI floppy disk drive attached to either of these controllers. `si` is the SCSI-3 host adaptor, and `se` is the 3/E SCSI interface on the 3/E SCSI/Ethernet board.

```
controller  sc0 at vme24d16 ? csr 0x200000 priority 2 vector scintr 0x40
disk        sd0 at sc0 drive 0 flags 0
disk        sd1 at sc0 drive 1 flags 0
tape        st0 at sc0 drive 32 flags 1
disk        sd2 at sc0 drive 8 flags 0
tape        st1 at sc0 drive 40 flags 1
#disk       sf0 at sc0 drive 8 flags 2
controller  si0 at vme24d16 ? csr 0x200000 priority 2 vector siintr 0x40
controller  si0 at obio ? csr 0x140000 priority 2
disk        sd0 at si0 drive 0 flags 0
disk        sd1 at si0 drive 1 flags 0
tape        st0 at si0 drive 32 flags 1
disk        sd2 at si0 drive 8 flags 0
tape        st1 at si0 drive 40 flags 1
#disk       sf0 at si0 drive 8 flags 2
controller  se0 at vme24d16 ? csr 0x300000 priority 2 vector se_intr 0x40
disk        sd0 at se0 drive 0 flags 0
disk        sd1 at se0 drive 1 flags 0
tape        st0 at se0 drive 32 flags 1
disk        sd2 at se0 drive 8 flags 0
tape        st1 at se0 drive 40 flags 1
```

`zs` refers to the onboard serial ports. `zs0` controls serial ports A and B, and `zs1` provides the keyboard and mouse interface. These lines should never be

removed.

```
device    zs0 at obio ? csr 0x20000 flags 3 priority 3
device    zs1 at obio ? csr 0x00000 flags 0x103 priority 3
```

mti is the Systech ALM terminal multiplexer board.

```
device    mti0 at vme16d16 ? csr 0x620 flags 0xffff priority 4
vector    mtiintr 0x88
device    mti1 at vme16d16 ? csr 0x640 flags 0xffff priority 4
vector    mtiintr 0x89
device    mti2 at vme16d16 ? csr 0x660 flags 0xffff priority 4
vector    mtiintr 0x8a
device    mti3 at vme16d16 ? csr 0x680 flags 0xffff priority 4
vector    mtiintr 0x8b
```

mcp is the ALM-2 terminal multiplexer board.

```
device    mcp0 at vme32d32 ? csr 0x01000000 flags 0x1ffff priority 4
vector    mcpintr 0x8b
device    mcp1 at vme32d32 ? csr 0x01010000 flags 0x1ffff priority 4
vector    mcpintr 0x8a
device    mcp2 at vme32d32 ? csr 0x01020000 flags 0x1ffff priority 4
vector    mcpintr 0x89
device    mcp3 at vme32d32 ? csr 0x01030000 flags 0x1ffff priority 4
vector    mcpintr 0x88
```

ie is the Intel Ethernet interface found on all machines except the 3/50 and 3/60. ie0 is the interface on the CPU board and the 3/E SCSI Ethernet board. ie1 is the optional second interface used on a gateway machine, and le0 is the Lance Ethernet found on the 3/50 and 3/60.

```
device    ie0 at obio ? csr 0xc0000 priority 3
device    ie0 at vme24d16 ? csr 0x31ff02 priority 3 vector ieintr 0x74
device    ie1 at vme24d16 ? csr 0xe88000 priority 3 vector ieintr 0x75
device    le0 at obio ? csr 0x120000 priority 3
```

tm is the CPC Tapemaster 1600 BPI 1/2" 9-track tape controller. mt refers to any 9-track drive attached to this controller.


```

controller  tm0 at vme16d16 ? csr 0xa0 priority 3 vector tmintr 0x60
controller  tm1 at vme16d16 ? csr 0xa2 priority 3 vector tmintr 0x61
tape        mt0 at tm0 drive 0 flags 1
tape        mt1 at tm1 drive 0 flags 1

```

`xtc` is the Xylogics 472 6250 BPI 1/2" 9-track tape controller. `xt` refers to any 9-track drive attached to this controller.

```

controller  xtc0 at vme16d16 ? csr 0xee60 priority 3 vector xtintr 0x64
controller  xtc1 at vme16d16 ? csr 0xee68 priority 3 vector xtintr 0x65
tape        xt0 at xtc0 drive 0 flags 1
tape        xt1 at xtc1 drive 0 flags 1

```

`gpone` is the Sun Graphics Processor. `cgtwo` is the Sun-3 Color frame buffer found on deskside Sun workstations. `cgfour` is the Sun-3 Color frame buffer found on 3/60 and 3/110 desktop workstations. `bwtwo` is the monochrome frame buffer.

```

device      gpone0 at vme24d16 ? csr 0x210000
device      gpone0 at vme24d32 ? csr 0x240000
device      cgtwo0 at vme24d16 ? csr 0x400000 priority 4
            vector cgtwointr 0xa8
device      cgfour0 at obmem 4 csr 0xff000000 priority 4
device      cgfour0 at obmem 7 csr 0xff300000 priority 4
device      cgfour0 at obmem 7 csr 0xff400000 priority 4
device      bwtwo0 at obmem 1 csr 0xff000000 priority 4
device      bwtwo0 at obmem 2 csr 0x100000 priority 4
device      bwtwo0 at obmem 3 csr 0xff000000 priority 4
device      bwtwo0 at obmem 4 csr 0xff000000
device      bwtwo0 at obmem 7 csr 0xff000000 priority 4
device      bwtwo0 at obmem 8 csr 0x1000000
device      bwtwo1 at obmem 7 csr 0xff300000 priority 4
device      bwtwo1 at obmem 7 csr 0xff400000

```

`vpc` is the Systech VPC-2200 Versatec printer/plotter interface.

```

device      vpc0 at vme16d16 ? csr 0x480 priority 2 vector vpcintr 0x80
device      vpc1 at vme16d16 ? csr 0x500 priority 2 vector vpcintr 0x81

```

`des` is the hardware encryption chip. `fpa` is the Sun Floating Point Accelerator.

```

device      des0 at obio ? csr 0x1c0000
device      fpa0 at virtual ? csr 0xe0000000

```

3.3. Sun-2 GENERIC Configuration File

The following is the GENERIC configuration file for a Sun-2 system.

```
#
# GENERIC SUN2
#
machine      "sun2"
cpu          "SUN2_120"      # generic for machine type 1 (Multibus)
cpu          "SUN2_50"      # generic for machine type 2 (VMEbus)
ident        GENERIC
timezone     8 dst
maxusers     4
options      INET
options      SYSACCT
options      QUOTA
options      NFS
options      NIT
options      IPCMESSAGE # SystemV IPC Message Facility
options      IPCSEMAPHORE # SystemV IPC Semaphore Facility
options      IPCSHMEM # SystemV IPC Shared-Memory Facility

config       vmunix      swap generic

pseudo-device  pty
pseudo-device  bk
pseudo-device  ether
pseudo-device  loop
pseudo-device  nd
pseudo-device  win128
pseudo-device  dtop4
pseudo-device  ms3
pseudo-device  kb3

# connections for machine type 1 (SUN2_120)
controller virtual 1 at nexus ? # virtual preset
controller obmem 1 at nexus ? # on board memory
controller obio 1 at nexus ? # on board io
controller mbmem 1 at nexus ? # Multibus memory
controller mbio 1 at nexus ? # Multibus io

# connections for machine type 2 (SUN2_50)
controller virtual 2 at nexus ? # virtual preset
controller obmem 2 at nexus ? # on board memory
controller obio 2 at nexus ? # on board io
controller vme16 2 at nexus ? # 16 bit address VMEbus (16 bit data)
controller vme24 2 at nexus ? # 24 bit address VMEbus (16 bit data)

controller ipc0 at mbio ? csr 0x40 priority 2
controller ipc1 at mbio ? csr 0x44 priority 2
disk ip0 at ipc0 drive 0
disk ip1 at ipc0 drive 1
disk ip2 at ipc1 drive 0
disk ip3 at ipc1 drive 1
```

```

controller xyc0 at mbio ? csr 0xee40 priority 2
controller xyc0 at vme16 ? csr 0xee40 priority 2 vector xyintr 0x48
controller xyc1 at mbio ? csr 0xee48 priority 2
controller xyc1 at vme16 ? csr 0xee48 priority 2 vector xyintr 0x49
disk xy0 at xyc0 drive 0
disk xy1 at xyc0 drive 1
disk xy2 at xyc1 drive 0
disk xy3 at xyc1 drive 1
controller sc0 at mbmem ? csr 0x80000 priority 2
controller sc0 at vme24 ? csr 0x200000 priority 2 vector scintr 0x40
disk sd0 at sc0 drive 0 flags 0
disk sd1 at sc0 drive 1 flags 0
tape st0 at sc0 drive 32 flags 1
disk sd2 at sc0 drive 8 flags 0
tape st1 at sc0 drive 40 flags 1
#disk sf0 at sc0 drive 8 flags 2
controller scl1 at mbmem ? csr 0x84000 priority 2
disk sd2 at scl1 drive 0 flags 0
disk sd3 at scl1 drive 1 flags 0
tape st1 at scl1 drive 32 flags 1
#disk sf1 at scl1 drive 8 flags 2
device sky0 at mbio ? csr 0x2000 priority 2
device sky0 at vme16 ? csr 0x8000 priority 2 vector skyintr 0xb0
device zs0 at obio 1 csr 0x2000 flags 3 priority 3
device zs0 at obio 2 csr 0x7f2000 flags 3 priority 3
device zs1 at obmem 1 csr 0x780000 flags 0x103 priority 3
device zs1 at obio 2 csr 0x7f1800 flags 0x103 priority 3
device zs2 at mbmem ? csr 0x80800 flags 3 priority 3
device zs3 at mbmem ? csr 0x81000 flags 3 priority 3
device zs4 at mbmem ? csr 0x84800 flags 3 priority 3
device zs5 at mbmem ? csr 0x85000 flags 3 priority 3
device mti0 at mbio ? csr 0x620 flags 0xffff priority 4
device mti1 at mbio ? csr 0x640 flags 0xffff priority 4
device mti2 at mbio ? csr 0x660 flags 0xffff priority 4
device mti3 at mbio ? csr 0x680 flags 0xffff priority 4
device mti0 at vme16 ? csr 0x620 flags 0xffff priority 4
vector mtiintr 0x88
device mti1 at vme16 ? csr 0x640 flags 0xffff priority 4
vector mtiintr 0x89
device mti2 at vme16 ? csr 0x660 flags 0xffff priority 4
vector mtiintr 0x8a
device mti3 at vme16 ? csr 0x680 flags 0xffff priority 4
vector mtiintr 0x8b
device ie0 at obio 2 csr 0x7f0800 priority 3
device ie0 at mbmem ? csr 0x88000 priority 3
device iel1 at mbmem ? csr 0x8c000 flags 2 priority 3
device iel1 at vme24 ? csr 0xe88000 priority 3 vector ieintr 0x75
device ec0 at mbmem ? csr 0xe0000 priority 3
device ecl1 at mbmem ? csr 0xe2000 priority 3
controller tm0 at mbio ? csr 0xa0 priority 3
controller tm0 at vme16 ? csr 0xa0 priority 3 vector tmintr 0x60
controller tm1 at mbio ? csr 0xa2 priority 3
controller tm1 at vme16 ? csr 0xa2 priority 3 vector tmintr 0x61

```

```

tape      mt0 at tm0 drive 0 flags 1
tape      mt1 at tm1 drive 0 flags 1
controller xtc0 at mbio ? csr 0xee60 priority 3
controller xtc0 at vme16 ? csr 0xee60 priority 3 vector xtintr 0x64
controller xtc1 at mbio ? csr 0xee68 priority 3
controller xtc1 at vme16 ? csr 0xee68 priority 3 vector xtintr 0x65
tape      xt0 at xtc0 drive 0 flags 1
tape      xt1 at xtc1 drive 0 flags 1
device    ar0 at mbio ? csr 0x200 priority 3
device    ar1 at mbio ? csr 0x208 priority 3
device    gpone0 at vme24 ? csr 0x210000 priority 4 vector gponeintr 0xac
device    cgtwo0 at vme24 ? csr 0x400000 priority 4 vector cgtwointr 0xa8
device    cgone0 at mbmem ? csr 0xec000 priority 3
device    bwtwo0 at obmem 1 csr 0x700000 priority 4
device    bwtwo0 at obio 2 csr 0x0 priority 4
device    bwone0 at mbmem ? csr 0xc0000 priority 3
device    vp0 at mbio ? csr 0x400 priority 2
device    vpc0 at mbio ? csr 0x480 priority 2
device    vpc0 at vme16 ? csr 0x480 priority 2 vector vpcintr 0x80
device    vpc1 at mbio ? csr 0x500 priority 2
device    vpc1 at vme16 ? csr 0x500 priority 2 vector vpcintr 0x81
device    pi0 at obio 1 csr 0x1800
device    des0 at obio 1 csr 0x1000
device    des0 at obio 2 csr 0x7f1000
device    tod0 at obio 1 csr 0x3800
device    tod0 at vme24 ? csr 0x200800

```

3.4. Sun-3 GENERIC Configuration File

The following is the GENERIC configuration file for a Sun-3 system.

```

#
# GENERIC SUN3
#
machine    "sun3"
cpu        "SUN3_160"      # (Sun-3/160 or Sun-3/75 cpu)
cpu        "SUN3_50"
cpu        "SUN3_260"
cpu        "SUN3_110"
cpu        "SUN3_60"
cpu        "SUN3_E"
ident      GENERIC
timezone   8 dst
maxusers   4
options    INET
options    SYSACCT
options    QUOTA
options    NFS
options    NIT
options    IPCMESSAGE      # SystemV IPC Message Facility
options    IPCSEMAPHORE    # SystemV IPC Semaphore Facility

```

```

options          IPCSHMEM          # SystemV IPC Shared-Memory Facility

config          vmunix          swap generic

pseudo-device   pty
pseudo-device   bk
pseudo-device   ether
pseudo-device   loop
pseudo-device   nd
pseudo-device   win128
pseudo-device   dtop4
pseudo-device   ms3
pseudo-device   kb3
pseudo-device   mcpa64

# connections for machine type 1 (SUN3_160)
controller      virtual 1 at nexus ?
controller      obmem 1 at nexus ?
controller      obio 1 at nexus ?
controller      vme16d16 1 at nexus ?
controller      vme24d16 1 at nexus ?
controller      vme32d16 1 at nexus ?
controller      vme16d32 1 at nexus ?
controller      vme24d32 1 at nexus ?
controller      vme32d32 1 at nexus ?

# connections for machine type 2 (SUN3_50)
controller      virtual 2 at nexus ?
controller      obmem 2 at nexus ?
controller      obio 2 at nexus ?

# connections for machine type 3 (SUN3_260)
controller      virtual 3 at nexus ?
controller      obmem 3 at nexus ?
controller      obio 3 at nexus ?
controller      vme16d16 3 at nexus ?
controller      vme24d16 3 at nexus ?
controller      vme32d16 3 at nexus ?
controller      vme16d32 3 at nexus ?
controller      vme24d32 3 at nexus ?
controller      vme32d32 3 at nexus ?

# connections for machine type 4 (SUN3_110)
controller      virtual 4 at nexus ?
controller      obmem 4 at nexus ?
controller      obio 4 at nexus ?
controller      vme16d16 4 at nexus ?
controller      vme24d16 4 at nexus ?
controller      vme32d16 4 at nexus ?
controller      vme16d32 4 at nexus ?
controller      vme24d32 4 at nexus ?
controller      vme32d32 4 at nexus ?

```

```

# connections for machine type 7 (SUN3_60)
controller    virtual 7 at nexus ?
controller    obmem 7 at nexus ?
controller    obio 7 at nexus ?

# connections for machine type 8 (SUN3_E)
controller    virtual 8 at nexus ?
controller    obmem 8 at nexus ?
controller    obio 8 at nexus ?
controller    vme16d16 8 at nexus ?
controller    vme24d16 8 at nexus ?
controller    vme32d16 8 at nexus ?
controller    vme16d32 8 at nexus ?
controller    vme24d32 8 at nexus ?
controller    vme32d32 8 at nexus ?

controller    xyc0 at vme16d16 ? csr 0xee40 priority 2 vector xyintr 0x48
controller    xyc1 at vme16d16 ? csr 0xee48 priority 2 vector xyintr 0x49
disk          xy0 at xyc0 drive 0
disk          xy1 at xyc0 drive 1
disk          xy2 at xyc1 drive 0
disk          xy3 at xyc1 drive 1
controller    sc0 at vme24d16 ? csr 0x200000 priority 2 vector scintr 0x40
disk          sd0 at sc0 drive 0 flags 0
disk          sd1 at sc0 drive 1 flags 0
disk          sd2 at sc0 drive 8 flags 0
#disk        sd3 at sc0 drive 9 flags 0
tape          st0 at sc0 drive 32 flags 1
tape          st1 at sc0 drive 40 flags 1
#disk        sf0 at sc0 drive 8 flags 2
controller    si0 at vme24d16 ? csr 0x200000 priority 2 vector siintr 0x40
controller    si0 at obio ? csr 0x140000 priority 2
disk          sd0 at si0 drive 0 flags 0
disk          sd1 at si0 drive 1 flags 0
disk          sd2 at si0 drive 8 flags 0
disk          sd3 at si0 drive 9 flags 0
tape          st0 at si0 drive 32 flags 1
tape          st1 at si0 drive 40 flags 1
#disk        sf0 at si0 drive 8 flags 2
controller    se0 at vme24d16 ? csr 0x300000 priority 2 vector se_intr 0x40
disk          sd0 at se0 drive 0 flags 0
disk          sd1 at se0 drive 1 flags 0
disk          sd2 at se0 drive 8 flags 0
disk          sd3 at se0 drive 9 flags 0
tape          st0 at se0 drive 32 flags 1
tape          st1 at se0 drive 40 flags 1
device        zs0 at obio ? csr 0x20000 flags 3 priority 3
device        zs1 at obio ? csr 0x00000 flags 0x103 priority 3
device        mti0 at vme16d16 ? csr 0x620 flags 0xffff priority 4
              vector mtiintr 0x88
device        mti1 at vme16d16 ? csr 0x640 flags 0xffff priority 4
              vector mtiintr 0x89
device        mti2 at vme16d16 ? csr 0x660 flags 0xffff priority 4

```

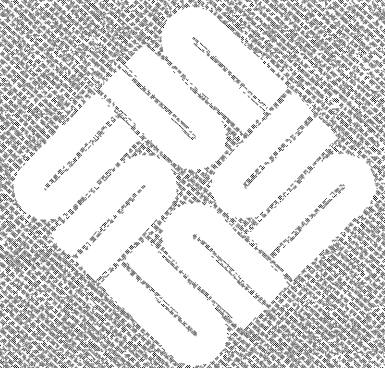
```

vector mtiintr 0x8a
device      mti3 at vme16d16 ? csr 0x680 flags 0xffff priority 4
vector mtiintr 0x8b
device      mcp0 at vme32d32 ? csr 0x01000000 flags 0x1ffff priority 4
vector mcpintr 0x8b
device      mcp1 at vme32d32 ? csr 0x01010000 flags 0x1ffff priority 4
vector mcpintr 0x8a
device      mcp2 at vme32d32 ? csr 0x01020000 flags 0x1ffff priority 4
vector mcpintr 0x89
device      mcp3 at vme32d32 ? csr 0x01030000 flags 0x1ffff priority 4
vector mcpintr 0x88
device      ie0 at obio ? csr 0xc0000 priority 3
device      ie0 at vme24d16 ? csr 0x31ff02 priority 3 vector ieintr 0x74
device      ie1 at vme24d16 ? csr 0xe88000 priority 3 vector ieintr 0x75
device      le0 at obio ? csr 0x120000 priority 3
controller  tm0 at vme16d16 ? csr 0xa0 priority 3 vector tmintr 0x60
controller  tm1 at vme16d16 ? csr 0xa2 priority 3 vector tmintr 0x61
tape        mt0 at tm0 drive 0 flags 1
tape        mt1 at tm1 drive 0 flags 1
controller  xtc0 at vme16d16 ? csr 0xee60 priority 3 vector xtintr 0x64
controller  xtc1 at vme16d16 ? csr 0xee68 priority 3 vector xtintr 0x65
tape        xt0 at xtc0 drive 0 flags 1
tape        xt1 at xtc1 drive 0 flags 1
device      gpone0 at vme24d16 ? csr 0x210000 priority 4
vector gponeintr 0xac
device      cgtwo0 at vme24d16 ? csr 0x400000 priority 4
vector cgtwointr 0xa8
device      cgfour0 at obmem 4 csr 0xff000000 priority 4
device      cgfour0 at obmem 7 csr 0xff300000 priority 4
device      cgfour0 at obmem 7 csr 0xff400000 priority 4
device      bwtwo0 at obmem 1 csr 0xff000000 priority 4
device      bwtwo0 at obmem 2 csr 0x100000 priority 4
device      bwtwo0 at obmem 3 csr 0xff000000 priority 4
device      bwtwo0 at obmem 4 csr 0xff000000
device      bwtwo0 at obmem 7 csr 0xff000000 priority 4
device      bwtwo0 at obmem 8 csr 0x1000000 priority 4
device      bwtwo0 at obmem 7 csr 0xff000000 priority 4
device      bwtwo1 at obmem 7 csr 0xff300000 priority 4
device      bwtwo1 at obmem 7 csr 0xff400000
device      vpc0 at vme16d16 ? csr 0x480 priority 2 vector vpcintr 0x80
device      vpc1 at vme16d16 ? csr 0x500 priority 2 vector vpcintr 0x81
device      des0 at obio ? csr 0x1c0000
device      fpa0 at virtual ? csr 0xe0000000

```


Changes To Hardware and Software in Release 3.5

Changes To Hardware and Software in Release 3.5	67
4.1. New Hardware Functionality	67
Sun-3 Eurocard	67
Sun-3/60	67
ALM-2	67
380MB SCSI Drive	68
1GB SMD Drive	68
4.2. Graphics Enhancements	68
4.3. SunView Enhancements	68
Double-Buffering Support in Release 3.5	68
How Double-Buffering Works	68
Prototypical Application Code	69
Pixwin Interface	69
pw_dbl_access() :	69
pw_lock()	69
pw_unlock()	70
pw_dbl_flip()	70
pw_dbl_release()	70
pw_dbl_set()	70
pw_dbl_get()	71
Pixrect Interface	71
pr_dbl_get()	71
pr_dbl_set()	72



Changes To Hardware and Software in Release 3.5

4.1. New Hardware Functionality

Release 3.5 provides software support for the following new hardware:

- Sun-3 Eurocard
- Sun-3/60
- ALM-2
- 380MB SCSI Drive
- 1GB SMD Drive

Sun-3 Eurocard

The Sun-3 Eurocard is a board set that implement the Sun-3 workstation architecture on standard, double-height VMEbus boards. This board set consists of a CPU board, a monochrome video display controller board, and a SCSI/Ethernet board.

Sun-3/60

The Sun-3/60 is a desk-top, high-performance workstation that supports color, greyscale, high-resolution monochrome, and standard monochrome displays and up to 24 MBytes of memory on a single board. The Sun-3/60 was supported on SunOS™ 3.4, but was not bootable directly from the 3.4 tapes.

ALM-2

The Asynchronous Line Multiplexer, Model 2, (ALM-2) is a combination of hardware and software that increases the communication capacity and performance of the Sun-3 desktop and cabinet-mounted workstations and file servers by off-loading the low-level communication tasks to a specialized serial communication board. Each ALM-2 supports up to 16 serial communication lines and one Centronics-compatible parallel printer port. When combined with the Sun programming environment, the ALM-2 offers a powerful and flexible vehicle for the implementation of multiuser and timesharing computer systems.

380MB SCSI Drive

The Micropolis 1558 disk drive offers 327MB of formatted storage and 380 MB of unformatted storage in the form of a SCSI device that is addressed by the `sd` driver. The disk is form factor compatible to the industry standard 5 1/4" disks. The drive uses the ESDI interface, and the disk is a direct drop-in replacement for the 141MB disk.

1GB SMD Drive

The NEC D2363 and Hitachi DK815-10 disk drives offer 900MB of formatted storage, and 1GB of unformatted storage in the form of an SMD device that is addressed by the `xy` driver. This disk comes in a nine inch rack mount form factor.

4.2. Graphics Enhancements

Release 3.5 contains support for hardware double-buffering. Both the *SunView* and *Pixrect* libraries contain additional routines for double-buffering. For an explanation of these routines, see *Pixrect Interface* in the section below.

4.3. SunView Enhancements

The next section describes this support.

Double-Buffering Support in Release 3.5

NOTE *Pre-3.5 binaries running on a 3.5-based system with other double-buffering applications may not behave correctly. For example, when you move or resize a window, it may not repaint correctly; or when you try to bring up your SunView menu, it may have disappeared. You must rebuild these binaries on 3.5 or link them with 3.5.*

Double-buffering allows one image to be displayed while another is being drawn. It can be simulated by manipulating the colormap (see *Double Buffering* in Section 7.5, *Color*, in the SunView Programmer's Guide), or done in hardware on frame buffers that support it. Please note that simulating it severely restricts the number of colors available. The following routines will allow programs to do true hardware double-buffering on the CG5(4S)¹ color frame buffer and on future frame buffers that support double-buffering.

How Double-Buffering Works

Double-buffering is treated as an even scarcer resource than color maps, since only one window can be truly double-buffered at any one time. The cursor controls which window will flip the display buffers. `pw_lock()` calls (as needed) the *pixrect* routines that set the hardware buffer control bits. Applications are able to run the same code on non-double-buffered displays and it will be as if the double-buffering calls were never made.

¹ The CG5 board is binary compatible with both the Sun-3 Color Board and the Sun-2 Color Board. CG5 is necessary for hardware double-buffering. In the interim, a variant of the Sun-3 Color Board is used.

Prototypical Application Code

```

rect    rectangle;
Pixwin  *pw;
    rectangle.r_left= ...;
    ...
    if (!pw_dbl_get (pw, PW_DBL_AVAIL))
        { ... if program cares ... }
    pw_dbl_access (pw);
    while (rendering_frames) {
        ... calculate one frame ...
        pw_lock (pw, &rectangle);
        ... render one frame ...
        ... may include unlocks and locks ...
        pw_dbl_flip (pw);
        pw_unlock (pw);
    }
    pw_dbl_release (pw);

```

Pixwin Interface

The notion of the “active” double-buffering window is important. There is at most one active window at a time. If a double-buffering window has the cursor in it, then it is the active double-buffering window. If the cursor leaves the active window, that window remains active until the cursor enters another double-buffering window. If the active double-buffering window dies, goes iconic, or becomes totally obscured, and the cursor is not left in a double-buffering window, then the top-most visible double-buffering window becomes the active window (if there is one). Only the active window will be allowed to write to a single buffer. The notion of active will change only during a `pw_dbl_flip()` call.

`pw_dbl_access()`

```

pw_dbl_access (pw)
    Pixwin *pw;

```

The window associated with the pixwin is marked as being a double-buffering window, and so is the desktop (the screen on which the window resides). If one window of a pixwin is double-buffering, then any other pixwin of that window is implicitly double-buffering.

`pw_dbl_access()` resets the window’s data structure so that first frame will be rendered to the background. The very first double buffer sets both READ and WRITE to the background. `pw_dbl_access()` should only be called when ready to actively animate.

`pw_lock()`

```

pw_lock(pw, rectangle)
    Pixwin *pw;
    rect *rectangle;

```

Do a `pw_lock()` as before.

If the pixwin's window has not been accessed for double-buffering then there is no change, that is to say both buffers will be written to.

If the window is marked as accessible for double-buffering and the window is "active" (see above) then set the frame double-buffering control to whatever this window requested with its last `pw_dbl_set` call. If there was no `pw_dbl_set` call, then set `WRITE` and `READ` to the background. Change the frame buffer double-buffering control bits to reflect this.

`pw_unlock()`

```
pw_unlock(pw)
    Pixwin *pw;
```

Unchanged. In particular, it does not set the write control bits through the `pixrect` driver.

`pw_dbl_flip()`

```
pw_dbl_flip(pw)
    Pixwin *pw;
```

If the pixwin's window has not been accessed for double-buffering do nothing.

If the window is accessible for double-buffering then potentially flip the display. The display is flipped only if the window is "active" (see above).

`pw_dbl_flip()` determines if its window has become active.

The flip can be done inside or outside of a lock region although it may be preferable to place inside a lock region just before an unlock so that calculations for the next frame can proceed even if another window momentarily grabs the lock. Flip also syncs with vertical retrace

`pw_dbl_release()`

```
pw_dbl_release(pw)
    Pixwin *pw;
```

Signifies the end of double-buffering by the window associated with the pixwin. `pw_dbl_release()` should be called as soon as a program has completed a section of active animation. `pw_dbl_release()` will copy the foreground to the background. Therefore it is important to leave the animation loop after a `pw_dbl_flip()` has been done and before drawing the next frame has started. Otherwise, the window will contain a partial frame after the release.

`pw_dbl_set()`

```
pw_dbl_set(pw, attribute_list)
    struct pixwin *pw;
    attribute_list;    /* Zero terminated. */
```

An attribute list consists of pairs of attributes and values. The list is terminated by a zero.

Table 4-1 *Pixwin-Level set Attributes*

<i>Attribute</i>	<i>Possible Values to set</i>
PW_DBL_WRITE	PW_DBL_FORE, PW_DBL_BACK, PW_DBL_BOTH
PW_DBL_READ	PW_DBL_FORE, PW_DBL_BACK

SunView provides the ability for an actively double-buffering window to write to both buffers so that instrument gauge readings can be set in a real-time simulator (as an example). If `pw` is not the active double buffer, the frame buffer control bits are not changed

`pw_dbl_get()`

```
pw_dbl_get(pw, attribute)
struct pixwin *pw;
attribute;
```

Table 4-2 *Pixwin-Level get Attributes*

<i>Attribute</i>	<i>Possible Values Returned</i>
PW_DBL_AVAIL	PW_DBL_EXISTS
PW_DBL_DISPLAY	
PW_DBL_WRITE	PW_DBL_FORE, PW_DBL_BACK, PW_DBL_BOTH
PW_DBL_READ	PW_DBL_FORE, PW_DBL_BACK

The attribute value returned from `pw_dbl_get()` does not reflect the true state of double buffering hardware. This is especially true if the active double buffer is not this `pixwin`.

Pixrect Interface

NOTE

It is not recommended that you mix `pw_dbl_set()` and `pw_dbl*` as the programs may not behave correctly. You should try to use `pw_dbl_set()` instead.

`pr_dbl_get()`

```
pr_dbl_get(pr, attribute)
struct pixrect *pr;
attribute;
```

Table 4-3 *Pixrect-Level get Attributes*

<i>Attribute</i>	<i>Possible Values Returned</i>
PR_DBL_AVAIL	PR_DBL_EXISTS
PR_DBL_DISPLAY	PR_DBL_A, PR_DBL_B
PR_DBL_WRITE	PR_DBL_A, PR_DBL_B, PR_DBL_BOTH, PR_DBL_NONE
PR_DBL_READ	PR_DBL_A, PR_DBL_B

pr_dbl_set()

```
pr_dbl_set(pr, attribute_list)
    struct pixrect *pr;
    attribute_list;    /* Zero terminated. */
```

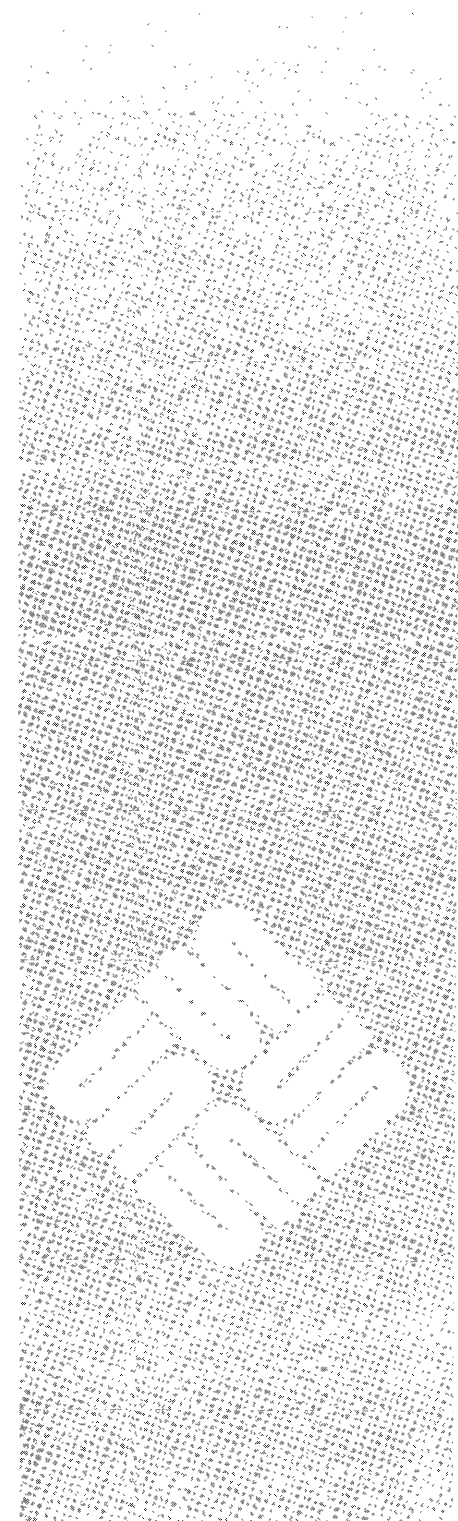
An attribute list consists of pairs of attributes and values. The list is terminated by a zero. If the display is already in the state requested, return. If PR_DBL_DISPLAY is in the list, then the process may block for up to a single video frame's time (15 ms). PR_DBL_DISPLAY_DONTBLOCK is a set() only attribute.

Table 4-4 *Pixrect-Level set Attributes*

<i>Attribute</i>	<i>Possible Values to Set</i>
PR_DBL_WRITE	PR_DBL_A, PR_DBL_B, PR_DBL_BOTH
PR_DBL_READ	PR_DBL_A, PR_DBL_B
PR_DBL_DISPLAY_DONTBLOCK	PR_DBL_A, _B
PR_DBL_DISPLAY	PR_DBL_A, _B

Bugfixes Since Release 3.4

Bugfixes Since Release 3.4	75
5.1. Graphics Bugs Fixed	75
<i>SunCore</i> Bug Fixes	75
Text not Clipping	75
<i>Pirect</i> Bugs Fixed	75
pr_polypoint on GP may Mess Up the First Point	75
pr_load Misbehaves in Case of Error	75
<i>SunCGI</i> Bugs Fixed	76
Bad Name Argument	76
close_cgi_pw() Fails	76
Visible Mouse Cursor	76
Compile <i>CGI</i> Program	76
<i>Graphics Processor</i> Bug Fixes	76
PR_PGON_TEX does not Clip Texture to Screen Correctly	77
Incorrect GP1_PR_ROP_TEX Semantics	77
GP1_PR_PGON_TEX Problem	77
5.2. SunOS Bugs Fixed	77
Utility Bugs Fixed	77
2.59 sysdiag Disables Disk Testing in Systems with GP/GB	77
rasfilter8to1.1g Has Wrong Filename Suffix	77
sysdiag Assumes You are Always on the Console	77
sysdiag sptest Fails	77
make does not Handle Square Bracket Characters	77



look Dumps Core on Long Lines	77
SCCS <code>deledit</code> Duplicates Random Lines in a File	78
SCCS Uses Delta Times for Diffs	78
Problem with Sendmail	78
Core Dump	78
Duplicate Lines	78
Commands Ignored	78
SunPro Files not Installed	78
<code>make</code> did not Build	78
Printer Daemon Disappears	78
Undocumented <code>fsck</code> Error Message	79
Termcap Entry Breaks	79
5.3. Kernel Bugs Fixed	79
Kernel Bus Error on Sirius	79
Shell Script Crashes 3/200 Series Kernel	79
Colormap Updates Fail	79
System Flooded with <code>zs</code> Interrupts	79
Sysdiag Core Dump	80
Sysdiag Fills Root Partition	80
Physio Bug Causes <code>writev(2V)</code> Failure	80
Physio Breaks <code>readv</code>	80
SCSI MD21 (141 MB) Errors Cause System Hang	80
System Panics When Using <code>ttya</code> with SCSI-3	80
SCSI Disk Driver Hangs	80
UNIX Hangs When Booting if Tape Drive On-line	80
Demo Programs Fail with a Core Dump	81
<code>diag</code> Causes Error Message	81
<code>ptrace</code> Interaction with Interrupting Slow System Calls	81
Misspelled Error Message	81
<code>cgtwo</code> Driver Watchdogs	81
Booting from 1/2 Inch Tape Hangs 3/260 System	81
Device Driver Panic	81
Setting Raw Mode Panics System	81
Error Handling Inconsistent	82
<code>mti</code> Error is Cryptic	82

TOD Chip Failure	82
sys pt too small message	82
Incorrect Comment	82
AF_UNIX SOCK_STREAM OOB	82
-ffpa Kernel Bus Error	82
Sun-2 SCSI Driver Bug	83
Redebugging Crashes System	83
5.4. Network Bugs Fixed	83
Network Routing Deamon in.routed Dies Periodically	83
Routing Daemon in.routed Sometimes Uses Wrong Interface	83
rcp Broadcast Address Computes Incorrectly	83
Bad ICMP Address Mask Reply	83
TCP/IP File Transfer Hangs	83
NFS Attribute Cache Functions Incorrectly	83
flocf () Very Slow	83
if.c Changes to Support MCP and Subnets	84
TCP Performance Problem	84
RSTAT (3R) Returns 0	84
Incorrect ping Response	84
Wrong Loopback Address	84
rexid Race Condition	84
5.5. System Administration Bugs Fixed	84
rpc.lockd Fails to Free	84
Symbolic Links not Installed	84
5.6. SunView Bugs Fixed	84
pw_line () Draws in Window Space Instead of Canvas Space	84
pw_line () Draws Incorrectly When pw_batch_on ().....	85
Problem with pw_line and pw_polyline	85
tektool: Menu Missing Items and no Longer in Walking Style	85
tektool Key Does Not Work	85
tektool Could Core Dump	85
Crosshair Cursors not Working	85

Screenblank Too Effective	85
5.7. Compiler Bugs Fixed	86
Optimizer Substitution Error	86
dbx Shows Segmentation Violation	86
Recursive #includes Confuse dbx	86
ld -A Produces Bogus Symbol Tables	86
printf Padding Strings with Leading Zero's are Broken	86
dbx Failure	86

Bugfixes Since Release 3.4

This chapter describes the bug fixes provided for Release 3.5. The bug fix descriptions are organized into the following sections:

- Graphics
- SunOS™
- Kernel
- Network
- System Administration
- SunView
- Compiler

5.1. Graphics Bugs Fixed

This section describes the bug fixes in the *SunCore*, *SunCGI*, and *Pixrect* graphics software, and *Graphics Processor (GP)* microcode.

SunCore Bug Fixes

The following *SunCore* bug is fixed for Release 3.5.

Text not Clipping

Transforming a segment that contains text now works properly. Previously, when doing a transformation of a segment that contained text, the text came onto the screen although it should have been clipped out of the picture.

Pixrect Bugs Fixed

The following *Pixrect* bugs are fixed for Release 3.5.

`pr_polypoint` on GP may Mess Up the First Point

`gp1_polypoint` did not initialize a rasterop chip register properly, so the first pixel drawn may have been incorrect. This is fixed for Release 3.5.

`pr_load` Misbehaves in Case of Error

The subroutine `pr_load_image` in `pr_io.h` contains these lines of code:

```

case RT_STANDARD:
    if (fread(image, 1, imagesize, input) == imagesize)
```

```

        return pr;

case RT_BYTE_ENCODED:
    if (read_encoded(input, rh->ras_length,
        (u_char *) image, imagesize) == 0)
        return pr;

default:

```

If the `fread` fails in the case `RT_STANDARD`, the code now returns the correct error.

SunCGI Bugs Fixed

The following *SunCGI* bugs are fixed for Release 3.5.

Bad Name Argument

`inquire_cell_array` and `inquire_pixel_array` have an argument, *name*, that now specifies which view surface to use. Previous releases used this as a one-based index into a name table. In Release 3.5, this is corrected to use a zero-based index similar to the other functions in the library.

`close_cgi_pw()` Fails

If a `close_cgi_pw()` call was done with no view surface active, the following message appeared:

```
WIN ioctl number 406C6713: Inappropriate ioctl for device
```

This problem often occurred when using *SunCGI* with canvases. The canvas would hang, although the parent process could still be interrupted. This is fixed for Release 3.5.

Visible Mouse Cursor

When a CGI application was run outside of the window system, an arrow cursor appeared in the middle of the screen. Since mouse tracking was not enabled, the cursor image could not be moved out of the screen. This is fixed for Release 3.5.

Compile CGI Program

The 3.4Beta *CGI* library required the use of the `suntools` library in order to resolve the function name `window_get()`. This library is no longer required in Release 3.5.

Graphics Processor Bug Fixes

The following *Graphics Processor* bugs are fixed for Release 3.5.

`PR_PGON_TEX` does not Clip Texture to Screen Correctly

When a textured polygon is clipped to the left hand side of the screen, the texture offset now advances correctly.

Incorrect `GP1_PR_ROP_TEX` Semantics

`GP1_PR_ROP_TEX` from a 1 bit deep source pixrect now has the correct semantics.

`GP1_PR_PGON_TEX` Problem

When `GP1_PR_PGON_TEX` displays polygons, the polygons no longer draw incorrectly (on the GP) when a portion of the polygon intersects the left edge of the screen (not the window).

5.2. SunOS Bugs Fixed

This section describes the bug fixes in the SunOS.

Utility Bugs Fixed

The following utility bugs are fixed in Release 3.5.

2.59 `sysdiag` Disables Disk Testing in Systems with GP/GB

Version 2.59 `sysdiag` no longer disables the testing of disks in systems with GP/GB present.

`rasfilter8tol.1g` Has Wrong Filename Suffix

`rasfilter8tol` was moved from section 1G to section 1, and now builds correctly.

`sysdiag` Assumes You are Always on the Console

One of the options for `sysdiag` changes the terminal type. Selecting this option caused `sysdiag` to change the terminal type for the console, regardless of whether you were on the console or not. `sysdiag` then exited and you had to start it up again. This is fixed for 3.5.

`sysdiag sptest` Fails

`sptest` in `sysdiag` no longer fails across all products. The test will no longer timeout while waiting for received data during loopback tests.

`make` does not Handle Square Bracket Characters

With older versions of `make`, bracket characters [] could be included in target filenames. The latest version of `make` did not permit this use. The Release 3.5 version of `make` now permits this use.

`look` Dumps Core on Long Lines

`look` declared its internal working buffers to be 250 characters long. If incoming lines are longer than that, `look` no longer overwrites storage, which led to incorrect results, including core dumps.

**SCCS `deledit` Duplicates
Random Lines in a File**

After doing an `SCCS delget` on a file, the new version of the file no longer has adjacent lines duplicated.

**SCCS Uses Delta Times for
Diffs**

If one person sets their clock into the future, checks out a file, edits it, and checks it back in, then another person with the correct date checks out, edits, and checks in the file, SCCS no longer adds duplicate lines to the file.

Problem with Sendmail

Sendmail no longer has a problem when a machine has two names, or when mail is sent to a name that is not the first name in the hosts database.

Core Dump

The `overwrite` routine in 4.3 `-lcurses` no longer core dumps in certain circumstances such as when a smaller window overwrites another window.

**Duplicate Lines
Commands Ignored**

When the `!!` command in `vi` was used to execute a sub-process that returned a line longer than `vi` could handle, the following error message appeared on the screen:

```
line too long
```

After the message appeared, any commands entered were echoed to the screen, and `vi` ignored those commands until a new line was typed in. This is now fixed.

SunPro Files not Installed

The `install_sunproscript` (under `/usr/sunpro`) no longer fails to install the SunPro files. The script now contains code to install a standalone, and no longer fails on servers.

make did not Build

The 3.2 version of `make` will now build specified objects when those objects depend on another object that has no build lines in that make file.

Printer Daemon Disappears

When a customized line printer sent `lpr` output on an HP Laserjet™, followed by a raster file to the printer, then followed by an ASCII text file, (for example, `/etc/termcap`) followed by removing the raster file job with `lprm`, and finally performing an `lpq`, this message appeared:

```
no daemon present
```

Printing did not appear until `lpr` was run again. This problem did not occur on 2.0 systems, or when the file being removed was a straight text file. This is now fixed for Release 3.5.

Undocumented `fsck` Error Message

The `fsck` error message

```
HOLD BAD BLOCK
```

was undocumented. From the error message, it was not clear why it occurred, or what action to take. This error message has now been fixed to read:

```
BAD MODE: MAKE IT A FILE?
```

Termcap Entry Breaks

Calling `initscr()` in `libcurses` no longer causes the shell to respond with the following error message:

```
Memory fault - core dumped
```

and `dbx` no longer reports a segmentation violation in `_doprnt()`. This message occurred only when `TERM=wy` or `TERM=wyse-50` shared the same entry.

5.3. Kernel Bugs Fixed

This section describes the bug fixes in the kernel.

Kernel Bus Error on Sirius

Running a load average of ~100 on a 16 mb `sirius` no longer produces a kernel bus error.

Shell Script Crashes 3/200 Series Kernel

Running the Sun-2 3.0 version of the `ksh` on a Sun 3/200 series machine caused the system to crash with the message:

```
panic: writeback error
```

This is fixed for Release 3.5.

Colormap Updates Fail

On the Sun-3/60, the `cgfour` driver now handles colormap updates correctly if the update index is not 0, 1, 2, or 3.

System Flooded with `zs` Interrupts

The `synca` and `syncb` pins of the keyboard/mouse `scc` (Zilog 8530 serial communications controller, `zs`) are now connected. Under severe stress testing, noise created transitions on those lines that flooded the system with interrupts.

Sysdiag Core Dump

`sysdiag's softfp` and `mc68881` tests no longer core dump with illegal instructions under severe stress testing.

Sysdiag Fills Root Partition

Sysdiag put its log into `/usr2`, which on most systems is in the root partition. The root partition had very little space and the sysdiag log files can be very large and often filled the file the file system. Sysdiag is now configurable, so space is no longer a problem.

**Physio Bug Causes
writev(2V) Failure**

`sun/sys/sys/vm_swp.c:physio()` no longer increments the wrong `iovec` pointer, which caused the `writev(2V)` system call to fail when the third argument (`iovcnt`) was greater than one. `writev` only transferred the data from the area in memory specified in the first `iovec` structure, and returned the length specified in the first `iovec` structure.

Physio Breaks readv

`readv()` only worked on block devices, apparently because of a bug where `physio` did not increment the `iov` pointer each time it did a transfer, and therefore it only noticed the first vector, and was incapable of doing a scattered read. This bug is fixed for Release 3.5.

**SCSI MD21 (141 MB) Errors
Cause System Hang**

Some Emulex MD21 SCSI disk controller hard (not able to be corrected by controller) errors are now handled properly, no longer causing a SCSI subsystem hang.

**System Panics When Using
ttya with SCSI-3**

System no longer gets a

```
panic: Bus error
```

while using `ttya` when running with a configured kernel on a system with a SCSI-3(beta) host adapter.

SCSI Disk Driver Hangs

When a write fault occurred, (or another write-related error that the controller cannot correct occurred), the driver tried to retry the command but did not set up the DMA properly. The SCSI bus hung in `DATA_OUT` phase. This appeared as a hang in the SCSI subsystem. This bug is fixed for Release 3.5.

**UNIX Hangs When Booting if
Tape Drive On-line**

On a Sun 3/160, if you attempted to boot UNIX while the tape drive was on-line (with the Xylogics 472), the system printed out the last probe for the SCSI devices (`st1`) and then hung, apparently while probing `xtc0`. This is fixed for Release 3.5.

Demo Programs Fail with a Core Dump

When a partition was larger than 512 megabytes, programs whose inodes were past the 512 megabyte point, no longer core dump with a segmentation violation.

diag Causes Error Message

When using `diag` to perform long reads or writes with a SCSI-3 host adaptor, typing `(Control) (C)` no longer causes confusing error messages.

ptrace Interaction with Interrupting Slow System Calls

When a slow system call such as `select` or `read` from a terminal is interrupted while a program is under the control of a debugger, the debugger is now able to call a function in the program being debugged.

Misspelled Error Message

Error message:

```
Priviledge violation
```

in `trap.c` now reads:

```
Privilege violation
```

`cgtwo` Driver Watchdogs

When a `cgtwo` interrupt vector was not specified in the kernel configuration file, the resulting kernel would watchdog in `cgtwoattach()` when booted. This bug is now fixed.

Booting from 1/2 Inch Tape Hangs 3/260 System

Booting from 1/2 inch tape (Tapemaster controller and CDC tape drive) on a 3/260, no longer hangs the system.

Device Driver Panic

The serial port device driver would make direct calls to `tty` line discipline routines for the old/new `tty` discipline (`sys/tty.h`) without checking to ensure that the line in question was actually running one of those disciplines.

If the port was using a line discipline (such as Berknet or the consulting streamlined high-speed input discipline) where the raw or canonical input queues may not be clists, this would cause kernel panics. This bug is fixed for Release 3.5.

Setting Raw Mode Panics System

Setting raw mode after you enabled the `bk(4)` line discipline no longer causes the system to panic with the message:

```
panic: bus error
```

Error Handling Inconsistent

SCSI tape drives behaved inconsistently. Different things happened when different types of SCSI tape drives were opened for writing and the tape drive was write protected or contained no cartridge. This is now fixed.

mti Error is Cryptic

An error message given by the Multiple Terminal Interface driver, `mti.c`, was cryptic. The message read:

```
mti<number>: error <error number>
```

The error conditions that produced it indicated either hardware problems or driver bugs -- neither of which was indicated by the error message.

TOD Chip Failure

The 3.x kernel can now handle failure of the Time of Day chip on the 3/50.

sys pt too small message

Installing the 3.2 `movc.o` kernel patch on a 3.3 machine caused the boot to fail with a

```
sys pt too small
```

error message. This is fixed for Release 3.5.

Incorrect Comment

In `/usr/include/sys/buf.h`, the line

```
long b_resid; /* words not transferred after error */
```

was incorrect. bytes not words are transferred after the error.

AF_UNIX SOCK_STREAM OOB

AF_UNIX SOCK_STREAM OOB data is now a supported feature of SunOS 3.5.

-ffpa Kernel Bus Error

A FORTRAN program compiled with `-ffpa` produced the following error messages:

```
FPA KERNEL BUS ERROR
FPA BUS ERROR: IERR == 20efff
```

This bug is fixed for Release 3.5.

Sun-2 SCSI Driver Bug

The Emulex SCSI tape controller and the DocuPro page scanner did not work together correctly on the SCSI bus when used with the Sun-2 SCSI board. This has been fixed for Release 3.5.

Redebugging Crashes System

When you attempted to re-debug a program that ran under `dbx`, that you interrupted while it was opening a named pipe, the following error occurred:

```
panic: bus error
```

This bug is fixed for Release 3.5.

5.4. Network Bugs Fixed

This section describes the bug fixes in the network.

**Network Routing Daemon
`in.routed` Dies Periodically**

The network routing daemon `in.routed` no longer dies periodically on Sun machines.

**Routing Daemon `in.routed`
Sometimes Uses Wrong
Interface
`rpc` Broadcast Address
Computes Incorrectly**

The routing daemon `in.routed` no longer uses the incorrect interface.

In the `rpc` routine that does broadcast `rpc` requests, the broadcast address now computes correctly when subnetting is enabled. Previously, broadcast requests caused problems on the Ethernets and did not deliver the packet where it should go.

Bad ICMP Address Mask Reply

A vendor's software had a bug that returned netmasks with the incorrect byte order. This caused net booting in SunOS 3.3 and 3.4 to have severe problems, essentially making the network unusable. This bug is fixed for Release 3.5

TCP/IP File Transfer Hangs

Negative initial sequence numbers no longer cause TCP connections to hang. As long as initial sequence numbers remained positive, no problems were observed, but at seemingly random times, FTP connections, for example, would time out.

**NFS Attribute Cache Functions
Incorrectly**

During periods of heavy NFS operations, the attributes of a file in use with the NFS server mounts no longer get out of sync.

`flock()` Very Slow

`lockf()`, which uses the lock manager in 3.2 and later releases, no longer functions very slowly. This happened even when the network was not involved. This is fixed for Release 3.5.

if.c Changes to Support MCP and Subnets

There is no longer a conflict between the changes made to `if.c` to support the subnets and the ones required to support the multiple MCP devices.

TCP Performance Problem

TCP performance has been improved.

RSTAT (3R) Returns 0

The RPC system call `rstat(3R)` no longer takes as one of its arguments, a pointer to a structure of type `statstime(rpcsvc/rstat.h)`, and no longer fills the structure after the return.

Incorrect ping Response

`ping` responded that a machine was up whenever the machine performing the `ping` got an `icmp` packet, regardless of where the `icmp` came from.

Wrong Loopback Address

The loopback address used in the name server resolver is incorrect. Lookups outside of the current domain would hang, even when `/usr/etc/in.named` was running.

rexdb Race Condition

Under certain conditions, the RPC remote execution daemon `rpc.rexd` left directories in `/tmp`, instead of removing them.

5.5. System Administration Bugs Fixed

This section describes the bug fixes for system administration.

rpc.lockd Fails to Free

The `rpc.lockd` and `rpc.statd` daemons now free RPC CLIENT that it created through `clntudp_create()`. Previously, when the daemon called `clntdup_create()` repeatedly, it used excess memory and swap area.

Symbolic Links not Installed

The symbolic links `/usr/bin/{clear_colormap,rasfilter8tol,rastrepl,screenload}` were not present on a server upgraded to 3.4. They are included for Release 3.5.

5.6. SunView Bugs Fixed

This section describes the bugs fixed in SunView.

pw_line() Draws in Window Space Instead of Canvas Space

The `pixwin` function `pw_line()` no longer draws in window space instead of canvas space. The sister function `pw_vector()` correctly drew in canvas space.

**pw_line() Draws Incorrectly
When pw_batch_on()**

If you turned `pw_batch_on()` and used `pw_line` to draw a line greater than 1 pixel in width, the line was drawn incorrectly. The width of the displayed line was sometimes incorrect and the ends of the line appeared clipped if the line was drawn inside a "bounding box" drawn with `pw_vector`. This is fixed for Release 3.5.

**Problem with pw_line and
pw_polyline**

Nothing would happen, after starting `suntools` by default on a system with a graphics processor and using `pw_line` or `pw_polyline` to draw a vector from left to right whose starting point had a negative *x* coordinate. However, if you started `suntools` with the option `-d /dev/cgtwo0`, these two functions worked fine. To be affected, the line had to be textured or have a width greater than 1. This is fixed for Release 3.5.

**tektool: Menu Missing Items
and no Longer in Walking Style**

In Release 3.4, the code for `tektool` was mistakenly backed up to the 3.0 level. The 3.4 `tektool` is identical to the 3.0 `tektool` - thus no walking menus in 3.4. Release 3.5 contains the 3.2 `tektool`, which has the walking menus.

tektool Key Does Not Work

The 3.5 version of `tektool` supports the use of `[Props]` key.

tektool Could Core Dump

A particular data stream caused `tektool` to go away with a core dump and produced the message

```
(1) bad patrem = 47104, pp[4] = 12288
abort - core dumped
```

This bug is fixed for Release 3.5.

Crosshair Cursors not Working

The crosshair cursors did not work in a `CANVAS_FAST_MONO` canvas on a desktop in a `cg4` frame buffer. The crosshair cursor was not erased properly, and many repetitions of the following set of error messages appeared on the console:

```
Kernel cursor Roperr 3
Kernel cursor Roperr 4
```

Screenblank Too Effective

The `screenblank` program allowed both `-k` and `-m` options to be used while in `suntools` and/or console mode, which prevented you from getting the screen back once the video was turned off. Once the video was off, you had to reboot the system.

5.7. Compiler Bugs Fixed

This section describes the bugs fixed in the compiler.

Optimizer Substitution Error

Previously, the optimizer occasionally substituted the register `a4` instead of the register `d5` in the file of assembler instructions that were output from the compilation of a FORTRAN program. When this substitution occurred, the FORTRAN program behaved abnormally after optimization.

`dbx` Shows Segmentation Violation

In `dbx`, a segmentation violation no longer occurs when using the `stepi` command to skip past a 68881 multiplication instruction.

Recursive `#includes` Confuse `dbx`

Programs with lexically recursive `#include` statements can now be debugged with `dbx` or `dbxtool`.

`ld -A` Produces Bogus Symbol Tables

The symbol table that was output from the command `ld -A` no longer contains incorrect values.

`printf` Padding Strings with Leading Zero's are Broken

The documentation for `printf(3S)` stated that: If the field width for an `s` conversion is preceded by a 0, the string is right adjusted with zero-padding on the left. This is fixed for Release 3.5.

`dbx` Failure

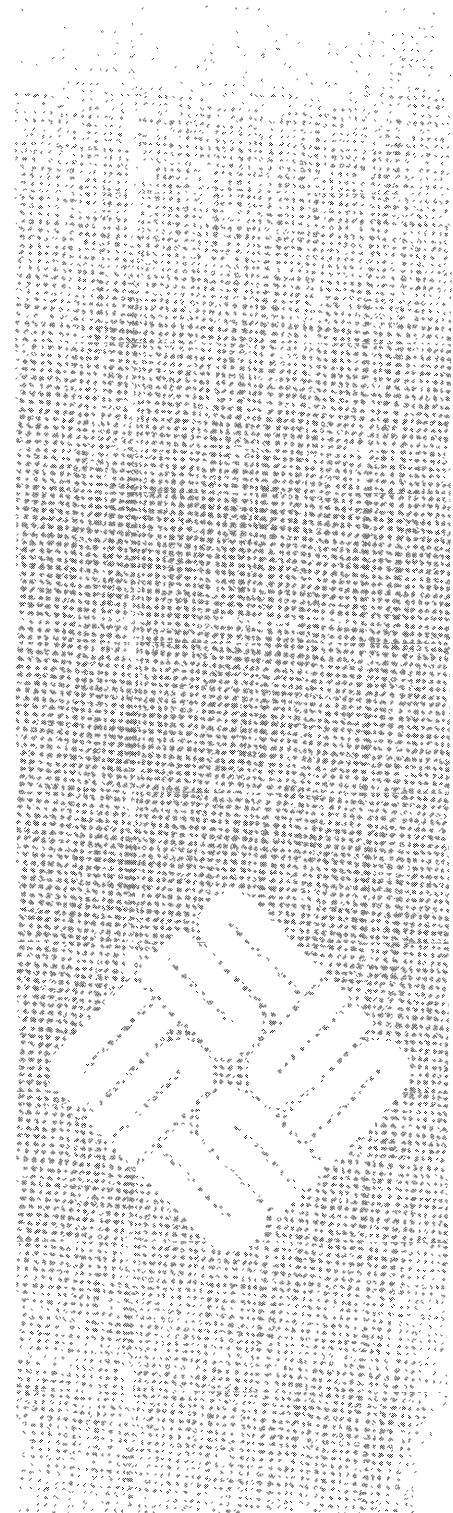
When you used `ld -r` to combine a number of object modules into one big module, and then loaded everything with `ld`, `dbx` sometimes failed on the resulting program. The error message was long, but always contained:

```
gettype: file index too large
```

This bug is fixed for Release 3.5.

3.4 Enhancements

3.4 Enhancements	89
6.1. Graphics Enhancements	89
Pixrect Enhancements	89
Unstructured Text Functions	89
Graphics Utilities	90
Ordered Dithering with <code>rasfilter8to1</code> (1)	90
Single Binary for Graphics Utilities	90
<code>screendump</code> (1)	90
<code>clear_colormap</code> (1)	90
SunCGI Enhancements	90
Signal Handling	90
Set VDC Extent	91
Additional CGI Error Messages	91
Open a CGI Canvas	92
6.2. SunView Enhancements	93
Improvements to <code>cmdtool</code>	94
Scroll Mode	94
Tty Mode	94
How Tty Mode Works	94
Compatibility of Applications with <code>cmdtool</code>	95
Compatibility of SunOS Releases with <code>cmdtool</code>	95
Changing Modes Manually	96
Tips and Tricks	96



Things to Watch For	97
Menu Enhancements	98
Stay-up Menus	98
Row-Column Order in Menus	99
Menu Compatibility	99
Corresponding New Menu Attributes	99
Text Subwindow Enhancements	99
“Quadruple-clicking” Now Selects the Entire Document	99
Positive Confirmation When ‘Find’ Fails	99
The Upper Context is Set Correctly After ‘Find’	99
Text Subwindow’s Temporary File Name Shortened	99
Specifying Different Defaults	99
Miscellaneous Enhancements Visible to the User	100
New Tty Subwindow Attributes	100
TTY_TTY_FD	101
TTY_PID	101
TTY_ARGV_DO_NOT_FORK	101
Sample Uses of New Tty Attributes	101
Example Using TTY_TTY_FD to Read/Write Standard Input/Output	101
Example Using TTY_PID to Fork a New Process on Child Death	103
SunView Library Changes	104
lint Libraries	104
Profiled Libraries	104
Improved Attribute-Value List Error Handling	105
6.3. Kernel Enhancements	105
6.4. Networking Enhancements	105
Name Resolver Library	105
New IP, TCP, and ICMP from 4.3BSD	105
RPC Programs Read Broadcast Address	105
6.5. File Formats	105

3.4 Enhancements

Because this is a upgrade release (as well as a full boot release) incorporating Release 3.4, the following 3.4 enhancements are detailed in this chapter.

- Graphics
- SunView
- Kernel
- Networking
- File formats

The new SunPro optional software provides two enhancements, the `filemerge` program and a new version of `make`. `filemerge` is discussed in its manual page in Appendix C, and `make` is discussed in a separate manual packaged with your Release 3.4 software.

6.1. Graphics Enhancements

This section describes enhancements that have been made to the *Pixrect*, *SunCGI* and *SunCore* graphics software.

Pixrect Enhancements

Release 3.4 contains the following additional functionality for the *Pixrect* library.

Unstructured Text Functions

The following unstructured functions for drawing text have been added to the *Pixrect* library:

```
pr_text(pr, x, y, op, font, text)
Pixrect *pr;
int x, y, op;
Pixfont *font;
char *text;

pr_ttext(pr, x, y, op, font, text)
Pixrect *pr;
int x, y, op;
Pixfont *font;
char *text;
```

These functions correspond to the `pixwin` functions `pw_text` and `pw_ttext`. `prs_text` and `prs_ttext` macros are also provided, although they are

identical to `pf_text` and `pf_ttext`, respectively.

Graphics Utilities

This section describes the enhancements to graphics utility programs.

Ordered Dithering with `rasfilter8to1(1)`

In Release 3.4, an ordered dither option has been added to `rasfilter8to1(1)`. This technique often gives better results than thresholding when displaying color images on a monochrome screen. It accepts filename arguments on the command line and accepts any type of rasterfile as input. See the manual page for `rasfilter8to1(1)` in Appendix C.

Single Binary for Graphics Utilities

1) In Release 3.4, the utility programs `clear_colormap(1)`, `rastrepl(1)`, `rasfilter8to1(1)`, `screendump(1)`, and `screenload(1)` have been merged into a single binary. (In `/usr/bin`, `clear_colormap(1)`, `rastrepl(1)`, `rasfilter8to1(1)`, and `screenload(1)` are symbolic links to `screendump(1)`.)

`screendump(1)`

- 1) In Release 3.4, a no-copy option has been added to `screendump(1)`. This option reduces virtual memory usage when dumping a memory frame buffer at the cost of lengthening the time interval that the frame buffer contents must be stable to ensure a consistent output image.
- 2) In Release 3.4, the heuristic for finding a color frame buffer has also been improved.

See the manual page for `screendump(1)`.

`clear_colormap(1)`

In Release 3.4, `clear_colormap(1)` has several new options listed below. See the `clear_colormap(1)` manual page for more information.

- 1) `clear_colormap(1)` clears the overlay plane and initializes the overlay enable plane by default on `cgfour` frame buffers. This behavior can be disabled with the `-o` option.
- 2) `clear_colormap(1)` can specify a frame buffer device with the `-f` option.
- 3) `clear_colormap(1)` can disable screen clearing with the `-n` option.

SunCGI Enhancements

Release 3.4 contains the following additional functionality for the *SunCGI* library.

Signal Handling

SunCGI's signal handling has been changed in Release 3.4 to use the Notifier. *SunCGI* applications should now use the Notifier instead of `signal(3)`. As an example, it is easier for an application to request notification of `SIGWINCH`'s than to use `set_up_sigwinch`. For compatibility with previous releases, `set_up_sigwinch` is still available and performs as documented.

Set VDC Extent

```

cgipw_set_vdc_extent(desc, c1, c2)
Ccgiwin *desc;
Ccoor *c1, *c2;
    /* bottom left and top right corner of VDC space */

```

In *CGIPW* mode, `cgipw_set_vdc_extent` changes the coordinate mapping from pixel coordinates to normalized VDC coordinates with origin in the lower-left corner. This function is the *CGIPW* “equivalent” of `vdc_extent`, and its arguments — `c1` and `c2` — are identical to those passed to the standard `vdc_extent` function.

Since all view surfaces share a single VDC mapping, you must use the same kind of coordinates for each surface. In Release 3.2, `cgipw_set_vdc_extent` was designed to work for a single view surface, but could work with multiple view surfaces if each were called discretely using the same arguments (`c1,c2`). You must call `cgipw_set_vdc_extent` for each surface (after creating the surface with `open_cgi_pw`) before setting any other attributes or drawing primitives. The sequence of these calls is unimportant — either (*Open; Set; Open; Set; Open; Set*) or (*Open; Open; Open; Set; Set; Set*) will work.

Once the `cgipw_set_vdc_extent` function is called, the only way to return to `pixwin` coordinates is to call `close_pw_cgi`, then `open_pw_cgi` and begin again. The *CGIPW* client should continue using the same *CGIPW* functions (`cgipw_polyline`, `cgipw_rectangle`), because `cgipw_set_vdc_extent` changes only the mapping of coordinate values passed to *CGIPW* functions, and nothing else in *CGIPW* mode.

Additional CGI Error Messages

The Release 3.4 version of *SunCGI* detects additional error conditions. Below is a list of the error codes and their associated error messages.

- | | |
|---------------|--|
| EFILACC [113] | Unable to access file. An attempt to open or access a file or device failed due to its nonexistence, improper permissions, or other configuration-dependent errors. |
| ECGIWIN [114] | <code>Ccgiwin</code> descriptor is invalid. A <i>CGIPW</i> function was called with a view surface descriptor that is <code>NULL</code> , or that refers to a view surface that is not open. This is probably caused by using the descriptor returned from a failed <code>open_cgi_pw</code> or <code>open_cgi_canvas</code> call. |
| EBADDATA [95] | Contents of input data record are invalid. The <code>Cinrep</code> data structure passed by an application to an input function is incorrect. |

Open a CGI Canvas

```

Error open_cgi_canvas(canvas, desc, name)
Canvas *canvas;
Ccgiwin *desc;
Cint *name;

```

In order to make *SunCGI* compatible with *SunView* canvases, the Release 3.4 version of *SunCGI* contains a new view surface initialization function. This must be used instead of `open_cgi_pw`, so *SunCGI* knows to handle coordinate transformation and window repainting in a way that is compatible with the *Canvas* package.

`open_cgi_canvas` is used in place of `open_cgi_pw` to initialize *SunCGI* to use a canvas. This is needed to give *SunCGI* the canvas handle, which is a higher-level object than a `pixwin`. After calling this initialization function, the resultant descriptor can be treated like that from `open_cgi_pw` for calling any *CGIPW* function, including `close_cgi_pw`.

With the exception of input functions, *CGIPW* functions should work correctly with canvases. In particular, the new *SunCGI* extension `cgipw_set_vdc_extent` will correctly map the VDC extent to the underlying canvas. *SunCGI* input should not be used with canvases, since the *Canvas* package handles all input events on the canvas by calling a client handler function. *SunCGI* has no knowledge of this handler, and would consume input events the *Canvas* package expects, thus interfering with scrollbars and tool border functions such as menus.

Figure 6-1 contains an example program that illustrates a simple use of canvases with *SunCGI*.

Figure 6-1 *CGI Canvas Example Program*

```

#include <suntool/sunview.h>
#include <suntool/canvas.h>
#include <cgipw.h>

Frame   frame;
Canvas  canvas;
Ccgiwin vpw;
int     name, canvas_event_proc();

main()
{
    frame = window_create(NULL, FRAME, 0);
    canvas = window_create(frame, CANVAS,
        CANVAS_AUTO_SHRINK, FALSE,
        WIN_EVENT_PROC, canvas_event_proc,
        CANVAS_WIDTH, 1000,
        CANVAS_HEIGHT, 1000,
        WIN_VERTICAL_SCROLLBAR, scrollbar_create(0),
        WIN_HORIZONTAL_SCROLLBAR, scrollbar_create(0),
        0);
}

```

```

open_pw_cgi();
open_cgi_canvas(canvas, &vpw, &name);
window_main_loop(frame);
}

canvas_event_proc(window, event)
Window window;
Event *event;
{
    Ccoor lr, ul;

    if (event_is_down(event))
        return;

    switch (event_id(event)) {

    case MS_LEFT:
        ul.x = event_x(event);
        ul.y = event_y(event);
        lr.x = event_x(event) + 10;
        lr.y = event_y(event) + 10;
        cgipw_rectangle(&vpw, &lr, &ul);
        break;

    case MS_RIGHT:
        window_done(frame);
        break;

    default:
        break;
    }
}

```

6.2. SunView Enhancements

Release 3.4 has several enhancements to SunView over Release 3.2, which, in turn, had numerous enhancements and performance improvements compared with Release 3.0. This section describes the enhancements.

Improvements to `cmdtool`

You can now run programs such as `more` and `vi` in a `cmdtool(1)`. The changes that support this are as follows:

- `cmdtool` directly supports “raw,” “cbreak,” and “no-echo” modes (used by `rlogin`, `more`, `su`, and so on). These modes are documented in the `tty(4)` manual page in the *UNIX Interface Reference Manual*.
- In programs that perform cursor motion (such as `vi`, game programs, and any terminal-based applications that use `curses`), `cmdtool`'s text subwindow is automatically overlaid by a `tty` subwindow. The `cmdtool` then responds to the cursor motion escape sequences sent out by the program (documented in `console(4S)`).

Scroll Mode

Normally, `cmdtool` only sends the line you are typing to the application when you press **Return**. This means that you can edit the current command as you type it using the mouse and standard text operations.

When an application takes `cmdtool` out of cooked, echo mode, every key-stroke you make is immediately sent to the application. This means that you cannot move the caret² backwards to edit the line you are typing — those characters have already been sent to the application. This happens, for example, when you `rlogin` to another machine.

Of course you can still use the Put and Get commands, and, by selecting ‘Enable Edit’ from `cmdtool`'s menu, you can edit anywhere in `cmdtool`'s transcript at any time, but the changes will not be sent to the application.

Tty Mode

Basically, well-behaved applications that go into cursor motion mode, such as `vi`, automatically set `cmdtool` to the `tty` mode. The window clears, the scrollbar vanishes, and the caret turns into a rectangular cursor to indicate that the program now controls the insertion point. Since the cursor is moved around under program control, you can't scroll or edit in `tty` mode, though Put and Get still work. See “Tips and Tricks” below for a way to get the benefit of scrolling while in `tty` mode.

When a well-behaved program is finished, it will automatically return you to scroll mode.

How Tty Mode Works

Many applications, such as full-screen editors, want to control text display all over their window and position the cursor anywhere within it, instead of interacting with the user on a line-by-line basis. Most such applications use `termcap(5)` or `curses(3X)` (which uses `termcap`) to interact with the display.³ Some terminals have different modes for the two kinds of interaction, just as `cmdtool` does, so `termcap` has always allowed a terminal entry to specify

² In terminal-based programs, the application controls where new characters that you type will appear. In `tty` mode, this *insertion point* is marked by a black rectangle which is usually referred to as the *cursor*. In SunView, the *cursor* is also the pointer moved by the mouse independently of each window's insertion point. The SunView text facility lets you set the insertion point for editing using the mouse. The triangle that marks the insertion point in panels and text subwindows is called the *caret*; the active caret flashes by default, though you can turn this off in `defaultsedit`.

- a `ti` escape sequence to send to the terminal when entering cursor motion mode
- a `te` escape sequence to send to the terminal when exiting cursor motion mode.

In Release 3.4, there is a new entry in `termcap`, called `sun-cmd`, for `cmdtool`. This is the same as the ordinary Sun console `termcap` entry (`sun`), except that it defines `ti` and `te` escape sequences. The 3.4 `cmdtool` understands these escape sequences, and changes from scroll mode to `tty` mode and back as described above.

Compatibility of Applications with `cmdtool`

Well-behaved applications that want to use cursor motion should send `ti` and `te` sequences to the terminal they are running on to set it into cursor mode and out of it, provided that the `termcap` entry for the terminal specifies them. Thus they will work without modification in `cmdtool`.

CAUTION

Applications that use cursor motion must send these escape sequences for `tty` mode to work. There are several ways this can happen:

- If an application use `curses`, then `curses` will send the escape sequences automatically.
- If an application uses `termcap` directly, it must send out the `ti` and `te` sequences itself when entering and exiting cursor motion mode.
- Similar comments apply if you are using System V's `terminfo(5V)` or `curses(3V)`, both in the optional System V compatibility package. In this the capabilities to enter and exit cursor mode are called `smcup` and `rmcup`, respectively.
- If an application uses a private database of terminal capabilities, a special entry for `cmdtool` must be added, with the private database's equivalents for `ti` and `te` defined.
- If an application doesn't use `termcap` or its own equivalent, it must be changed to send out the escape sequences described by `cmdtool`'s `ti` and `te` sequences; if the application is running in a `shelltool`, the escape sequences have no effect.

Compatibility of SunOS Releases with `cmdtool`

If you are in `cmdtool` and you `rlogin` to another machine that has an older `termcap`, you will get the message

Type `sun-cmd` unknown

and `tty` mode will not work. The work-around is to set the terminal type to `sun` and manually enter `tty` mode when desired (see *Changing Modes Manually* below). The solution is to merge the `sun-cmd` entry into the other machine's `termcap`, or copy the entire `3.4/etc/termcap` file to the other machine.

Changing Modes Manually

It is sometimes useful to be able to set `cmdtool` to `tty` mode or `scroll` mode yourself. In Release 3.4, the menu in each mode has a new menu item to change to the other mode: in `scroll` mode there is a menu item to 'Disable Scroll' and in `tty` mode there is a menu item to 'Enable Scroll.' Remember that in normal use, `cmdtool` jumps between the two modes automatically.

Tips and Tricks

You can 'Split' `cmdtool` into two or more separate subwindows using the menu command of the same name. Each subwindow can be scrolled independently. When a program sends `cmdtool` into `tty` mode, only one subwindow is overlaid by a `tty` subwindow. It is the one with the *keyboard focus* at the time; by default, the subwindow that the cursor is in has the keyboard focus. The other subwindows can still be scrolled to view and select previous commands, as shown in this example:

```
cmdtool - /bin/csh
E ./          intro.ms.ps      n7.c          t10.c
  .tags       intro.ms.ps.mod   n7.o          t10.c.tut
MY_README    jag/              n8.c          t10.o
Makefile     jagmacs.ne       n8.o          t6.c
README       logo2.ps         n9.c          t6.c.tut
SCCS@        makedev*         n9.o          t6.o
adobe@      makedev.c       newtest.ps    tdef.h
core         n1.c             ni.c          test.dit
d.h          n1.o             ni.c.tut      test.dit.ps
dev.h        n2.c             ni.o          test.nor.dit
devps/       n2.o             nii.c         test.nor.dit.ps
diffs        n3.c             nii.o         test.nr
diffs.full   n3.o             pstroff*     test.ps
diffs.full%  n4.c             s.h          tw.h
ext.h        n4.o             sfxlist*     v.h
hytab.c      n5.c             suftab.c
polar% vi file_list
i
E
In the pstroff directory are the following files
  Makefile      creates files
  README        read this first
  [ ]
~
~
~
~
~
~
~
~
"file_list" [New file]
```

Figure 6-2 A *Split* `cmdtool`

In addition to the menu items provided to change modes, it is also possible to change from one mode to the other via a function key or an alias. The two escape sequences are `ti` and `te` in the `sun-cmd` entry in `/etc/termcap`:

`ti` to begin `tty` mode: `\E[>4l`

and

`te` to return to `scroll` mode: `\E[>4h`

`\E` means the Escape character.

Note: In normal use, you do not need to use these techniques to go from one mode to the other — it happens automatically.

So, to go from one mode to the other, you can map the two escape sequences to keys in your `~/ .ttyswrc` file. The following example maps them to top function keys **F8** and **F9**:

```
# go to tty mode
mapo T8 \E[>4l
# go to scroll mode
mapo T9 \E[>4h
```

You can also set these up as aliases in your `~/ .cshrc` file as follows:

```
if ($?WINDOW_PARENT) then
    alias gotty 'echo -n "^[[>4l"'
    alias goscroll 'echo -n "^[[>4h"'
endif
```

(If you are using the C-shell, to get the Escape character `^[[` in the file, you may have to precede it with **CTRL-V**.)

Things to Watch For

File completion in the C-shell (`set filec`) does not work properly in a `cmdtool`.

You must be careful when typing ahead, because characters typed while `cmdtool` changes modes may be lost or sent to the other mode's window.

The scroll mode of `cmdtool` picks up most of the *Text* defaults in `defaultsedit`. However, if *Text/Font* is different than the font used in tty subwindows (because *SunView/Font* is different, or because the `-wt fontname` frame command line argument was used), then the font will *not* change when the subwindow goes from one mode to the other. Otherwise, for example, the number of rows and columns would be different in different modes.

In normal use, you should not have to change modes manually. However, there are a few reasons why you may need to:

- Some programs do not output the correct escape sequence to go into and out of tty mode.
- In Release 3.4, only the tty subwindow supports file completion (`set filec`) in the C-shell.
- In Release 3.4, only the tty subwindow can embolden characters (used, for example, by `man(1)` to highlight words).
- When the root file system fills up so that `cmdtool` cannot write to its transcript file (`/tmp/tty.txt.Process_id`) or its edit log (`/tmp/TextProcess_id.counter`), `cmdtool` switches into tty mode. It does not detect when you have freed up enough disk space; you have to explicitly put it back into scroll mode.

Menu Enhancements

Stay-up Menus

You can change the behavior of pop-up menus so that they stay up even when you take your finger off the right button on the mouse. Normally, a menu pops up as soon as you press the mouse's right button, and disappears when you release it. However, if you set *Stay_up* in the *Menu* category of `defaultsedit(1)` to *True*, then the first click of the right button displays the menu. The second click selects the highlighted menu item, if any, and makes the menu go away. Between the two clicks, the menu remains visible, and grabs all window input — you cannot type or select in any window while the menu is up.

Note: The right button on the mouse is usually the menu button, although this can be changed using the *Input* category of `defaultsedit`.

If you hold the right button down in *Stay_up* mode, the cursor changes to the following shape to indicate that the menu will appear when you release the button:



Figure 6-3 *Cursor Indicating “Stay Up” Menu Imminent*

Another new attribute in the same category lets you set your menus so that menu strings come up centered; set *Center_string_items* in the *Menu* category of `defaultsedit(1)` to *True*. This helps to differentiate different menu options. Try it to see if you like it. Here is a sample.

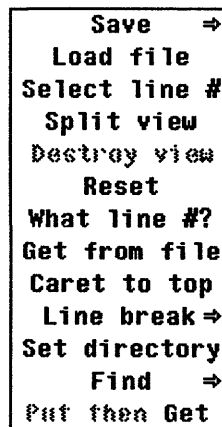


Figure 6-4 *Sample Centered Menu*

Row-Column Order in Menus	Some applications put up menus in several columns, for example the menu for <code>mailtool</code> 's <code>folder</code> button. The default order for the items is row-major order, so that adjacent items are in the same row and different columns. But if you set <code>Items_in_column_major</code> in the <code>Menu</code> category of <code>defaultsedit(1)</code> to <code>True</code> , then items are laid out in columns. This is the way <code>ls(1)</code> displays files.
Menu Compatibility	These effects (and many others available in the <code>Menu</code> category of <code>defaultsedit</code>) only work if you have <code>Walking_Menus</code> set to <code>Enabled</code> in the <code>SunView</code> category of <code>defaultsedit</code> .
	<i>NOTE</i> In Release 4.0, the default will be walking menus, although Sunwindows-style menus will still be provided for compatibility with releases prior to 3.0.
Corresponding New Menu Attributes	The menu attributes that correspond to these new features are <code>MENU_STAY_UP</code> , <code>MENU_CENTER</code> and <code>MENU_COLUMN_MAJOR</code> in <code><suntool/walkmenu.h></code> . Each takes a Boolean value that turns the feature on or off.
Text Subwindow Enhancements	A variety of bug fixes and enhancements have been made to the text subwindow package (used in <code>textedit(1)</code> and many other programs):
“Quadruple-clicking” Now Selects the Entire Document	The fourth click in succession of the left mouse button selects the entire contents of the text subwindow. As before, the first click selects a character (and sets the insertion point); the second click selects a word; and the third click selects a line. The defaults <code>Multi_click_space</code> and <code>Multi_click_timeout</code> in the <code>Text</code> category of <code>defaultsedit</code> determine the maximum distance in space and time between clicks for them still to be considered a multi-click.
Positive Confirmation When ‘Find’ Fails	When ‘Find’ in a text subwindow fails to find characters matching the selection, the text subwindow flashes and beeps. ⁴
The Upper Context is Set Correctly After ‘Find’	When ‘Find’ succeeds and displays the matching characters, it now displays the lines above the match (the “upper context”) instead of positioning the match at the top of the window. <code>Upper_context</code> in the <code>Text</code> category of <code>defaultsedit</code> determines the number of lines above the selection that are displayed.
Text Subwindow’s Temporary File Name Shortened	The temporary file that the text subwindow uses to store your edits to the file has been shortened to <code>/tmp/TextProcess_id.counter</code> from <code>/tmp/EtHosthostnumberProcessProcess_idCountercounter</code> .
Specifying Different Defaults	The <code>defaultsedit</code> program maintains a personal <code>~/defaults</code> file that records your SunView preferences; this defaults file is examined when tools start up. Often you want different tools to use different defaults. For example, you might want a <code>textedit</code> used to edit multiple files to have a small font, while you want <code>mailtool</code> to use a large font.

⁴ Flashing and beeping can be individually turned off by setting `Audible_Bell` and `Visible_Bell` in the `SunView` category of `defaultsedit` to `Disabled`.

In 3.4, tools now look at the `DEFAULTS_FILE` environment variable (if it exists) to determine where to read the user's preferences. You can set this to point to a special defaults file before you invoke a tool. If you are a C-shell user, you would go through steps like the following to create this special file:

```
% mv ~/.defaults ~/.defaults.hold
% defaultsedit
(Make a special set of defaults.)
Save them and quit defaultsedit.)
% mv ~/.defaults ~/.defaults.special
% mv ~/.defaults.hold ~/.defaults
```

Then, when you want to start up a tool, `mailtool` for example, using the special defaults, type:

```
% (setenv DEFAULTS_FILE ~/.defaults.special; mailtool) &
```

This is especially useful if you use several machines with different size and resolution displays — you can make a second `.defaults` file that specifies larger fonts for text and menus, larger scrollbars, and the like. You can select this automatically from your `~/.login` file.

```
if (`hostname` == "high-res_machine") then
    setenv DEFAULTS_FILE ~/.defaults.hires
else
    setenv DEFAULTS_FILE ~/.defaults
endif
```

Miscellaneous Enhancements Visible to the User

If you make changes to one of the categories in `defaultsedit`, then try to change to another category, the program warns you that your changes will be lost if you do not 'Save' them.

New Tty Subwindow Attributes

New attributes for tty subwindows give programmers more control over when and whether a process is forked to run in a frame's tty subwindow.

Table 6-1 *New Tty Subwindow Attributes*

Attribute	Type	Description
<code>TTY_TTY_FD</code>	int	File descriptor of the tty associated with the tty subwindow.
<code>TTY_PID</code>	int	Get only. Process ID of tty subwindow child or handle of client running in the tty subwindow. A value of <code>TTY_INFINITY</code> says there is no program or client running in the tty subwindow. If get returns a value of <code>-1</code> , the child was not forked successfully.

TTY_TTY_FD

You can use `TTY_TTY_FD` to read and write to the pseudotty in the tty subwindow using standard UNIX I/O routines. Note that `TTY_TTY_FD` is the file descriptor of the pseudotty, not the file descriptor of the tty subwindow. The latter is used for some low-level window manipulation procedures.

TTY_PID

You can use `TTY_PID` to monitor the state of the child process running in the tty window via the Notifier using `notify_set_wait3_func()`. The client's `wait3()` function gets called when the state of the process in the tty subwindow changes. It can then do something useful, such as destroying the tty window or starting up another process in the tty subwindow.

You can set `TTY_PID` as well as get it, but if you set it then you are responsible for setting the `notify_set_wait3_func()` to catch the child's death, and for making the standard input and standard output of the child go to the pseudotty.

TTY_ARGV_DO_NOT_FORK

There is a new potential value for `TTY_ARGV`. Setting a value of `TTY_ARGV_DO_NOT_FORK` tells the system not to fork a child in the tty subwindow. In combination with `TTY_FD`, this allows the tool to use standard I/O routines to read and write to the tty subwindow. This simplifies porting terminal-oriented graphics programs, which interact with the user on the model of *write a prompt... read a reply*, to SunView. However, in most cases you should redesign programs to use a real windowing interface made up of SunView components.

This new capability makes obsolete the work-around required in previous releases of SunView if you wanted a window program to read and write from its own tty subwindow. The old approach was demonstrated in the example programs `typein.c` and `loopback.c` listed in the *Example Programs* chapter of the *SunView Programmer's Guide* and included in optional software in `/usr/src/sun/suntool`. A revised version of `typein.c` is in Section 12.1, "Errata and Addenda for the SunView Programmer's Guide," of this manual, and is also on the release tape.

Sample Uses of New Tty Attributes

Here are some other sample uses of the new attributes.

Example Using `TTY_TTY_FD` to Read/Write Standard Input/Output

The following program is the framework for the same kind of *write a prompt... read a reply* kind of program as `typein`, but is a more straightforward conversion to SunView. Instead of rewriting applications to move their main flow of control into `window_main_loop()`, you can call `notify_do_dispatch()` to tell the Notifier to run when it can. The Notifier will handle selections, menu processing, window resizing, and so on, each time the program does a `read()` or `write()`. Read "Porting Programs to SunView" in the *Notifier* chapter of the *SunView Programmer's Guide* for more information.

```
#include <stdio.h>
#include <sys/wait.h>
#include <suntool/sunview.h>
```

```

#include <suntool/tty.h>
#include <suntool/textsw.h>

#define BUFSIZE 1000
static int      my_done;

static Notify_value
my_notice_destroy(frame, status)
    Frame      frame;
    Destroy_status status;
{
    if (status != DESTROY_CHECKING) {
        my_done = 1;
        (void)notify_stop();
    }
    return (notify_next_destroy_func(frame, status));
}

main(argc, argv)
    int      argc;
    char     *argv[];
{
    Frame     base_frame;
    Tty       ttysw;
    int       tty_fd;
    char      buf[BUFSIZE];

    my_done = 0;
    base_frame = window_create(NULL, FRAME,
                               FRAME_ARGC_PTR_ARGV,    &argc, argv,
                               0);

    ttysw = window_create(base_frame,          TTY,
                          TTY_ARGV,          TTY_ARGV_DO_NOT_FORK,
                          0);

    tty_fd = (int)window_get(ttysw, TTY_TTY_FD);
    dup2(tty_fd, 0);
    dup2(tty_fd, 1);

    (void)notify_interpose_destroy_func(base_frame, my_notice_destroy);
    window_set(base_frame, WIN_SHOW, TRUE, 0);
    (void)notify_do_dispatch();
    puts(prompt_to_user);
    while (gets(buf)) {
        if (my_done)
            break;
        /*
         * This is where the meat of the program
         * would be if this were a real program.
         */
        puts(buf);
        (void)notify_do_dispatch();
    }
}

```



```

    exit(0);
}

```

Example Using TTY_PID to Fork a New Process on Child Death

This program is notified of the death of its tty subwindow's child and forks a new one. To do this, you interpose a `wait3()` function using `notify_interpose_wait3_func()`, which "watches" the process ID of the child program running in the tty subwindow.

```

#include <stdio.h>
#include <sys/wait.h>
#include <suntool/sunview.h>
#include <suntool/tty.h>
#include <suntool/textsw.h>

static char    *my_argv[] = {"my_prog", 0};

static Notify_value
my_wait3(ttysw, pid, status, rusage)
    Tty        ttysw;
    int        pid;
    union wait  *status;
    struct rusage *rusage;
{
    int        child_pid;

    notify_next_wait3_func(ttysw, pid, status, rusage);
    if (! (WIFSTOPPED(*status))) {
        window_set(ttysw,
            TTY_QUIT_ON_CHILD_DEATH, FALSE,
            TTY_ARGV,                my_argv,
            0);
        child_pid = (int>window_get(ttysw, TTY_PID);
        notify_interpose_wait3_func(ttysw, my_wait3, child_pid);
    }
    return NOTIFY_DONE;
}

main(argc, argv)
    int        argc;
    char    *argv[];
{
    Frame    base_frame;
    Tty      ttysw;
    int      child_pid;
    int      tty_type_flag = 0;

    base_frame = window_create(NULL, FRAME,
        FRAME_ARGC_PTR_ARGV,    &argc, argv,
        0);
    ttysw = window_create(base_frame, TTY,
        TTY_QUIT_ON_CHILD_DEATH, FALSE,

```

```

        TTY_ARGV,                my_argv,
        0);
    child_pid = (int>window_get(ttysw, TTY_PID);
    notify_interpose_wait3_func(ttysw, my_wait3, child_pid);

    window_main_loop(base_frame);
    exit(0);
}

```

SunView Library Changes

lint Libraries

New lint libraries for `libsuntool` and `libsunwindow` have been created. You can run your SunView applications against these using the `lint(1)` program to find potential bugs in them at compile time. The lint libraries are

```

/usr/lib/llib-lsuntool.ln
/usr/lib/llib-lsunwindow.ln

```

To run `lint` against the program `mysunview.c`, you would type something like:

```
% lint mysunview.c -lsuntool -lsunwindow -lpixrect
```

Profiled Libraries

Profiled versions of `libsuntool`, `libsunwindow`, and `libtoolmerge`⁵ are available in Release 3.4 if you want to analyze the execution of your SunView applications. The profiled libraries are

```

/usr/lib/libsuntool_p.a
/usr/lib/libsunwindow_p.a
/usr/lib/libtoolmerge_p.a
(there is no profiled version of libpixrect, yet)

```

The profiled libraries are only installed if during installation you answer `y` to the following question:

```
Do you want to install the new window profiled libraries ? :
```

during the 3.4 upgrade procedure; see step 10 in section 2.4. To compile the program `mysunview.c` in preparation for profiling with `gprof(1)`, you would type something like

```
% cc -pg mysunview.c -lsuntool_p -lsunwindow_p -lpixrect
```

The `gprof(1)` and `cc(1)` manual pages have more information on profiling.

⁵ `libtoolmerge` contains the `.o` files for all the tools in the `suntools` and `othertools` tool merges. It is used only when you build your own custom-merged tool set, as documented in Section 4.8, "SunView Enhancements," of the *Release 3.2* manual.

Improved Attribute-Value List Error Handling

The maximum length of attribute-value lists supported by the SunView packages (see `ATTR_STANDARD_SIZE` in `<sunwindow/attr.h>`) has always been 250. In this version of SunView, if the number of attributes in a list you pass to SunView exceeds this size, the attribute package now prints

```
Number of attributes(nnn) in the attr list exceeds
the maximum number(nnn) specified. Exit!
```

on standard error and exits with exit status 1.

There is an addendum for this in Chapter 8, “Errata and Addenda for the SunView Programmer’s Guide,”.

6.3. Kernel Enhancements

Release 3.4 changes the way in which soft ECC errors are handled by Sun-3/260s. Previously, these errors appeared on the console. Now they only appear in the file `/usr/adm/messages`.

6.4. Networking Enhancements Name Resolver Library

Release 3.4 includes the following enhancements to networking capabilities.

The Internet name resolver library, which was described in the `resolver(3)` and `resolver(5)` manual pages, is now included in the file `/usr/lib/libresolv.a`.

New IP, TCP, and ICMP from 4.3BSD

The kernel network code that implements IP (Internet Protocol), TCP (Transmission Control Protocol), and ICMP (Internet Control Message Protocol), has been updated to include many features of 4.3BSD for better congestion control.

RPC Programs Read Broadcast Address

Release 3.3 introduced the ability to set the broadcast address for any interface and accept any of the several “standard” broadcast addresses now in use. Release 3.4 includes new copies of RPC programs, such as `ypbind`, `rup`, and `rusers`, that read the broadcast address from the interface instead of always using the network number plus a host number of all zeros.

6.5. File Formats

The `termcap` database now contains a new entry, `sun-na`, to describe Sun workstations with arrow keys disabled. This was done to support SunSimplify database screens.

3.4 Bug Fixes

3.4 Bug Fixes	109
7.1. Language-related Bugs Fixed	109
as Bugs	109
68020 Addressing Bug	109
Assembler Long Jump Bug	109
C-related Bugs	110
op REG Compiler Error	110
Long Filenames Crashed cc	110
C Compiler Error with -f68881 Set	110
Casting Error Bug	110
Optimizer Failure	110
Optimizer Lost Address Register	110
cc Mishandled ', ' Operator	111
3.1 Optimizer Bug	111
cc -f68881 Bug	111
Division by Powers of 2 Bug	111
doscan Bug	111
Coercion Bug	111
Unneeded File Bug	111
cc Conversion Bug	112
Type-casting Bug	112
Integral-conversion Bug	112
No Overwrite Warning Bug	112

Peephole Optimizer Bug	112
Optimizer Register Use Bug	112
Register Variable Bug	112
Debugger-related Bugs	113
dbx #include Bug	113
dbx Printing Bug	113
dbx Stack Trace Bug	113
dbx Function Pointer Bug	113
dbx Casting Bug	113
Incorrect Help Bug	114
dbx Symbol Table Bug	114
dbx Error Message Bug	114
dbx Prefix Bug	114
where Command Bug	114
catch and ignore Report Bug	114
dbx clear Command Bug	114
dbx "No Program" Bug	114
dbx Array Index Bug	114
Private Procedure Bug	115
Symbolic Link Bug	115
Double Negative Bug	115
Numeric Conversion Bug	115
FORTRAN-related Bugs	115
Complex Number Compare Bug	115
INCLUDE Bug	115
f77 'op REG' Error Bug	115
Number Scaling Error Bug	116
f77 read Bug	116
FORTRAN Uppercase Bug	116
f77 read Bug	116
Bug in Assembly Phase of Inline Expansion	116
Inline Expansion Bug	116
f77 -a Bug	117
f77 Compiler Bug	117
Incorrect Handling of Backslash Characters	117

Problems with Profiling Options	117
Redundant I/O-List Parentheses Bug	117
system and fork Error Bug (1003162)	118
Pascal-related Bugs	118
Pascal Filename Bug	118
Library Bugs Fixed	118
prof.h Failure	118
Missing Library Routine	118
System V putpwent Bug	118
System V ioctl Bug	118
System V-related curses Bug	118
Duplicate Library Routine	118
Utility Bugs Fixed	118
tcov Failed on SIGCHILD	119
lint Error Output	119
Miscellaneous Bugs	119
pxp Bug	119
Compiler "Invalid" Options Bug	119
Double-Precision Transcendental Bug	119
7.2. Graphics Bug Fixes	119
Pixrect Bug Fixes	119
File I/O	119
Pixrect File I/O Rewrite	119
Type Declarations Added	120
pr_load	120
RMT_RAW Colormaps	120
Writing to a Rasterfile	120
RT_BYTE_ENCODED Rasterfiles	120
Rasterfile Filters	120
lint Library	120
Macros	121
rop_fastloop Macro	121
Memory Pixrects	121
Private Data Format for Memory Pixrects	121
8, 16, and 32-bit Memory Pixrect Support	122

pr_vector	122
pr_get	122
pr_put	123
pr_rop	123
Text	123
Memory Leaks with pf_open	123
pr_close	123
Opening vfont (5) Fonts	123
pf_ttext String Lengths	124
Miscellaneous	124
pr_getcolormap with bwone Pixrects	124
pr_batchrop	124
Global Definitions Removed from pr_traprop	124
pr_vector	124
pr_polypoint	124
pr_polygon_2	124
pr_replrop	124
pr_rop	125
pr_stencil	125
pr_line	125
Frame Buffer Device Drivers	125
gpone (4S)	125
FBIIOGVIDEO ioctl	125
cgfour 4S Support for the Sun-3/60 Frame Buffer	125
Graphics Utilities	125
rasfilter8tol (1)	125
rastrepl (1) and RT_BYTE_ENCODED Rasterfiles	126
SunCGI Bug Fixes	126
Resizing Viewports	126
NOCLIP	126
Clear View Surface	126
Cvwsurf	126
Quitting a View Surface Tool	126
Pixwin Output	126
CGIPW and SunView	126

Request Input	126
cfmksizespecmode	127
GP View Surfaces	127
Rectangle Perimeters	127
Arc	127
Freed Memory with Await Event	127
Multiple Character Keyboard Events in CGIPW	127
Inquire Device Identification	127
View Surface Table	127
Obsolete Code Removed	127
Close View Surface	127
Deactivate View Surface	127
CGIPW Validity Check	127
CGIPW Set VDC Extent	128
Scaling	128
Inquire Text Extent	128
Transparent Text	128
FORTRAN Input	128
VALUATOR Input with Await Event	128
Ignored Input with Await Event	128
Bad Data Structure Error Code with Await Event	128
SunCore Bug Fixes	128
Fat Vectors	128
raster Structure	128
put_raster	128
Clipped Vectors with GP View Surfaces	129
Polygon Vertex Limit	129
Backspace Key	129
7.3. SunView Bug Fixes	129
Defaults, .*rc Files, and Filters	129
.textswrc Processes Shell Metacharacters	129
Fixes to Input Filters	129
.ttyswrc String Changes	130
Blank lines in ~/.ttyswrc	130
Mail Aliases in defaultsedit	131

. <code>suntools</code> Can Accept Blank Lines	131
Other Bug Fixes Visible to the User	131
Cursor Correct on Sun-3/110	131
lockscreen <code>-e</code> on the Sun-3/110LC Color Display	131
‘UnZoom’ when Iconic	131
Characters Lost on Multi-character Input	131
Bug Fixes in <code>cmdtool</code>	131
Bug Fixes in <code>textedit</code> and the Text Subwindow	132
Programs Exiting in <code>shelltool</code> and <code>cmdtool</code>	133
Not Enough Swap Space in <code>shelltool</code> and <code>cmdtool</code>	133
Fixes to Tty Subwindows	133
Security Hole in <code>lockscreen</code> Fixed	134
Miscellaneous Fixes to Tools	134
User Interface Cleanup	134
Bug Fixes Visible to the Programmer	134
Window Geometry Fixes	134
WIN_ROWS When There is No Frame Header	134
Panel Choices Too High	135
‘Next’ Panel Item Created Too Low	135
Choice Item Positioning	135
Other Fixes Visible to the Programmer	135
Miscellaneous Text Subwindow Fixes	135
Pop Up Subframe Restrictions	136
WIN_FONT Now Works for Tty Subwindows	136
event_set_time()	136
KEY_BOTTOMFIRST and KEY_BOTTOMLAST Defined	136
Diagonal pw_copy() Fix	136
pw_line() in a Canvas	136
Click_to_Type in Panels	137
Cursor Positioning Escape Sequence in Tty Subwindows	137
Odd-Sized Icons	137
Internal Name Changes	137
Example Program <code>filer</code>	137
Other Fixes to Example Programs	137
7.4. SunOS Bug Fixes	138

Installation and Configuration-related Bugs	138
Missing Sun-3/110 Configuration Files	138
/sys/sundev/mti_conf.c File Missing	138
Remote Install Problems	138
Sun-3/50-4 Tape Cartridge Hangs	138
bzero Routine Could Not Handle Large MAXUSERS	138
Sun-3 Kernel Could Not Be Built Without Sun-3/260 Defined	139
Sun-3s with Early PROMs Had Rebooting Problems	139
3.3UPGRADE Did Not Work on Systems with 3.2EXPORT	139
Standalone Programs Did Not Exit Correctly	139
Kernel-related Bugs	139
Cursor Motion Not Coordinated with Mouse Motion	139
Memory Device Driver Bug	139
Bug in kadb	140
Error Reporting Problems with SCSI Tape Drives	140
VPC-2200 Parallel Printer/Plotter Interface Problems	140
Daisy Chained Shoe Boxes on Sun-3/50 Sometimes Hung	140
setrlimit Sometimes Did Not Return Error Codes	140
Occasional Spurious User Bus Errors Around Page Boundaries	140
New Version of the ie Ethernet Driver	140
Incompatible Routine in System V curses Library	140
Daylight Savings Times Were Incorrectly Calculated	141
MAKEDEV Did Not Create a vpc Node	141
Kernel Did Not Recognize Pseudo-terminals	141
Hard Links to Directories Caused System Panic	141
readlink Bug	141
Problems with kmem	141
indent-related Bug Fixes	141
4.3 BSD Bug Fixes	141
indent Problems with typedefs	141
Comment Blocks Were Incorrectly Formatted	141
Missing Semicolons Confused indent	142
indent Misformats Certain Combinations	142

Formatting Problems with Negative Floating-point Numbers	142
indent Put Spaces Around "\$"	142
Utilities-related Bug Fixes	143
Processes Failed to Reset Terminal Modes	143
Using sccs with a setuid Front End	143
Bug in dkinfo	143
Reply Sometimes Garbled Addresses	143
vgrind Omitted "?" Identifier	143
Numeric Fields Failed in tbl	143
.TH Macro Bug	143
Bug in maze Demonstration	144
Bug in chesstool	144
Complex Makefiles Caused Hash Table Overflow	144
indxbib Dropped Core	144
Shell-related Bug Fixes	144
Bug in login Command	144
time Memory Statistics Printed in Pages	144
7.5. Networking	144
Problems Using ifconfig for Diskless Clients	144
Bugs in TCP/IP	145
Net Booting on Non-zero Subnets	145
Telnet Server End-of-line Conventions	145
Frozen Configuration Files Ignored Domain Information	145
Lance Ethernet Driver Bug	145
ping Sent Bad Packets	145
rpcinfo Did Not Work on Non-networked Standalones	145
NFS Clients Sometimes Wrote Garbage	145
Software Loopback lo0 Address Could Not Be Set	146
FBIOGVIDEO Was Sometimes Unreliable	146
7.6. System Administration	146
Diskless Clients Can Now Reboot After Changing Ethernet Address with ifconfig	146
/etc/dkinfo	146
NFS Server Port Checking	146
Wrong termcap Entry for TVI-925	146

3.4 Bug Fixes

Release 3.4 was a major bug fix release. This chapter describes bugs from Releases 3.2 and 3.3 that were fixed in 3.4. The bug fix descriptions are organized into the following sections:

- languages
- graphics
- SunView
- SunOS
- networking
- system administration

7.1. Language-related Bugs Fixed

as Bugs

68020 Addressing Bug

This section describes the languages bugs that were fixed in Release 3.4.

Previous versions of `as` would not properly assemble statements of the form

```
movl    @(10:w,d3:1:4), d2
```

on Sun-3s.

The problem arose because `as` permitted you to specify a coprocessor ID or an FPA base register after the opcode, but did not require it to come *immediately* after the opcode.

This bug has been fixed in Release 3.4.

Assembler Long Jump Bug

`as` sometimes generated many lines like the following:

```
as: error (t.s:36277): Multiply defined symbol (phase error)
```

if a program contained long `jmp` instructions.

If you used the `-j` option of `as` to correct this during the assembly, the resulting object file did not work with the loader. The problem with `-j` was due to the processing of link instructions. This has been fixed in 3.4.

C-related Bugs

The following C language-related bugs have been fixed in Release 3.4.

op REG Compiler Error

Previous versions of `cc` could generate an error message like

```
"interp.c", line 14: compiler error: no table entry for op REG
```

if you wrote a program with chained structure assignments in a single statement. The bug has been fixed in this release.

Long Filenames Crashed `cc`

Previous versions of `cc` limited filenames to 100 characters. If you invoked the compiler with a filename argument exceeding 100 characters, `ccom` would dump core with a segmentation fault.

This bug has been fixed.

C Compiler Error with `-f68881` Set

In previous versions of `cc`, compiling a C program that cast an unsigned value to a double would generate

```
...compiler error: expression causes compiler loop: try simplifying
```

if the `-f68881` compiler flag was set. The bug has been fixed.

Casting Error Bug

Previously, `cc` sometimes incorrectly performed type conversions with unsigneds and bit fields. In binary operations involving an integer and an unsigned, sometimes the unsigned quantity would be converted to integer, rather than the integer being converted to unsigned.

The bug has been fixed in this release.

Optimizer Failure

Before the 3.2 release, the C code optimizer sometimes generated code that used one or more incorrect registers, usually when doing compares.

The bug has been fixed in this release.

Optimizer Lost Address Register

In Release 3.1, `cc` lost track of address register offsets if you used the `-O` option. A workaround was to disable one of the `c2` optimizations, at a cost of a slight reduction in code quality, as follows:

```
cc -Ooption c2 -dcoalesce
```

The bug has been fixed in this release.

cc Mishandled ',' Operator

Previous versions of `cc` mishandled C's "comma" operator during code generation. The problem resulted in a

```
compiler loop: try simplifying
```

error message. The bug is fixed in this release.

3.1 Optimizer Bug

The Release 3.1 optimizer could generate bad code losing store instructions and violating register `d6`.

The error was caused by the optimizer's improper specification of a redundant operand pattern. This caused `a3@+` in some code to be deleted because the pattern included it in the same category as `a3`.

cc -f68881 Bug

In Release 3.2, if the `-f68881` flag was set, `cc` would not compile unary functions except floats or doubles.

The problem occurred because when the `-f68881` flag was set, `cc` incorrectly evaluated a double precision operand in a `d`-register pair for condition codes. The same code compiled properly if either `-fsoft` or `-fswitch` were set.

The bug has been fixed in this release.

Division by Powers of 2 Bug

In Release 3.2, the C compiler's special-case code to handle division by powers of 2 on Sun-3s (MC68020s) did not properly manage register use. The equivalent code generated for Sun-2s (MC68010s) was correct.

The bug has been fixed in this release.

doscan Bug

In previous releases, the 4.2BSD version of `doscan` did not properly handle bad hexadecimal input, passing it through to the calling procedure. The bug has been fixed in this release.

Coercion Bug

The Release 3.1 C compiler had a bug that caused it to generate incorrect ints that have been cast to shorts. A segment of code like the following:

```
int x,y;
if ((short) x == (short) y)...
```

did not work properly; instead of masking off the high-order bits and doing a 16-bit compare, it performed a 32-bit compare.

The bug has been fixed in this release.

Unneeded File Bug

Previously, `cc` would invariably access `/usr/lib/pcexterns.o`, even if it did not need to. This meant that `/usr/lib/pcexterns.o` had to exist whenever the compiler driver was run.

The bug has been fixed in this release.

cc Conversion Bug In a previous release, `cc` incorrectly evaluated compile-time unsigned-to-double conversions.

The bug has been fixed in this release.

Type-casting Bug Previously, the C compiler ignored type casting in `if` statements. Type casting outside of `if` statements was done correctly. The bug has been fixed in this release.

Integral-conversion Bug Previously, the C compiler sometimes deleted necessary small integral conversions in programs. The bug has been fixed in this release.

No Overwrite Warning Bug In Release 3.2, `cc` did not warn you if your source file would be overwritten. The bug has been fixed in this release.

Peephole Optimizer Bug Previously, the peephole optimizer `/lib/c2` generated incorrect code if it encountered a structure passed by value and the total size of the argument list was not a multiple of 4.

The bug has been fixed in this release.

Optimizer Register Use Bug Previously, the peephole optimizer `/lib/c2` lost track of actual register usage when overlapping memory operands were modified. If the operand in memory was modified, the value used by compiled code (a copy saved in a register) would not be changed and would be incorrect.

The bug has been fixed in this release.

Register Variable Bug Previously, `cc` would issue the following error message:

```
no table entry for op REG
```

error message when compiling a program with code like

```
int test[100];

main()
{
    register int a, b, c, d, e, f;

    test[a] = b & test[c & 0x1] & test[d & 0x1];
}
```

because it allocated too many registers for double indexing, leaving too few for the “&” operation.

The bug has been fixed in this release.

Debugger-related Bugs

The following debugger-related bugs were fixed in Release 3.4.

dbx #include Bug

In some previous releases, dbx would not stop at any lines in a program that had #include'd initialized variables, such as sccs id's in the file header.

The bug has been fixed in this release.

dbx Printing Bug

Previously, dbx printed incorrect values for odd-length char arrays in structures. For example, dbx would print that the value of c1 in the example below as "\0", rather than "abcde" at the last line.

```

struct {
    char c1[5];
} s;
main()
{
    s.c1[0] = 'a';
    s.c1[1] = 'b';
    s.c1[2] = 'c';
    s.c1[3] = 'd';
    s.c1[4] = 'e';
}

```

The bug was fixed in this release.

dbx Stack Trace Bug

Previously, dbx would fail to get a correct stack trace of a program that died in the middle of any routine that did not do a link to establish a stack frame. The bug has been fixed in this release.

dbx Function Pointer Bug

Previously, the dbx command

```
print &gen_replrop
```

returned

```
&gen_replrop = gen_replrop()
```

The command now prints out the address of the function in hexadecimal, in addition to the name of the function referred to.

dbx Casting Bug

Previously, dbx would dump core rather than cast a negative floating-point constant to type double. The bug has been fixed in this release.

- Incorrect Help Bug** Previously, the `dbx help` command's description of `dbx's clear` command was wrong, implying that you could give more than one line number to be cleared. The bug has been fixed in this release.
- dbx Symbol Table Bug** In a previous release, `dbx` would dump core if you tried to debug a file without a symbol table. The bug has been fixed in this release.
- dbx Error Message Bug** Previously, `dbx` gave an incorrect error message if you tried to debug a program without a core file and tried to list the beginning of the program.
The bug has been fixed in this release.
- dbx Prefix Bug** Previously, `dbx` failed to find source files in some situations if you specified them with prefix paths. It would work properly if you did your debugging in the object files' directory. The bug has been fixed in this release.
- where Command Bug** Previously, `dbx's where` command sometimes looped when dealing with complicated expressions in FORTRAN programs with the `-f68881` flag set. Either simplifying the program or using a different floating-point option got around the problem.
The bug has been fixed in this release.
- catch and ignore Report Bug** Previously, `dbx's catch` and `ignore` commands did not properly report the status of signals that are to be caught or ignored. The bug has been fixed in this release.
- dbx clear Command Bug** Previously, if you issued `clear` while in `dbx`, you would get the following message:
- ```
<file "" was not compiled ... "-g">
```
- instead of clearing the current breakpoint.  
The bug has been fixed in this release.
- dbx "No Program" Bug** Previously, if you invoked `dbx` with no arguments, then issued `run` *with* arguments, `dbx` died with the following error message:
- ```
dbx: fatal error: bad address
```
- The bug has been fixed in this release.
- dbx Array Index Bug** Previously, `dbx` refused to index an array with a variable of type `char`, giving the following message:
- ```
subscript charvar is the wrong type
```

This type of operation is permitted in Release 3.4.

#### Private Procedure Bug

Previously, dbx would not recognize Pascal private procedures or functions, giving the error message

```
dbx: internal error: level error in symbol entry for outer
```

instead.

The bug has been fixed in this release.

#### Symbolic Link Bug

Previously, dbx would not find an executable to debug if it was symbolically linked. The bug has been fixed in this release.

#### Double Negative Bug

Previously, dbx would die if you told it to print the negative of a double or do an assign where the variable being assigned to and the value being assigned are double type and the value is negative. The bug has been fixed in this release.

#### Numeric Conversion Bug

Previously, dbx would sometimes fail to convert integers properly to floating-point. The bug has been fixed in this release.

#### FORTTRAN-related Bugs

The following FORTRAN bugs were fixed in Release 3.4.

#### Complex Number Compare Bug

Previous versions of f77 would return the following message from the loader:

```
undefined:
 Fz_eq
 Fz_ne
```

if the program being compiled compared two complex or doublecomplex numbers.

The bug has been fixed in this release.

#### INCLUDE Bug

Previously, an INCLUDE in a FORTRAN program would result in the compiler failing and dumping core. This was caused by the compiler getting confused and losing track of the name of the source file being compiled.

The bug has been fixed in this release.

#### f77 'op REG' Error Bug

Previously, f77 would sometimes generate

```
compiler error: no table entry for op REG
```

when you tried to compile a program using doublecomplex numbers.

The bug has been fixed in this release.

## Number Scaling Error Bug

Previously, the following code produced a segmentation violation at runtime if the number was 0.0 or less:

```

 write (*, " (-6pf7.3)" 0.0
 stop
 end
or
10 write (*, 10) 0.0
 format (-6pf7.3)
 stop
 end

```

The bug has been fixed in this release.

## f77 read Bug

Previously, the FORTRAN list-directed runtime routines did not correctly test for invalid input. This resulted in the `read` statement, in some cases, correctly detecting the error and stopping, but not flagging an error.

The bug has been fixed in this release.

## FORTRAN Uppercase Bug

f77 accepts uppercase source and converts it to lowercase by default. In such a case, neither `dbx` or `dbxtool` recognize uppercase if pointed at.

Using the `-U` option resulted in compiler options, such as `OPEN`, not being recognized.

The bug in `-U` operation has been fixed in this release.

## f77 read Bug

Previously, the FORTRAN list-directed runtime routines did not correctly test for invalid input. This sometimes resulted in a program dumping core after receiving invalid input.

The bug has been fixed in this release.

Bug in Assembly Phase of  
Inline Expansion

Previously, some f77 programs compiled on Sun-2s would fail during the assembly phase of inline expansion if the program called for converting a FORTRAN character variable to an integer. If the optimizer was invoked, some MC68020 instructions were included in the expanded code.

The bug has been fixed in this release by removing all non-MC68010 instructions from the files `/usr/lib/{fsoft, fswitch, fsky}.il`.

## Inline Expansion Bug

In Release 3.2, using

```
f77 -O
```

automatically resulted in inline code expansion.

The bug has been fixed in this release. Inline expansion only occurs now when an inline expansion template file is listed in the command line invocation.

**f77 -a Bug**

In Release 3.2, compiling a FORTRAN program with the `-a` flag set would cause a core dump when you tried to run the program. The bug has been fixed in this release.

**f77 Compiler Bug**

In previous releases of f77, statements of the form

```
WRITE(FRMAT4),300, 'ERR=10)NC
```

caused the optimizer to fail. The problem occurred if any variable in the control list of an I/O statement was a constant substring or had a subscript greater than 1. You received a message like the following:

```
f77 -c -O -ffpa - Compiler generated error
find_parent: couldn't find parent of T[n]
***Error code 1
```

If the `-O` option had been specified, the following error message would be generated:

```
compiler (iropt) error: find_parent: couldn't find parent of T[#]
```

This bug has been fixed in Release 3.4.

**Incorrect Handling of Backslash Characters**

Previously, f77 did not correctly handle the backslash character (`\`). It converted a double backslash (`\\`) to a single backslash, requiring you to type four backslashes (`\\\\`) for a double backslash. The compiler converted `\0` to the ASCII character `0`, which is the string terminator for C, rather than FORTRAN. This has been fixed in Release 3.4.

**Problems with Profiling Options**

Previously, f77 occasionally generated

```
as: warning (/tmp/f77pass1.1984.d.2.s:70): Undefined L-symbol
Undefined:
LPG1_158098785
```

messages from the assembler if you used the `-p` or `-pg` profiling options. The problem was related to very long jumps in the code sent to the assembler. This assembler bug was fixed in Release 3.4 and is documented in Chapter 6 of the release manual.

**Redundant I/O-List Parentheses Bug**

Previously, the compiler reported an error if an expression in the I/O list of a `write` statement was enclosed in a redundant set of parentheses.

The bug has been fixed in this release.

This bug has been fixed in Release 3.4.

- system and fork Error Bug (1003162)** Previously, the `system` and `fork` routines in the FORTRAN library could dump core at runtime.  
The bug has been fixed in this release.
- Pascal-related Bugs** The following Pascal bug was fixed in Release 3.4.
- Pascal Filename Bug** In Release 3.2, `pc` failed to compile a program if you invoked it with the `-i` option set and passed it a source filename containing a period.  
The bug has been fixed in this release.
- Library Bugs Fixed** The following bugs in the libraries were fixed in Release 3.4.
- prof.h Failure** The include file `prof.h`, described in `prof(3)`, did not work as described in the manual page. Trying to use it so resulted in “undefined loop” error messages.  
The bug has been fixed in this release.
- Missing Library Routine** In Release 3.2, the C library routine `putpwent` was in the System V library but was missing from the 4.2BSD library. `putpwent` writes out a password structure to be placed in `/etc/passwd`.  
That omission has been corrected in this release.
- System V putpwent Bug** The UNIX System V version of `putpwent` incorrectly handled negative UIDs or GIDs as unsigned quantities.  
The bug has been fixed in this release.
- System V ioctl Bug** In an earlier release, `ioctl` could set incorrect terminal modes in the UNIX System V environment if you tried to set them to support a full eight-bit data path.  
The bug has been fixed in this release.
- System V-related curses Bug** The UNIX System V version of `curses` dumped core rather than just not working correctly, if the value of term `TERM` was incorrectly set. The bug has been fixed in this release.
- Duplicate Library Routine** In Release 3.2, the `_doprnt` routine appears twice in `/lib/c.a`.  
The bug has been fixed in this release.
- Utility Bugs Fixed** The following bugs were fixed in the language-related utilities.

- tcov Failed on SIGCHLD** In previous releases, the `tcov` block count statistics coverage were incorrect if the program received ANY signal.
- During normal execution, the parent process receives a SIGCHLD when the child process exits; if *any* module in the parent was compiled with the `-a` switch, the parent immediately executes `exit(2)` and writes out the test coverage results.
- The bug has been fixed.
- lint Error Output** Release 3.2's version of `lint` sent its list of complaints to `stderr` rather than `stdout`, as did earlier versions of `lint`.
- This bug has been fixed.
- Miscellaneous Bugs** Below are miscellaneous language-related bugs that were fixed in Release 3.4.
- pxp Bug** In Release 3.3, Pascal included a new extension implementing public and private procedure declarations. However, `pxp` was not concurrently enhanced to recognize the keywords `private` and `public`, flagging them as syntax errors in a warning message.
- The bug has been fixed in this release.
- Compiler "Invalid" Options Bug** In previous releases, some invalid combinations of floating-point options were incorrectly flagged as warnings, rather than fatal errors. These combinations were `-fsky` with `-m68020` and `-ffpa` with `-m68010`.
- The bug has been fixed in this release.
- Double-Precision Transcendental Bug** In Release 3.2, double-precision trigonometric functions produced incorrect results for certain multiples of  $\pi/4$  when using the `-fsoft` or `-fsky` options. The problem does not occur with either `-f68881` or `-ffpa`.
- 7.2. Graphics Bug Fixes** This section describes bugs that have been fixed in the *Pixrect*, *SunCGI* and *Sun-Core* graphics software.
- Pixrect Bug Fixes** This section describes the bug fixes for the *Pixrect* library.
- File I/O** The following subsections describe bug fixes to the file I/O facilities of the *Pixrect* graphics library.
- Pixrect File I/O Rewrite** In Release 3.4, the `pixrect` file I/O facilities have been completely rewritten. A number of bugs, including memory, file descriptor, and process leaks have been fixed. Chapter 7 of this manual contains a new version of the *File I/O* chapter from the *Pixrect Reference Manual* that reflects changes to the `pixrect` I/O facilities.

- Type Declarations Added** In Release 3.4, type declarations of the file I/O functions have been added to the include file `<pixrect/pr_io.h>`. It is no longer necessary for a *Pixrect* application to declare them explicitly.
- pr\_load** In Release 3.4, it is no longer necessary for *Pixrect* applications to determine the colormap type of a rasterfile in order to load the file's colormap with `pr_load` or `pr_load_colormap`. These functions can automatically load any valid colormap.
- RMT\_RAW Colormaps** In Release 3.4, colormaps of the RMT\_RAW type can be dumped and loaded; they are treated as a single array of bytes. In previous releases, the RMT\_RAW type was documented and defined but unimplemented.
- Writing to a Rasterfile** In Release 3.4, `pr_dump`, `pr_dump_header`, `pr_dump_image` and `pr_dump_init` properly handle any type of *pixrect*. In previous releases, it was necessary to set the *copy\_flag* argument passed to `pr_dump` or `pr_dump_init` if there was any possibility that the *pixrect* being dumped would be something other than a primary memory *pixrect*.
- RT\_BYTE\_ENCODED Rasterfiles**
- 1) In Release 3.4, the run length encoding and decoding functions used to implement the RT\_BYTE\_ENCODED rasterfile type no longer operate in place. Using `pr_dump` to write an RT\_BYTE\_ENCODED rasterfile will always succeed and will not destroy the input *pixrect*, even if the *copy\_flag* argument is not set.
  - 2) In Release 3.4, `pr_dump_image` can write RT\_BYTE\_ENCODED rasterfiles. In previous releases, `pr_dump_image` could only write RT\_OLD and RT\_STANDARD formats.
  - 3) In Release 3.4, `pr_load` no longer fails on some correctly encoded input files.
- Rasterfile Filters**
- 1) In Release 3.4, the directories in the user's \$PATH environment variable are searched for all rasterfile filters. In previous releases, all filters except `convert.65535` had to be installed in the directory `/usr/lib/rasfilters`, and `convert.65535` would only be found in the process's current directory.
  - 2) In Release 3.4, the filter code waits indefinitely for the filter process to exit. This prevents race conditions and the accumulation of zombie processes, but may cause *Pixrect* applications to mysteriously hang if undebugged rasterfile filters are installed in `/usr/lib/rasfilters` or other shared directories.
- lint Library** The following section describes bug fixes to the `lint(1)` facilities of the *Pixrect* graphics library. You should rerun `lint(1)` on all *Pixrect*-based applications.
- 1) In Release 3.4, the *Pixrect* `lint(1)` libraries (`llib-lpixrect` and `llib-lpixrect.ln` in `/usr/lib/lint`) and the header file `<pixrect/pixrect.h>` have been improved so that the types of arguments to all *pixrect* functions and macros will be checked.



- 2) The arguments to `pf_text` in the `lint(1)` library have been changed to match the documented calling sequence.
- 3) Errors in the arguments for `pf_textbatch` and `pr_load_std_image` have been corrected.
- 4) New entries have been added for the GP support functions `gpl_alloc`, `gpl_post`, and `gpl_sync`, and for the new text functions `pr_text`, `pr_ttext`, `prs_text`, and `prs_ttext`.

## Macros

The following section describes bug fixes to macros in the *Pixrect* graphics library.

- 1) In Release 3.4, the `prs_replrop` macro was added to the `<pixrect/pixrect.h>` include file. In previous releases, this macro was documented but not present.
- 2) A spurious semicolon following the declaration of `prs_destroy` was deleted.
- 3) `pr_close` is a macro as documented, instead of a function, as it was in previous releases.

## rop\_fastloop Macro

The `rop_fastloop` macro in the include file `<pixrect/pr_util.h>` is conditionally defined to generate an unrolled loop only on 68010 systems, and the supporting `cases8` macro is only defined in that case. Rolled loops usually provide better performance on systems with an instruction cache, including all 68020 systems.

## Memory Pixrects

The following sections describe bug fixes to the memory *pixrect* facilities of the *Pixrect* graphics library.

### Private Data Format for Memory Pixrects

A new memory *pixrect* private data format has been defined, which includes a bit plane mask to support the `cgfour(4S)` color memory frame buffer. The document *Release 3.2 Manual for the Sun Workstation* contains a more detailed description of the `cgfour(4S)` frame buffer. All memory *pixrect* functions have been modified to support this new format, which is defined by `struct mprp_data` in `<pixrect/memvar.h>`. The plane mask is normally accessed with `pr_putattributes` and `pr_getattributes`. Most *pixrect* client programs will not have any reason to directly create memory *pixrects* with plane masks. But if this is necessary for some special purpose, the function in Figure 7-1 should be used.

Figure 7-1 *Memory Pixrect Program Example*

```

#include <sys/types.h>
#include <pixrect/pixrect.h>
#include <pixrect/pr_util.h>
#include <pixrect/memvar.h>

Pixrect *mem_create_with_planemask(w, h, depth)
int w, h depth;
{
 Pixrect *pr;
 struct mprp_data *mprd;

 /*
 * Create a normal memory pixrect with no plane mask,
 * replace its private data (struct mprp_data) with
 * a struct mprp_data, set the plane mask flag,
 * and initialize the plane mask.
 */
 if (pr = mem_create(w, h, depth))
 if (mprd = alloctype(struct mprp_data)) {
 mprd->mpr = *mpr_d(pr);
 free(mpr_d(pr));
 pr->pr_data = (caddr_t) mprd;
 mprd->mpr.md_flags |= MP_PLANEMASK;
 mprd->planes = ~0;
 } else {
 pr_destroy(pr);
 pr = 0;
 }
 return pr;
}

```

### 8, 16, and 32-bit Memory Pixrect Support

In Release 3.4, support for memory pixrects with a depth other than 1-bit has been improved. All pixrect functions except `pr_vector` support 16 and 32-bit memory pixrects. `pr_line` can be used to draw vectors in these pixrects.

### `pr_vector`

- 1) In Release 3.4, vectors drawn into memory pixrects with `pr_vector` properly wrap polygons drawn with `pr_polygon_2`.
- 2) In Release 3.4, drawing vertical vectors into an 8-bit memory pixrect is reliable. In previous releases, `pr_vector` sometimes scribbled on random memory, often causing a segmentation violation.

### `pr_get`

In Release 3.4, `pr_get` returns the correct result when applied to a reverse video memory pixrect.

- `pr_put`
- 1) In Release 3.4, using `pr_put` to write to a pixel in a memory `pixrect` correctly truncates its *value* argument as documented. In previous releases, any non-zero argument caused a *value* of 1 to be written to a 1-bit memory `pixrect`.
  - 2) In Release 3.4, using the `pr_put` function with the *value* argument set to 0 on a reverse video memory `pixrect` works properly. In Release 3.2, *value* was always treated as 1.
- `pr_rop`
- 1) In Release 3.4, the semantics of operations involving a 1-bit source `pixrect` and an 8, 16, or 32-bit destination `pixrect` are consistent with the documentation and the behavior of `cgone` `pixrects`.
  - 2) In Release 3.4, use of the `op PIX_NOT(PIX_DST)` with a non-NULL source `pixrect` is handled properly. In previous releases, it was treated as `PIX_DST`.
  - 3) In Release 3.4, operations involving small, 8-bit memory `pixrects` are reliable.
  - 4) In Release 3.2, when `pr_rop` was called with a 1-bit memory `pixrect` destination and NULL source, the least significant bit of the color encoded in the `op` was used to determine whether the source value should be 0 or 1. In Release 3.4, any non-zero color is treated as 1. This is consistent with pre-3.2 releases and the other memory `pixrect` functions.
  - 5) In Release 3.4, using the `pr_rop` function on a 1-bit source and a narrow (16 bits-per-line) 8 or 16-bit memory `pixrect` destination is reliable.
- Text
- The following sections describe bug fixes to the text facilities of the *Pixrect* graphics library.
- Memory Leaks with `pf_open`
- In previous releases, a memory leak affecting repeated calls to `pf_open` and `pf_close` has been fixed. In Release 3.4, `pf_close` releases all resources allocated by `pf_open`.
- `pr_close`
- In previous releases, if `pf_close` was given a font argument which referred to a resident (compiled in) font that was not the default font, it would try to deallocate that font's data. This caused segmentation violations some time. In Release 3.4, any valid font can be passed to `pf_close`.
- Opening `vfont(5)` Fonts
- In Release 3.4, when `pf_open` loads a `vfont(5)` file, it normally sets the default character width to the width of (lowercase) `a` and the default character height to 1.5 times the height of (uppercase) `A` above the baseline. If `a` or `A` are not present in the font or have zero *width* and *up* values respectively, the `vfont(5)` header *maxx* and *maxy* values are used instead. In previous releases, `pf_open` would fail if `a` and `A` were present but had zero *width* or *up* values, making it impossible to load some valid fonts (primarily symbol and rotated fonts).

- pf\_ttext String Lengths** In Release 3.4, the `pf_ttext` function properly displays text strings of any length. In previous releases, the 81st and subsequent characters were positioned incorrectly.
- Miscellaneous** The following sections describe miscellaneous bug fixes to the *Pixrect* graphics library.
- pr\_getcolormap with bwone Pixrects** In previous releases, `pr_getcolormap` returned incorrect values when applied to a `bwone` pixrect. In Release 3.4, this bug has been fixed and the colormap behavior of `bwone` pixrects is identical to that of `bwtwo` pixrects.
- pr\_batchrop** In Release 3.4, any valid pixrect can be passed as a source pixrect to `pr_batchrop`. In previous releases, use of anything other than a primary 1-bit memory pixrect caused undefined results in some cases.
- Global Definitions Removed from pr\_traprop** In previous releases, several utility functions were inadvertently declared globally in the source file for `pr_traprop`, sometimes causing collisions with user-defined names when this function was linked into a pixrect program. The most notable example was a function called `init`. In Release 3.4, the utility functions are `static`.
- pr\_vector** In previous releases, `pr_vector` ignored the most significant 16 bits of the coordinates passed to it. In Release 3.4, it performs clipping calculations on the full 32-bit values. (Most other pixrect functions will not function properly when given coordinates that cannot be represented in 16 bits.)
- pr\_polypoint** In Release 3.4, the `pr_polypoint` function produces correct results on `bwone` pixrects for all `op` values. This function is described in the document *Release 3.2 Manual for the Sun Workstation*.
- pr\_polygon\_2**
- 1) In Release 3.4, the `pr_polygon_2` function replicates the source pixrect as necessary to tile an entire polygon. It is no longer necessary for the source pixrect to be as large as the destination area covered by a polygon drawn by `pr_polygon_2`.
  - 2) In Release 3.4, when `pr_polygon_2` returns an error, it will consistently be `PIX_ERR` rather than various small integer values.
  - 3) In Release 3.4, the `pr_polygon_2` function produces correct results when the destination is a GP pixrect and the source is not a 1-bit memory pixrect. In Release 3.2, pixels on the right edge of the destination were not drawn reliably, and other types of source pixrects gave indeterminate results.
- pr\_replrop** The `pr_replrop` function was rewritten to handle all combinations of destination and source coordinates properly, even if clipping is disabled, and to accept NULL source pixrects.

- `pr_rop` In Release 3.4, the semantics of operations with `pr_rop` involving a 1-bit source `pixrect` and a `cgtwo` or `gpone` destination `pixrect` are consistent with the `pixrect` documentation and the behavior of `cgone` and memory `pixrects`. In previous releases, each destination pixel was operated on using a value of either 0 or 255 (corresponding to 0 and 1 source `pixrect` values), then logically AND-ed with the color encoded in the `op`.
- `pr_stencil`
- 1) In previous releases, `pr_stencil` operations from an 8-bit memory source `pixrect` to a `cgtwo` or `gpone` destination `pixrect` did not always correctly draw pixels on the right edge of the destination `pixrect`. This bug has been fixed in Release 3.4.
  - 2) In previous releases, `pr_stencil` did not bother to synchronize with the GP when the source was a memory `pixrect`, sometimes causing spectacular screen corruption. This bug has been fixed in Release 3.4.
- `pr_line`
- 1) In Release 3.2, `pr_line` and `pr_polyline` could overwrite an internal buffer in rare circumstances when drawing horizontal un-clipped textured vectors. This bug has been fixed in Release 3.4.
  - 2) In previous releases, attempting to draw clipped textured polylines with the *balance* option set produced improper textures. This bug has been fixed in Release 3.4.
  - 3) In previous releases, single segment (solid) textured vectors were not always drawn. This bug has been fixed in Release 3.4.
- Frame Buffer Device Drivers** The following sections describe bug fixes to the frame buffer device drivers.
- `gpone (4S)` In Release 3.4, the `gpone(4S)` driver zeros the static blocks handed out by the `GP1IO_GET_STATIC_BLOCK ioctl`.
- `FBIIOGVIDEO ioctl` In Release 3.4, the `FBIIOGVIDEO ioctl` always returns the correct video enable status for all frame buffers except the `bwone(4S)`. In Release 3.2, it was also possible for the software state to disagree with the hardware state for `bwtwo(4S)` and `cgfour(4S)` frame buffers.
- `cgfour 4S Support for the Sun-3/60 Frame Buffer` In Release 3.4, the `cgfour` `pixrect` driver includes support for the Sun-3/60 color frame buffer.
- Graphics Utilities** This section describes the bug fixes for various graphics utility programs.
- `rasfilter8to1(1)`
- 1) In previous versions, the right edge of the output image from `rasfilter8to1(1)` appeared incorrectly. This has been fixed in Release 3.4.
  - 2) In previous releases, `rasfilter8to1(1)` would generate an incorrect output file if the input file width was odd. In Release 3.4, this bug has been fixed.

rastrepl(1) and  
RT\_BYTE\_ENCODED  
Rasterfiles

In previous releases, `rastrepl(1)` could not read 8-bit deep `RT_BYTE_ENCODED` rasterfiles. This bug has been fixed in Release 3.4.

SunCGI Bug Fixes

Release 3.4 contains the following bug fixes for the *SunCGI* software.

Resizing Viewports

In previous releases, if a viewport was made smaller than a clip rectangle, and then made larger, a smaller viewport would remain. This has been fixed in Release 3.4.

NOCLIP

*Warning:* When clipping is disabled in *SunCGI*, `pixwin` clipping is also disabled. Therefore, a *SunCGI* application can write outside the display area and produce unpredictable results, including system errors.

Clear View Surface

When using `clear_view_surface` with `clear_control` set to `clip_rectangle(8000, 9000, 10000, 11000)`, *SunCGI* would draw a large rectangle instead of clearing the specified area. This has been fixed in Release 3.4.

Cvwsurf

- 1) In Release 3.0, it was possible to create a new View Surface Tool by setting the `flags` field of the `Cvwsurf` structure. A bug was introduced in Release 3.2 that prevented this. This has been fixed in Release 3.4.
- 2) In previous releases, *SunCGI* would not allow an application to set the `ptr` field of the `Cvwsurf` structure to one of the advertised strings. This has been fixed in Release 3.4.

Quitting a View Surface Tool

After opening a view surface tool with a system call, the view surface tool would not close without selecting the `Quit` item from the *Frame* menu. This has been fixed in Release 3.4.

Pixwin Output

Previously, *SunCGI* relied on an asynchronous event (a `SIGWINCH` signal), to gain access to the output `pixwin`, which sometimes caused a core dump. This has been fixed in Release 3.4.

CGIPW and SunView

- 1) A mismatch between *CGIPW* and *SunView* use of `pixwin` regions prevented the integration of *CGIPW* on a *SunView* canvas that had scrollbars. This has been fixed in Release 3.4.
- 2) Problems prevented the proper use of *CGIPW* on a *SunView* canvas with a retained backing larger than the visible screen window area. This has been fixed in Release 3.4.

Request Input

In previous releases, `request_input` and `await_event` did wait-polling while awaiting an input trigger or timeout. This would cause intense CPU resource use during these calls. In Release 3.4, `request_input` and `await_event` use the `select(2)` mechanism to block, thus freeing the CPU until either the input trigger fires, or the timeout expires.

|                                             |                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| cfmksizepecmode                             | The FORTRAN library was missing the <code>cfmksizepecmode</code> function cited in Appendix G of the <i>SunCGI Reference Manual</i> . This function has been replaced by <code>cfmksizepecmode</code> in Release 3.4.                                                                                                                                                                                    |
| GP View Surfaces                            | When more than one <i>SunCGI</i> view surface was opened on a GP surface, a GP-specific attribute structure was freed, affecting subsequent view surfaces. This bug affected blanket windows and <code>view_surface</code> tools, but did not occur in <i>CGIPW</i> mode. This has been fixed in Release 3.4.                                                                                            |
| Rectangle Perimeters                        | In Release 3.2, perimeters drawn with a non-zero window offset in <i>x</i> or <i>y</i> by <code>rectangle</code> appeared at locations other than those specified. This has been fixed in Release 3.4.                                                                                                                                                                                                   |
| Arc                                         | Arcs having common <i>y</i> -value endpoints would sometimes invert the arc from the intended side of the points. Arcs are drawn correctly in Release 3.4.                                                                                                                                                                                                                                               |
| Freed Memory with Await Event               | The <code>await_event</code> function returned a pointer to freed memory, thus causing problems when making calls to input devices such as <code>IC_STRING</code> , <code>IC_STROKE</code> , <code>IC_LOCATOR</code> , and <code>IC_PICK</code> . This has been fixed in Release 3.4.                                                                                                                    |
| Multiple Character Keyboard Events in CGIPW | In <i>CGIPW</i> mode, multiple-character keyboard events were not immediately delivered when using Canvases. This has been fixed in Release 3.4.                                                                                                                                                                                                                                                         |
| Inquire Device Identification               | The <code>inquire_device_identification</code> function returned a null string instead of a device type. This has been fixed in Release 3.4.                                                                                                                                                                                                                                                             |
| View Surface Table                          | <i>CGIPW</i> functions used in conjunction with output primitives and attribute settings in Release 3.4 more accurately and efficiently search the view surface table.                                                                                                                                                                                                                                   |
| Obsolete Code Removed                       | In Release 3.4, obsolete <i>SunCGI</i> code from previous releases has been removed to improve maintenance.                                                                                                                                                                                                                                                                                              |
| Close View Surface                          | The <code>close_vws</code> function did not change from VSAC (at least one view surface active) or VSOP (at least one view surface open) to CGOP (CGI open, no view surfaces open) when the last view surface was freed.                                                                                                                                                                                 |
| Deactivate View Surface                     | In Release 3.4, the <code>deactivate_vws</code> function allows the view surface mode to change, so that a deactivated view surface will display output primitives if there is only one active workstation. Likewise, the <code>activate_vws</code> function now changes the view surface mode, so that an <i>open</i> view surface will display output primitives if it is the only active workstation. |
| CGIPW Validity Check                        | In Release 3.4, <i>SunCGI</i> checks the validity of <i>CGIPW</i> function descriptors to avoid segmentation violations and inadvertent writes to memory.                                                                                                                                                                                                                                                |

|                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CGIPW Set VDC Extent                           | The <code>cgipw_set_vdc_extent</code> function worked correctly with only one view surface in use. In Release 3.4, it works with more than one surface in use.                                                                                                                                                                                                                                                                                                                                   |
| Scaling                                        | Scaling failed to vary line width, perimeter width, and marker sizes when using VDC coordinates and multiple view surfaces. This has been fixed in Release 3.4.                                                                                                                                                                                                                                                                                                                                  |
| Inquire Text Extent                            | The <code>inquire_text_extent</code> function attempted to display output in inactive surface areas. This has been fixed in Release 3.4.                                                                                                                                                                                                                                                                                                                                                         |
| Transparent Text                               | In Release 3.4, text transparency now behaves as it did in Release 3.0.                                                                                                                                                                                                                                                                                                                                                                                                                          |
| FORTTRAN Input                                 | In previous releases, use of any of <i>SunCGI</i> input function through a FORTRAN compatibility function would result in a segment violation. In Release 3.4, the FORTRAN input mechanisms work properly.                                                                                                                                                                                                                                                                                       |
| VALUATOR Input with Await Event                | In previous releases, the receipt of the first input event when a VALUATOR logical input device is in EVENT mode disables the LID. The LID will be left in EVENT mode in Release 3.4.                                                                                                                                                                                                                                                                                                            |
| Ignored Input with Await Event                 | In previous releases, <code>await_event</code> would return the initial value for CHOICE, VALUATOR, or PICK logical input devices. This has been fixed in Release 3.4.                                                                                                                                                                                                                                                                                                                           |
| Bad Data Structure Error Code with Await Event | In previous releases, when an application passed a <code>Cinrep</code> structure with an invalid <code>measure</code> field to <code>await_event</code> , <i>SunCGI</i> would return the error code <code>EINQOVFL</code> (Input Queue Overflow) instead of <code>EBADDATA</code> . This has been fixed in Release 3.4.                                                                                                                                                                          |
| SunCore Bug Fixes                              | Release 3.4 contains the following bug fixes for the <i>SunCore</i> software.                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Fat Vectors                                    | <ol style="list-style-type: none"><li>1) In previous releases, the thickness of fat vectors drawn with <i>SunCore</i> would vary with the slope of the vector. While it is not possible to get exactly the same line width on raster displays, in Release 3.4 vectors have significantly more uniform widths regardless of the slope.</li><li>2) Fat vectors drawn on GP surfaces with <i>SunCore</i> would not be completely removed when erased. This has been fixed in Release 3.4.</li></ol> |
| raster Structure                               | The <code>raster</code> structure used in <code>put_raster</code> and <code>get_raster</code> is now defined in <code>&lt;usercore.h&gt;</code> . In Release 3.4, a <i>SunCore</i> application does not have to explicitly define this structure.                                                                                                                                                                                                                                                |
| put_raster                                     | In previous releases, <code>put_raster</code> in <i>SunCore</i> would invert the raster image on Sun-1 monochrome displays. This has been fixed in Release 3.4.                                                                                                                                                                                                                                                                                                                                  |



|                                       |                                                                                                                                                                                                                                                                 |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Clipped Vectors with GP View Surfaces | In <i>SunCore</i> on the 3.2 release, clipped vectors did not show up on GP view surfaces. This problem in the GP microcode has been fixed in Release 3.4.                                                                                                      |
| Polygon Vertex Limit                  | In previous releases, polygons in <i>SunCore</i> were limited to 100 vertices. This limit has been changed to allow up to 200 vertices in Release 3.4.                                                                                                          |
| Backspace Key                         | In previous releases, <code>await_keyboard</code> would use the <code>[CTRL-H]</code> as backspace, which would have differed from the previous terminal setting. In Release 3.4, <i>SunCore</i> uses the current terminal setting for the backspace character. |

### 7.3. SunView Bug Fixes

Release 3.4 has several bug fixes made to SunView since Release 3.2. This section first describes bug fixes that are visible to users of the system, followed by bug fixes that affect SunView programmers.

#### Defaults, `.*rc` Files, and Filters

Various bug fixes have been made in these categories, so you may find that something that did not work as expected before now works, or possibly vice versa. It is worth checking your `~/.*rc` files.

#### `.textswrc` Processes Shell Metacharacters

In the `~/textswrc` file itself, in the command line following the `FILTER` line, regular shell metacharacters such as `$`, `"`, and `'` are now processed by the shell. This shell is the one defined by your `SHELL` environment variable (usually `/bin/csh`); if the environment variable does not exist the shell used is `/bin/sh`. For example,

```
FILTER R13
/bin/echo $HOME
```

in previous releases would put the actual characters `"$HOME"` in the text subwindow when you press `[R13]`; now it inserts the path to your home directory as it would if you typed

```
/bin/echo $HOME
```

to your shell.

#### CAUTION

This means shell metacharacters you were using before must be escaped following the normal `"\"` conventions for your default shell. The sample text subwindow filters file, `/usr/lib/.textswrc`, has been updated; you can use this as a guide.

#### Fixes to Input Filters

The input filters `capitalize` and `insert_brackets` (documented in the *textedit\_filters(1)* manual page) have been improved. You usually use these filters by assigning them to function keys with the file `~/textswrc`, then invoking them in text subwindows.

`capitalize` no longer blanks out numeric characters.

Both `capitalize` and `insert_brackets` can handle larger selections. In order to do this, they process standard input in a temporary file, called

`/tmp/capProcessId` or `/tmp/insProcessId`. If the temporary file cannot be opened or deleted, each prints an error and copies standard input to standard output, so that if the selection is pending delete, it does not just disappear.

Previously, `insert_brackets` would dump core if it was not invoked with the right arguments. It now prints out

```
Usage: insert_brackets leftbracket rightbracket
```

on standard error (the console, if you invoke it in a `textedit` started from the root menu) and writes standard input to standard output unchanged. (That is, in a text subwindow, the pending delete selection is replaced by itself.)

Before Release 3.4, `insert_brackets` did not recognize all the escape character conventions in C. Now it supports the full rules (in *The C Programming Language* by Kernighan and Ritchie) for inserting control characters into strings; specifically, `\n`, `\t`, `\b`, `\r`, `\f`, `\\`, `\'`, and `\octal_number` are all valid escape sequences.

**NOTE** Check your `~/ .textswrc` files to see if you need to modify them to work under this scheme. For example, the old filter definition

```
R1 FILTER
insert_brackets \fI \fP
```

becomes

```
R1 FILTER
insert_brackets "\\fI" "\\fP"
```

The quotes cause the shell to leave the backslashes alone. `insert_brackets` will then change each pair of backslashes into a singlebackslash.

#### `.ttyswrc` String Changes

The similar file to map function keys in tty subwindows is `~/ .ttyswrc`. Keys can send longer strings, now up to 1024 characters.

In previous releases, you had to preface the `:` character with a `\` to output it; in 3.4, you can output colons without “quoting” them this way. You specify other special characters in strings in the `~/ .ttyswrc` file in the same way you do for strings in `termcap`; see `termcap(5)` for details.

#### Blank lines in `~/ .ttyswrc`

In Release 3.2, blank lines in a `.ttyswrc` file caused

```
unknown command
```

error messages to be printed when a tty subwindow was created. Now, blank lines are treated as comment lines, as they were in 3.0.

Mail Aliases in  
defaultsedit

In previous releases, if your `~/ .mailrc` file included long aliases, `defaultsedit` would not read them in successfully. So, when you saved your defaults, the long aliases were lost. This bug has been fixed in 3.4. Note, however, that the problem can recur in client programs that use the SunView defaults database and are not relinked with the new SunView libraries.

`.suntools` Can Accept Blank  
Lines

In previous releases, blank lines in `.suntools` would interfere with the automatic tool startup `suntools` provides. You can now have blank lines in your `.suntools` file.

### Other Bug Fixes Visible to the User

Cursor Correct on Sun-3/110

In Release 3.2, when opening an iconic window on the Sun-3/110LC display, the screen area under the cursor was sometimes erroneously set to display the color planes instead of the overlay plane.

`lockscreen -e` on the Sun-  
3/110LC Color Display

`lockscreen(1)` with its `-e` option no longer crashes the system.

'UnZoom' when Iconic

Selecting 'Resize  $\Rightarrow$  UnZoom' from the frame menu did not to work for iconic tools. This has been fixed in Release 3.4.

Characters Lost on Multi-  
character Input

In previous releases, if you held down keys mapped into multiple characters (such as the arrow keys<sup>6</sup>), some of those characters could sometimes be lost. Now, the kernel's input event queue expands to cope with the extra events, and shrinks down to a smaller size when the system has caught up.

Bug Fixes in `cmdtool`

In addition to the new functionality added to `cmdtool` (see *Improvements to cmdtool* above), there are several bug fixes to `cmdtool` in 3.4.

'Reset' in `cmdtool`

After selecting the 'Reset' menu item in `cmdtool`, you were unable to use the line editing keys (`Edit_back_char`, `Edit_back_word`, and `Edit_back_line` in the *Text* category of `defaultsedit`) until you typed a new line. This has been fixed.

Typing Ahead

Typing characters faster than the system was displaying them would create a false input buffer overflow condition. This has been fixed.

SIGIO Signals in `cmdtool`

The SIGIO signal is now delivered to the process running in `cmdtool`, even if that process has not emptied its input stream.

---

<sup>6</sup> The arrow keys can be mapped to emit SunView function key codes instead of cursor motion codes by setting `Arrow_Keys` in the *Input* category of `defaultsedit` to *False*.

## Bug Fixes in `textedit` and the Text Subwindow

**Note:** You may need to change your `.suntools` or `rootmenu` files along with other aliases to reflect this bug fix.

In addition to the new features added to the text subwindow, (see *Text Subwindow Enhancements* above), there are several bug fixes to `textedit`, and the text subwindow in general, in Release 3.4.

### `textedit` Width Set Correctly

In prior releases, the `-Ww nnn` flag to `textedit` would not take the size of the scrollbar into account, so that to make, say, a 60-column window you would have to set the width slightly larger. Now the width you supply is the number of columns in the typable portion of the text subwindow.

### User-Supplied Icon Label

`textedit` usually changes its icon's label to show the file being edited. In 3.4, you can supply your own icon label using the standard `-WLlabel` frame command line option.

### 'Store to named file' and 'Get from file' with Null Selection

'Store to named file' in the text subwindow menu requires you to select the file name to store to first. In previous releases, if you chose 'Store to named file' (or its variants 'Store & Quit' and 'Close & Store') when there was no selection, it would sometimes save the file to a bogus filename. Now it prints a detailed message asking you to select the desired filename. The same message appears if you choose 'Get from file' with a null selection.

### 'Get from file' with Non-existent Filename

The text subwindow package no longer prints

Stuff from file failed due to INTERNAL ERROR

when it cannot find the filename selected for 'Get from file.'

### Scrollbar Bubble More Accurate

If the text subwindow's contents ended without a newline, the scrollbar bubble's position would indicate text remained below the window's contents, even if the window showed the end of the file.

Also, the scrollbar bubble's position is now updated after Delete, Undo, and Get operations.

### Changing 'Line break' Mode

The caret now positions correctly after changing between 'Clip lines' and 'Wrap at character' modes.

### Text Subwindow Loses Track of Selection

It was possible to confuse the Selection Service during a secondary Get by letting up on the `[Get]` key before releasing the mouse button; this has been fixed.

### Highlighting Selections

In some cases, selecting `[Tab]` characters along with other characters would damage the screen image of characters; this is fixed in Release 3.4.

### Programs Exiting in shelltool and cmdtool

The message printed out when the program running in a `shelltool` or `cmdtool` exited with non-zero status used to be the following:

```
child of tty exited with return code nn
```

or

```
child of cmdtool exited with return code nn
```

The message has been changed to be more informative, either

```
A tty window has exited because its child exited.
Its child's process id was pid and it exited with return code nn
```

or

```
A command window has exited because its child exited.
Its child's process id was pid and it exited with return code nn
```

### Not Enough Swap Space in shelltool and cmdtool

Sometimes you can run out of swap space so that `shelltool` or `cmdtool` starts up, but it cannot start the program running in itself (usually this is a shell). In previous releases, the `shelltool` or `cmdtool` would just hang in this situation. Now the tool remains active and prints

```
Out of swap space. Cannot continue.
```

You can then 'Quit' the tool from its frame menu, and kill other processes until you have enough memory to try again.

### Fixes to Tty Subwindows

These bug fixes apply to programs that use tty subwindows, such as `shelltool` and `cmdtool` in tty mode.

#### clear in a Tty Subwindow

In previous releases, when a tty subwindow was cleared by running `clear(1)`, it did not reset its count of how many lines are currently displayed in itself, leading to problems when 'Page Mode' was enabled in `shelltool`.

#### Long Input Lines in Tty Subwindows

An unavoidable kernel constraint makes tty subwindows lock up if you insert or type a line longer than 255 characters into them. In Release 3.2, a 'Flush' menu item was added to the tty subwindow menu, allowing you to flush the entire input line. In Release 3.4, in addition to the menu item, a warning message pops up as you type. You can choose to flush the input queue or ignore the error condition and work in another window.

#### Highlighting Word Selections

Word selection in reverse video fields in `shelltool` highlights correctly in Release 3.4.

**Security Hole in  
lockscreen Fixed**

In previous releases, the password control in `lockscreen(1)` could be circumvented by using the `[Open]` (usually `[L5]`) function key to pop up a window above it briefly. The key is now disabled in `lockscreen`'s input mask.

**Miscellaneous Fixes to Tools****Changes in `defaultsedit`**

`defaultsedit` would sometimes lose changes made to the database when changing between categories. It now prompts you to save changes when you switch categories.

**Invalid Options for `perfmeter`**

In previous releases, if `perfmeter(1)` was invoked with an invalid option, it would never appear on the desktop but invisibly chew up CPU time. It now prints out a 'Usage' string when given an invalid option.

**`clock` Test Mode**

In Release 3.2, if you ran the `clock` in test mode (`-t`), it would print out

```
Notifier error: Unknown client
```

This has been fixed.

**Large Fonts in `chesstool`**

The `chesstool` game now works with large fonts.

**User Interface Cleanup**

The following bugs were fixed to make tools conform with Appendix B of the *SunView Programmer's Guide, Sun User Interface Conventions*.

- The cursor in `defaultsedit` and `mailtool` should change to an hourglass when either tool isn't responding to user input. It now does.
- In `iconedit`, the cursor would change to an hourglass, but the frame header would not change. The frame header now changes to show that `iconedit` is busy.

**Bug Fixes Visible to the  
Programmer**

The remainder of this section describes bug fixes visible to the programmer in Release 3.4. Some of the fixes are related to errata and addenda for the SunView programmer's manuals, given in Chapter 7.

**Window Geometry Fixes**

Several fixes have been made to window sizing and panel item layout. If you have programmed around bugs in window geometry, or if you are relying on anomalous behavior in Release 3.0 or 3.2, you should pay particular attention to this section. Also read the addenda relating to panel layout bugs, *Cautions for Default Panel Item Layout*, in Section 12.1, "Errata and Addenda for the SunView Programmer's Guide."

**`WIN_ROWS` When There is No  
Frame Header**

The value returned by the `WIN_ROWS` attribute for a frame used to be incorrect if you turned off the frame header (using the `-Wn` frame command line option or by setting `FRAME_SHOW_LABEL` to `FALSE`). This has been fixed.

- Panel Choices Too High** In Release 3.2, if the `PANEL_DISPLAY` mode for a panel choice item was `PANEL_CURRENT`, the current choice would be positioned a few pixels too high relative to the label. Release 3.4 restores the correct behavior from 3.0.
- “Next” Panel Item Created Too Low** When creating items in a panel subwindow in Release 3.2, if the position of the next item was not specified when it was created, it would sometimes be located a few pixels lower than the preceding item. This is fixed in 3.4.
- Choice Item Positioning** Choice Label baselines did not line up with choices by default. This has been fixed.

### Other Fixes Visible to the Programmer

- Miscellaneous Text Subwindow Fixes** **Scrollbar Bubble Positioning**  
In previous releases, the `TEXTSW_FIRST` and `TEXTS_FIRST_LINE` attributes did not set the scrollbar bubble. This is fixed in this release. Since `TEXTSW_FIRST` now sets the scrollbar mark, it makes `textsw_normalize_view()` unnecessary.

#### Read-Only Text Subwindow

You can make the file in a text subwindow read-only at the same time that you set the filename; that is, the following works:

```

window_set(textsw,
 TEXTSW_FILE, filename,
 TEXTSW_READ_ONLY, 1,
 0)

```

Before, you had to do this in two separate operations.

#### Text Subwindows with Scrollbars

Text subwindows are created with a scrollbar by default. In 3.0, you could not explicitly set the scrollbar for a text subwindow. In 3.2, you could create the scrollbar, but not specify its thickness. Now you can do both, but you must specify the `SCROLL_DIRECTION` of the scrollbar when you create it — see *Creating Scrollbars* in Section 12.1, “Errata and Addenda for the SunView Programmer’s Guide.”

Also, Release 3.4 has fixed strange behavior that occurred in previous releases if you explicitly created a text subwindow with a horizontal scrollbar.

#### Destroying the Scrollbar

In previous releases, the text subwindow would not always destroy the old scrollbar and adjust the text painting rect when you specified your own scrollbar.

#### `TEXTSW_RIGHT_MARGIN` Works

The attribute `TEXTSW_RIGHT_MARGIN` now works properly in text subwindows.

### Text Subwindow Storage

In previous releases, when a text subwindow was destroyed, it would not free the storage associated with `Again` logging. This has been fixed.

### Timer Events in Text Subwindows

The text subwindow package uses timer events to blink the caret. When a text subwindow was destroyed in earlier releases, sometimes the destroy event would be delivered while a timer event was still pending. The blink caret routine would then fail and dump core because the text subwindow no longer existed. The text subwindow now removes the timer when the last text subwindow view is destroyed.

### Pop Up Subframe Restrictions

You can use `window_loop()` to display a subframe containing a canvas. Prior to this release, you could only put a panel in subframes displayed in this manner.

These restrictions do not apply if you set `WIN_SHOW` to `TRUE` to display the subframe. `window_loop()` is usually used to prompt the user to acknowledge some event; `WIN_SHOW` is usually used to display secondary information or a *property sheet* while the user continues to work in the main frame.

Other restrictions on `window_loop()` (described in more detail in *Restrictions on Pop Up Windows* in Section 12.1, "Errata and Addenda for the SunView Programmer's Guide") remain:

- Pop up subframes displayed with `window_loop()` can only have one subwindow.
- You still cannot have a text subwindow in a pop up displayed with `window_loop()`.

### `WIN_FONT` Now Works for Tty Subwindows

The attribute `WIN_FONT` now correctly sets the font of a tty subwindow. `WIN_FONT` still does not work for frames. You cannot create a text subwindow with a given font; you have to create the text subwindow, then use `window_set()` to set the font. Variable width fonts only work for frames, canvases, and text subwindows.

### `event_set_time()`

There should have been a macro to set the timestamp of an event, to complete the `event_set_*(event, *) set`. Now there is, `event_set_time(event, time)`.

### `KEY_BOTTOMFIRST` and `KEY_BOTTOMLAST` Defined

In previous releases, the SunView event macro `event_is_key_bottom()` used the nonexistent constants `KEY_BOTTOMFIRST` and `KEY_BOTTOMLAST`. It now uses the existing *void* codes `KEY_BOTTOMLEFT` and `KEY_BOTTOMRIGHT`.

### Diagonal `pw_copy()` Fix

Diagonal `pw_copy()` calls made two image in previous releases.

### `pw_line()` in a Canvas

The routine `pw_line()` would draw in the coordinate space of the canvas subwindow. If the canvas was larger than the subwindow and scrolled, this would produce the wrong effect. `pw_line()` now draws in the coordinate space of the canvas itself.



**Click\_to\_Type in Panels**

In *Click\_to\_Type* mode (in the *SunView* category in `defaultsedit`) you could not set the keyboard focus to a panel with no text items, even though some of the items might have keyboard shortcuts. In Release 3.4, if `PANEL_ACCEPT_KEYSTROKE` is set, the panel will accept the keyboard focus.

**Cursor Positioning Escape Sequence in Tty Subwindows**

(Escape sequences for the Sun console and tty subwindow are documented in the `console(4S)` manual page.)

**Note:** The work-around for this bug was to modify the `cm` value in the Sun `termcap` entry. You do not need to apply the work-around any more.

The cursor positioning (CUP) escape sequence `ESC[#I;#JH` takes character position and line as parameters. If you omit the parameters, they are supposed to default to 1. In previous releases, the tty subwindow would sometimes “remember” the previous character position, so that if the character position was not specified, the cursor would not move to the beginning of the line.

**Odd-Sized Icons**

In previous releases, if you supplied a tool with an icon that was not the standard size (64 x 64 pixels) the icon would not display properly. This has now been fixed, so that you can use smaller or larger icons. However, icons still initially appear aligned on an invisible 64-pixel grid unless you specify their position with the `-WPxy` frame command line argument. `iconedit` produces only 64 x 64 or 16 x 16 patterns; you have to edit its output by hand for other sizes.

**Internal Name Changes**

The names of some routines used internally have been changed to reduce the likelihood of conflict with client names. Unless you have *SunView* source code, you should not use these internal names.

`draw_box` renamed

The routine `draw_box()` in `libsuntool` has been replaced by `_tool_draw_box()` to avoid possible name conflicts.

**Changes to Window Adjustment Cursor Names**

The names of some of the cursors used internally by the packages, such as `move_cursor`, could easily conflict with client programs. The internal names have been changed to `wmgr_*` to avoid conflict.

**Example Program `filer`**

The (optional) example program

`/usr/src/sun/suntool/examples/filer.c` would not compile in Release 3.2 because the routine `confirm_yes` was undefined. `filer.c` was meant to be accompanied by a simple confirmer package, `confirm.c` that includes this routine. The confirmer code was printed at the end of `filer.c` in Appendix B of the 3.2 *SunView Programmer's Guide*.

The file `/usr/src/sun/suntool/examples/confirm.c` is included in the optional software for 3.4, along with a simple Makefile for the examples.

**Other Fixes to Example Programs**

The `seln_demo.c` and `tty_io.c` example programs (part of the optional software) from 3.2 did not work properly. They both called `window_fit()` to resize their frames, but did not explicitly give a size for their subwindows. This caused them to shrink to a very small size, as explained in the addenda *Fitting Frames Around Subwindows* in Section 12.1, “Errata and Addenda for the *SunView Programmer's Guide*.” Revised versions of the programs are included in

## 3.4.

The program `typein.c` has been revised to use the new `tty` subwindow attributes described in Section 6.2.6, *New Tty Subwindow Attributes*, above. The program is printed out in full in *Revised typein.c Example Program*, in Section 12.1, “Errata and Addenda for the SunView Programmer’s Guide.”

## 7.4. SunOS Bug Fixes

This section contains bug fixes that were made to the kernel, utilities, and shells.

## Installation and Configuration-related Bugs

The following bugs, related to booting, configuring the kernel, and installing the operating system, were fixed in this release.

## Missing Sun-3/110 Configuration Files

The Sun-3/110 configuration files `/usr/sys/conf/SDST110` and `/usr/sys/conf/ND110` were not included in Release 3.2. These files are included in Release 3.4. (Refer to Chapter 3, *Reconfiguring Your Kernel*, for more information.)

`/sys/sundev/mti_conf.c` File Missing

The file `/sys/sundev/mti_conf.c` was not included in Release 3.2. It is included in Release 3.4.

## Remote Install Problems

During remote installations of Release 3.3, the upgrade script redirected the `dd` command’s `stderr` to standard output, which was then piped as standard input to `tar`. Thus, `dd`’s status messages were getting mixed up with the file transfer.

## Sun-3/50-4 Tape Cartridge Hangs

In Release 3.3, Sun 3/50-4s with shoe boxes containing Micropolis 1324 disks and Archive cartridge tape drives (serial 9050 B) experienced tape drive hangs. This occurred after upgrade and reconfiguration when users tried to access the tape drive with `tar`, and `rewind`.

This has been fixed in Release 3.4.

`bzero` Routine Could Not Handle Large `MAXUSERS`

The kernel’s `bzero` routine did not handle more than 256 Kbytes correctly. As a result, systems with Xylogics disks could get the following kernel panic message:

```
panic:memall intrans|want
```

when `MAXUSERS` was configured above a certain amount. For example, a Sun-3/160 would fail when `MAXUSERS` was set to 12 or above. Sun-3/260s and Sun-3/180s would fail when `MAXUSERS` was set to 24 or above.

This condition was happening because setting `MAXUSERS` too large caused the kernel to call its own `bzero` routine with greater than 256 Kbytes. Previously, the kernel’s `bzero` could not handle this size correctly.

In Release 3.4, the kernel’s `bzero` routine works correctly with sizes greater than 256 Kbytes. You can now set `MAXUSERS` to any number required without problems resulting from `bzero`’s incorrect handling of the larger request.

- Sun-3 Kernel Could Not Be Built Without Sun-3/260 Defined**
- In Release 3.2, you could not build a Sun-3 kernel from source code unless `SUN3_260` was defined in the configuration file. In this case, you would receive a message that `vac_flushall()` was undefined.
- This problem has been fixed in Release 3.4.
- Sun-3s with Early PROMs Had Rebooting Problems**
- If you added large amounts of memory to a Sun-3 with a pre-1.6 PROMs, it occasionally failed to reboot. As a result, you had to power cycle the system after it hung due to this problem.
- This problem has been fixed in Release 3.4.
- 3.3UPGRADE Did Not Work on Systems with 3.2EXPORT**
- Systems running 3.2EXPORT could not upgrade to Release 3.3. Any attempts to do so resulted in an error message and termination of the upgrade. This was because the upgrade script inadvertently did not accept 3.2EXPORT as a legitimate release.
- The installation script has been fixed. You can upgrade 3.2EXPORT to Release 3.4.
- Standalone Programs Did Not Exit Correctly**
- The Release 3.2 boot program caused standalone programs, such as `diag`, to behave incorrectly when they exited. For example, if you entered
- ```
diag> quit
```
- this restarted `diag`, rather than quit the program, as expected.
- This bug has been fixed in 3.4. Now standalone programs correctly return to the PROM monitor when exited.
- Kernel-related Bugs**
- The following additional kernel-related bugs were fixed in Release 3.4
- Cursor Motion Not Coordinated with Mouse Motion**
- In earlier releases, the cursor sometimes did not properly track mouse motion. For example, if you slowly moved the mouse in a horizontal direction, the cursor would move diagonally, from the lower left to the upper right of the screen. If you slowly moved the cursor in a vertical direction, the cursor once again would move diagonally.
- This behavior would occasionally occur once the window system was initialized. Additionally, this resulted in the mouse buttons not being recognized unless you moved the mouse while pressing a button. Previously, you had to unplug and replug the mouse, or, if necessary, reboot the system to clear the problem. In Release 3.4, the cursor tracks the mouse correctly at all times.
- Memory Device Driver Bug**
- In Release 3.2, a bug was introduced into the memory device driver. Though invisible to the users, this bug was fixed in Release 3.4 to improve memory device driver efficiency.

Bug in `kadb`

If you used a standalone sun-3 server, you had to manually invoke the kernel debugger `kadb`, because the PROM monitor did not run it automatically. Instructions for doing this are in the `kadb(8S)` manual pages in Appendix C.

Error Reporting Problems with SCSI Tape Drives

Sun 3/50s with MT02 SCSI tape drives were experiencing problems when you tried to append a file to a 1/4-inch tape. The application doing the write would seem to succeed, but actually, the tape would not move. Thus, the write would not happen. Furthermore, you would not receive a message to that affect. Instead, you occasionally received the following internal error message on the console:

```
st0: stintr: sense f0 0 5 0 0 0 6 3 33 0 0
```

This problem has been fixed in Release 3.4. The append now succeeds, and no error message is issued.

VPC-2200 Parallel Printer/Plotter Interface Problems

In prior releases, systems with the VPC-2200 parallel printer/plotter interface were experiencing compatibility problems with other drivers. The VPC-2200 was allocating an odd number of bytes of direct virtual memory access space. This caused problems for other drivers that expect their byte allocations on even boundaries. The SCSI driver was modified in Release 3.3 to handle this situation.

In Release 3.4, the VPC-2200 driver was modified so that it would allocate an even number of bytes. This ensures problem-free operation for drivers requiring even byte alignments that allocate memory after the VPC.

Daisy Chained Shoe Boxes on Sun-3/50 Sometimes Hung

In Release 3.3, Sun-3/50s with two daisy chained shoeboxes would not boot if the configuration had a 141 Mbyte drive (ESDI) designated as Unit 0 and a 71 Mbyte drive (ST506) as Unit 1. This problem has been fixed in Release 3.4.

`setrlimit` Sometimes Did Not Return Error Codes

In previous releases, the `setrlimit` system call sometimes had problems handling invalid arguments to it. In this release, `setrlimit` will return `EINVAL` errors when newly supplied resource limits conflict with each other.

Occasional Spurious User Bus Errors Around Page Boundaries

Previously, certain programs running on Sun-3s would occasionally dump core from a SIGSIGV signal. This was due to the error handling of some prefetch conditions. The problem was mostly corrected in the 3.2 release, and fully corrected in Release 3.4.

New Version of the `ie` Ethernet Driver

Release 3.4 contains a new version of the `ie` (Intel 82586) Ethernet Driver. This driver is faster and more reliable than the previous version of the driver.

Incompatible Routine in System `V curses` Library

In previous releases, the System `V curses` library contained `select.o`, a `select` system call emulation. This routine was removed from `curses` in Release 3.4 because it was incompatible with the existing `select` system call. The manual page `select(3V)` will not be deleted, however.

Daylight Savings Times Were Incorrectly Calculated	The <code>ctime</code> and <code>localtime</code> routines incorrectly calculated Daylight Savings Times for the United States, Canada, and Australia. This has been fixed in Release 3.4.
MAKEDEV Did Not Create a vpc Node	MAKEDEV <code>vpc</code> did not make a special file for a <code>vpc</code> device (that is, <code>/dev/vpc0</code>). Instead, it made a node for a <code>vp</code> device (<code>/dev/vp0</code>). This behavior was as documented in the <code>vpc(4S)</code> manual page. Both the behavior and the manual page have been corrected in Release 3.4. MAKEDEV <code>vpc</code> now makes a node for a <code>vpc</code> device.
Kernel Did Not Recognize Pseudo-terminals	Pseudo-terminals <code>/dev/ptyr0</code> (33) through <code>/dev/ptyrf</code> (48) were not recognized by the kernel. The actual number of pseudo-terminals has increased to 48, as documented.
Hard Links to Directories Caused System Panic	Making a hard link to a directory with <code>ln -f</code> caused a panic. Normally, the superuser can force a hard link to a directory. However, a bug in Sun Release 3.0 resulted in a system panic when this was attempted. This has been fixed in 3.4. Using <code>ln -f</code> to force a hard link to a directory no longer causes a panic.
readlink Bug	In Sun Release 3.2, a <code>readlink()</code> call against a special device returned a garbage value. This has been corrected in Release 3.4.
Problems with <code>kmem</code>	Earlier releases contained a bug that caused the kernel to crash with a bus error when <code>/dev/kmem</code> was accessed using high virtual addresses of the form <code>0xffffffffxx</code> . This has been fixed.
indent-related Bug Fixes	The following fixes were made to <code>indent</code> .
4.3 BSD Bug Fixes	Bug fixes from 4.3 BSD were incorporated into the <code>indent</code> program.
indent Problems with typedefs	The <code>indent</code> program now correctly generates and formats <code>typedefs</code> specified in the <code>.indent.pro</code> file. Formerly, only the first 14 <code>typedefs</code> were generated and formatted correctly.
Comment Blocks Were Incorrectly Formatted	The <code>indent</code> program now correctly formats groups of block comments such as the one below:

```

/*****
 * first comment
 *****/

        /*****
         * second comment
         *****/

```

Missing Semicolons Confused indent

In Release 3.4, C code with missing semicolons no longer confuses indent. For example, indent used to complain about this code:

```
struct x {
    struct x      *next;
    NAMEBLOCK    name
};
```

A semicolon is missing after name, but this is still legal C code.

indent Misformats Certain Combinations

The indent program used to misformat the first line of a C function when the function had an explicit return value specified, the function had no local variables declared, and the option `-nps1` was selected on the command line, or specified in `.indent.pro`. This following code:

```
long _cgi_shared_screen()
{
    if (errno == EBUSY)
        _cgi_shared_screen_state = EBUSY;
    return (0);
}
```

used to give the following results:

```
long      _cgi_shared_screen()
{
    if      (errno == EBUSY)
        _cgi_shared_screen_state = EBUSY;
    return (0);
```

This problem has been fixed in Release 3.4.

Formatting Problems with Negative Floating-point Numbers

Running indent on a C program containing a floating-point number with a negative exponent resulted in incorrect results. In this expression, indent used to put spaces around the minus sign, changing the value of the result as follows:

```
x = 100e - 5;
```

This problem has been fixed in Release 3.4.

indent Put Spaces Around "\$"

Even though the dollar sign is a valid character in an identifier, indent used to put spaces around it. For example:

```
abc$def      became      abc $ def
```

The dollar sign is treated correctly now.

Utilities-related Bug Fixes

The following utilities-related bugs were fixed in Release 3.4.

Processes Failed to Reset Terminal Modes

The system console was rendered inaccessible by processes that failed to reset terminal modes properly. Processes that set FIONBIO on their standard input, then exit without resetting the terminal mode, cause programs that do not handle EWOLDBLOCK errors to exit also. Programs that exit as a result of this error include `cs` and `login`, among others. A problem arose when a such process, having the console as its standard input, exited. This caused the shell to die and left FIONBIO set on the console. The new `login` process initiated on the console then received the EWOLDBLOCK error, causing it to die, and rendering the console inaccessible. The console would remain unavailable until the system was rebooted (or until a special program was executed over a remote login) to clear the console's terminal mode.

This problem has been averted. `login` now resets the terminal mode for the console, if appropriate.

Using `sccs` with a `setuid` Front End

When you used `sccs` with an SCCS subdirectory that has write access restricted to the owner, it disallowed creation of the lock (`p.file`) for editing for all but the owner. This is true even when the edit function is called from within a `setuid` front end. This has been corrected in 3.4beta. `sccs` now allows files to be checked out when called from within a `setuid` front end.

Bug in `dkinfo`

The program `/etc/dkinfo` failed to close file descriptors. It ran out on long lists of disks and partitions and exited prematurely.

This has been corrected in Release 3.4. `/etc/dkinfo` now closes file descriptors associated with each partition it opens.

Reply Sometimes Garbled Addresses

Certain mail-message header formats caused return addresses to be garbled by the `Reply` command in `Mail` (`/usr/ucb/mail`). This has been corrected.

`vgrind` Omitted “?” Identifier

The `vgrind` program was changed in Release 3.4 so that it considers `?` to be a valid identifier for `emacs` MLisp code that it formats.

Numeric Fields Failed in `tbl`

The `tbl` program was fixed in Release 3.3 to treat numeric data fields correctly, and the same fix applies to Release 3.4. As a side effect of a bug fix involving data overflow, the `tbl` in Release 3.2 formatted numeric data incorrectly when there were many columns of data.

`.TH` Macro Bug

The `.TH` macro in the `man` macro package did not process arguments consistently with other `man` macro implementations, or as documented in *Formatting Documents on the Sun Workstation*.

This has been fixed in this release. Argument four, when supplied, now sets a value for the page foot left (version); argument five now sets a value for the page head center (main).

Bug in maze Demonstration The maze demonstration halted and dumped core when run outside SunView on a high-resolution, black-and-white monitor, due to fixed-sized arrays that were too small for the particular screen. This has been corrected.

Bug in chesstool When `chesstool` was invoked with a large font size, not all buttons in the panel were visible. This has been corrected.

Complex Makefiles Caused Hash Table Overflow In Releases 3.0 and 3.2, very large and complex makefiles caused `make`'s internal hash table to overflow. In Release 3.4, the size of this table has been redefined to a suitably huge value.

indxbib Dropped Core In Releases 3.0 and 3.2, `indxbib` dropped core due to a memory fault when it processed files that had been run through `addbib`. This has been corrected.

Shell-related Bug Fixes The following bug fixes were made relative to the C and Bourne shells.

Bug in login Command An interaction between `cs``h`, `login`, and `sh` resulted in a situation whereby a user running `login` from within `cs``h` (typically by using the `cs``h` built-in `login` command), to log in as a Bourne shell user, found that `CTRL-C` interrupts were disabled for the resulting Bourne shell. Apparently, `login` did not completely account for and eliminate the effects of `cs``h`'s job control.

This has been corrected. Interrupts now work properly for Bourne shell sessions started by logging in from `cs``h`.

time Memory Statistics Printed in Pages The documentation for the `cs``h` built-in `time` command stated that timed-process memory size statistics were to be printed in Kbytes. However, in Releases 3.0 and 3.2, these figures were printed in machine pages. This led to discrepancies in reports for identical programs running on different machines, and to unrealistic-appearing statistics.

This has been corrected in this release. Memory size statistics are now printed in Kbytes, as stated.

7.5. Networking The following network-related bugs were fixed in Release 3.4.

Problems Using ifconfig for Diskless Clients If `ifconfig` was used to change the Ethernet address of a diskless client, that workstation would not reboot unless the `arp` table of its file server was first updated by hand to reflect the new address.

In Release 3.4, this has been corrected. `rarpd` has been modified to supply the correct client's Ethernet address to the server's `arp` table when the client reboots.

- Bugs in TCP/IP** The TCP and ICMP protocol support in the kernel was updated in Release 3.4 to include many bug fixes between 4.2BSD and 4.3BSD. This fixes many problems with congestion on slow-speed networks, such as initial retransmission timeouts that were too short and the handling of ICMP redirect messages.
- Net Booting on Non-zero Subnets** In Release 3.3, booting workstations over the Ethernet could fail in some cases on non-zero IP subnets, giving the message
- ```
nd: output error 51
```
- Network booting on subnets should be more reliable in Release 3.4.
- Telnet Server End-of-line Conventions** The Internet Telnet server (`/usr/etc/in.telnetd`) was changed in Release 3.2 to be compatible with 4.3BSD, which interprets the CRLF sequence as LF. Since this was incompatible with Release 3.0 and several other Telnet implementations, in 3.4 CRLF is again interpreted as CR. Release 3.4 should again allow other operating systems and terminal servers to send CRs.
- Frozen Configuration Files Ignored Domain Information** Release 3.2 added the ability to use the Yellow Pages domain as the default domain for `/usr/lib/sendmail`. If the `-bz` option was used on `sendmail` to create a “frozen” configuration file, the Yellow Pages domain could not be overridden. In Release 3.4, `/usr/lib/sendmail` will override the domain name with an explicit name in the “DD” line.
- Lance Ethernet Driver Bug** Earlier releases contained versions of the Lance Ethernet driver with a bug that, while mostly harmless, did cause `etherfind` to report that minimum packet sizes were four bytes larger than they actually were. This has been fixed.
- ping Sent Bad Packets** Earlier releases contained versions of `ping` that sometimes sent bad packets, which were then ignored by the remote host. Only even-length packets were treated correctly. This has been fixed.
- rpcinfo Did Not Work on Non-networked Standalones** Earlier releases contained versions of `rpcinfo`, which did not work on standalone, non-networked machines. This has been fixed in Release 3.4.
- NFS Clients Sometimes Wrote Garbage** Earlier releases contained versions of NFS in which clients could write garbage in certain very heavily loaded states. The problem was that, under certain circumstances, the NFS client’s output buffer was reused even though it was still busy. These circumstances were as follows:
- The client received a reply for a transmission of a certain request while waiting for a reply for a subsequent transmission of the same request.
  - Another process acquired the client handle, which contains the busy output buffer, then used XDR to put data into that buffer prior to the freeing of its last segment.

The NFS bug that caused these problems has been fixed.

### Software Loopback lo0 Address Could Not Be Set

Earlier releases contained kernel network code that could not correctly redefine the software loopback lo0. After an attempt to do so, routing to lo0 yielded the message

```
Network is unreachable
```

This has been fixed in Release 3.4. However, you may experience other problems with changing the loopback address, since Yellow Pages software always uses the standard loopback address.

### FBIIOGVVIDEO Was Sometimes Unreliable

Earlier releases contained versions of the system software that sometimes returned incorrect values on the FBIIOGVVIDEO ioctl. This has been fixed in 3.4.

## 7.6. System Administration

The following system-administration-related bugs were fixed in Release 3.4.

### Diskless Clients Can Now Reboot After Changing Ethernet Address with ifconfig

When changing the Ethernet address of a diskless client with the ifconfig command, you used to have to modify the arp table of its server by hand to get it to reboot properly. This bug has been fixed in Release 3.4, so that you no longer have to modify the arp table.

/etc/dkinfo

/etc/dkinfo no longer runs out on long sequences of disks that have many partitions, and no longer generates a

```
too many open files
```

error message.

### NFS Server Port Checking

Running an adb -w command on /usr/etc/rpc.mountd no longer generates a no symbol found error message.

### Wrong termcap Entry for TVI-925

In Release 3.2, the termcap entry for the TVI-925 was modified so that it would support the Wyse-50 in TVI-925 emulation mode. In fact, the Wyse-50 does not emulate the TVI-925 correctly. The TVI-925 has the newline glitch :xn:, meaning it ignores a newline after autowrap. The Wyse-50 in TVI-925 emulation mode does not have this characteristic. In Release 3.3 (and 3.4) the TVI-925 termcap entry is correct, and there is a new entry called wyse-925 to describe the Wyse-50 emulating a TVI-925.

---

## Errata and Addenda for 3.x Manuals

The following chapters describe errata and addenda for working with SunView in Release 3.5. These chapters also include the errata and addendum for 3.2 manuals and 3.4 manuals.

The errata and addendum chapters for previous releases are:

- Various language manuals
- *Pixrect Reference Manual*
- *SunCore Reference Manual*
- *SunCGI Reference Manual*
- *SunView Programmer's Guide*
- *SunView System Programmer's Guide*
- *System Administration Manual for the Sun Workstation*

The above errata and addenda chapters were printed in the Release 3.4 Manual. In addition, there is a new addendum chapter for the *Windows and Window-Based Tools: Beginner's Guide* that is for Release 3.5. This chapter is called

- *Working with SunView.*



# Language Manuals

|                                         |     |
|-----------------------------------------|-----|
| Language Manuals .....                  | 151 |
| 8.1. Language-related Addenda .....     | 151 |
| dbx setenv Documentation Bug .....      | 151 |
| as Addressing Modes Documentation ..... | 151 |
| Language-related Errata .....           | 152 |
| as Documentation Bug .....              | 152 |
| Debugging Documentation Bug .....       | 152 |
| lint Line Count Bug .....               | 152 |
| adb Signal Bug .....                    | 152 |



---

## Language Manuals

The following languages-related errata and addenda will be included in this section:

### 8.1. Language-related Addenda

#### dbx setenv Documentation Bug

Below is an addendum to languages documentation.

Insert the following paragraph in *Debugging Tools for the Sun Workstation*, immediately before Section 3.12, Page 30:

```
setenv
```

Set the environment variable *name* to the value of *string*. (See `cs(1)`).

#### as Addressing Modes Documentation

Insert the following section in the *Assembly Language Reference Manual* at the bottom of Page 42 after the paragraph that starts "Note that the 3.0 release...":

The memory indirect and program counter memory indirect addressing modes listed in the following tables are usable only when assembling for the MC68020. In each of these addressing modes, four user-specified values are used to generate the final operand address:

- base register
- base displacement
- index register
- outer displacement

All four user-specified values are optional. Both base and outer displacements may be null, word, or long. When a displacement is null, or an element is suppressed, its value is taken as zero in the effective address calculation.

In the case of memory indirect addressing, an address register (*an*) is used as a base register, and its value can be adjusted by an optional base displacement (*d'*). An index register (*ri*) specifies an index operand (*ri:L:s*) and finally, an outer displacement (*d*) can be added to the address operand, yielding the effective address.

Program counter memory indirect mode is exactly the same except that the program counter is used as the base register.

Some examples of these addressing modes follow:

```

an@ (d' :L, ri:L:s)@(d:L)
an@(d:L)@(d' :L,ri:L:s)
an@@
an@(d:L)@
an@(d' :L,ri:L:s)@
pc@@
pc@(d:L)@
pc@(d' :L,ri:L:s)@(d:L)
pc@(d:L)@(d' :L,ri:L:s)
@(d:L)@
@(d' :L,ri:L:s)@(d:L)
@(d:L)@(d' :L,ri:L:s)
@(d' :L,ri:L:s)@

```

## Language-related Errata

The following language-related errata have been reported in this release.

### as Documentation Bug

In the *Assembly Language Reference Manual*, Appendix B, Table B-1, Page 72, the group beginning with `fmove`, remove the last two lines in Column 3. They read

```

fmove fm, ea{dn}
fmove fm, ea{#k}

```

### Debugging Documentation Bug

Remove the description of the `attach` command described on Page 30 of the *Debugging Tools Manual*. The command does not exist in `dbx`.

### lint Line Count Bug

If a source program contains lines that are continued over more than one line, `lint` counts the continued lines as one single line. The line numbers reported by `lint` from that point no longer correspond to line numbers according to a text editor.

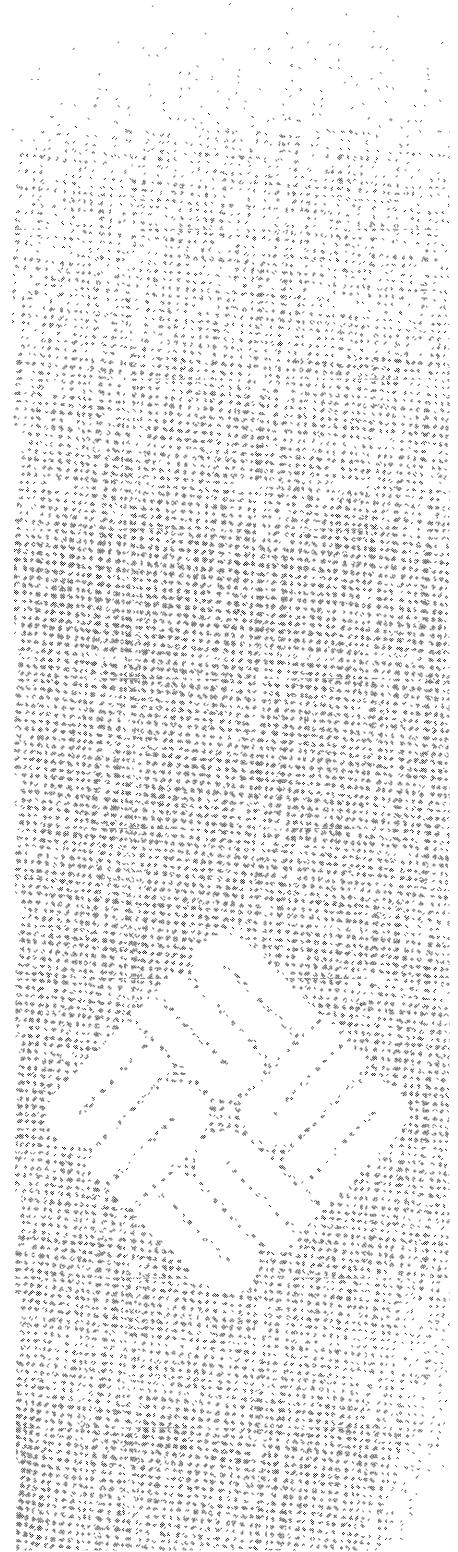
### adb Signal Bug

Sometimes, if you try to run `adb` on a core file where the stack includes a frame created by `_sigtramp`, `adb` will dump core. The problem does not occur with `dbx`.



## File I/O Facilities for Pixrects

|                                              |     |
|----------------------------------------------|-----|
| File I/O Facilities for Pixrects .....       | 155 |
| 9.1. Writing and Reading Raster Files .....  | 155 |
| Write Raster File .....                      | 155 |
| Read Raster File .....                       | 157 |
| 9.2. Details of the Raster File Format ..... | 158 |
| 9.3. Writing Parts of a Raster File .....    | 159 |
| Write Header to Raster File .....            | 159 |
| Initialize Raster File Header .....          | 160 |
| Write Image Data to Raster File .....        | 160 |
| 9.4. Reading Parts of a Raster File .....    | 160 |
| Read Header from Raster File .....           | 160 |
| Read Colormap from Raster File .....         | 161 |
| Read Image from Raster File .....            | 161 |
| Read Standard Raster File .....              | 161 |





---

## File I/O Facilities for Pixrects

Sun Microsystems, Inc. has specified a file format for files containing raster images. The format is defined in the header file `<rasterfile.h>`. The `pixrect` library contains routines to perform I/O operations between `pixrects` and files in this raster file format. This I/O is done using the routines of the C Library Standard I/O package, requiring the caller to include the header file `<stdio.h>`.

The raster file format allows multiple types of raster images. Unencoded and run-length-encoded formats are supported directly by the `pixrect` library. Support for customer-defined formats is implemented by passing raster files with non-standard types through filter programs. Sun supplied filters are found in the directory `/usr/lib/rasfilters`. This directory also includes sample source code for a filter that corresponds to one of the standard raster file types to facilitate writing new filters.

### 9.1. Writing and Reading Raster Files

#### Write Raster File

The sections that follow describe how to store and retrieve an image in a `rasterfile`.

```
int pr_dump(input_pr, output, colormap, type, copy_flag)
Pixrect *input_pr;
FILE *output;
colormap_t *colormap;
int type, copy_flag;
```

The `pr_dump` procedure stores the image described by a `pixrect` onto a file. It normally returns 0, but if any error occurs it returns `PIX_ERR`. The caller can write a rectangular subregion of a `pixrect` by first creating an appropriate `input_pr` via a call to `pr_region`. The output file is specified via `output`. The desired output type should either be one of the following standard types or correspond to a customer provided filter.

```
#define RT_OLD 0
#define RT_STANDARD 1
#define RT_BYTE_ENCODED 2
```

The `RT_STANDARD` type is the common raster file format in the same sense that memory `pixrects` are the common `pixrect` format: every raster file filter is required to read and write this format. The `RT_OLD` type is very close to the `RT_STANDARD` type; it was the former standard generated by old versions of

Sun software. The `RT_BYTE_ENCODED` type implements a run-length encoding of bytes of the `pixrect` image. This usually results in shorter files, although pathological images may expand by 50 percent. Specifying any other output type causes `pr_dump` to pipe a raster file of `RT_STANDARD` type to the filter named `convert.type`, looking first in directories in the user's `$PATH` environment variable, and then in the directory `/usr/lib/rasfilters`. `type` is the ASCII corresponding to the specified `type` in decimal. The output of the filter is then copied to `output`.

It is strongly recommended that customer-defined formats use a type of 100 or more, to avoid conflicts with additions to the set of standard types. The `RT_EXPERIMENTAL` type is reserved for use in the development of experimental filters, although it is no longer treated specially.

```
#define RT_EXPERIMENTAL 65535
```

`pr_dump` and other functions that start filters wait until the filter process exits before returning, so caution is advisable when working with experimental filters.

For `pixrects` displayed on devices with `colormaps`, the values of the pixels are not sufficient to recreate the displayed image. Thus, the image's `colormap` can also be specified in the call to `pr_dump`. If the `colormap` is specified as `NULL` but `input_pr` is a non-monochrome display `pixrect`, `pr_dump` will attempt to write the `colormap` obtained from `input_pr` (via `pr_getcolormap`). The following structure is used to specify the `colormap` associated with `input_pr`:

```
typedef struct {
 int type;
 int length;
 unsigned char *map[3];
} colormap_t;
```

The `colormap` type should be one of the Sun supported types:

```
#define RMT_NONE 0
#define RMT_EQUAL_RGB 1
#define RMT_RAW 2
```

If the `colormap` type is `RMT_NONE`, then the `colormap` length must be 0. This case usually arises when dealing with monochrome displays and 1-bit deep memory `pixrects`. If the `colormap` type is `RMT_EQUAL_RGB`, then the map array should specify the red (`map[0]`), green (`map[1]`) and blue (`map[2]`) `colormap` values, with each vector in the map array being of the same specified `colormap` length. If the `colormap` type is `RMT_RAW`, the first map array (`map[0]`), should hold `length` bytes of `colormap` data, which will not be interpreted by the `pixrect` library.

Finally, `copy_flag` specifies whether or not `input_pr` should be copied to a temporary `pixrect` before the image is output. The `copy_flag` value should be non-zero if `input_pr` is a `pixrect` in a frame buffer that is likely to be asynchronously modified. Note that use of `copy_flag` still will not guarantee that the correct image will be output unless the `pr_rop` to copy from the frame buffer is made uninterruptible.

```

#include <stdio.h>
#include <sys/types.h>
#include <pixrect/pixrect.h>
#include <pixrect/pr_io.h>

main()
{
 Pixrect *screen, *icon;
 FILE *output = stdout;
 colormap_t *colormap = 0;
 int type = RT_STANDARD;
 int copy_flag = 1;

 if (!(screen = pr_open("/dev/fb")) ||
 !(icon = pr_region(screen, 1050, 10, 64, 64)))
 exit(1);

 pr_dump(icon, output, colormap, type, copy_flag);
 pr_close(screen);

 exit(0);
}

```

Figure 9-1 *Example Program with pr\_dump***Read Raster File**

```

Pixrect *pr_load(input, colormap)
FILE *input;
colormap_t *colormap;

```

The `pr_load` function can be used to retrieve the image stored in a raster file into a `pixrect`. The raster file's header is read from `input`, a `pixrect` of the appropriate size is dynamically allocated, the colormap is read and placed in the location addressed by `colormap`, and finally the image is read into the `pixrect` and the `pixrect` returned. If any problems occurs, `pr_load` returns `NULL`.

As with `pr_dump`, if the specified raster file is not of standard type, `pr_load` first runs the file through the appropriate filter to convert it to `RT_STANDARD` type and then loads the output of the filter.

Additionally, if `colormap` is `NULL`, `pr_load` will simply discard any and all colormap information contained in the specified input raster file. If `colormap` is non-null, `pr_load` will load the colormap data even if the type and length specified do not match that of the file (see `pr_load_colormap` below).

```

#include <stdio.h>
#include <sys/types.h>
#include <pixrect/pixrect.h>
#include <pixrect/pr_io.h>

main()
{
 struct pixrect *screen, *icon;
 FILE *input = stdin;
 colormap_t colormap;

 colormap.type = RMT_NONE;

 if (!(screen = pr_open("/dev/fb")) ||
 !(icon = pr_load(input, &colormap)))
 exit(1);

 if (colormap.type == RMT_EQUAL_RGB)
 pr_putcolormap(screen, 0, colormap.length,
 colormap.map[0], colormap.map[1],
 colormap.map[2]);

 pr_rop(screen, 1050, 110, 64, 64, PIX_SET, icon, 0, 0);
 pr_close(screen);

 exit(0);
}

```

Figure 9-2 *Example Program with pr\_load*

## 9.2. Details of the Raster File Format

A handful of additional routines are available in the pixrect library for manipulating pieces of raster files. In order to understand what they do, you need to understand the exact layout of the raster file format.

The raster file is in three parts: first, a small header containing eight 32-bit int's; second, a (possibly empty) set of colormap values; third, the pixel image, stored a line at a time, in increasing y order.

The image is essentially laid out in the file the exact way that it would appear in a static memory pixrect. In particular, each line of the image is rounded out to a multiple of 16 bits, corresponding to the rounding convention used by static pixrects.

The header is defined by the following structure:

```

struct rasterfile {
 int ras_magic;
 int ras_width;
 int ras_height;
 int ras_depth;
 int ras_length;
 int ras_type;
 int ras_maptype;
 int ras_maplength;
};

```

The `ras_magic` field always contains the following constant:

```
#define RAS_MAGIC 0x59a66a95
```

The `ras_width`, `ras_height` and `ras_depth` fields contain the image's width and height in pixels, and its depth in bits-per-pixel, respectively. The depth is usually either 1 or 8, corresponding to the standard frame buffer depths.

The `ras_length` field contains the length in bytes of the image data. For an unencoded image, this number is computable from the `ras_width`, `ras_height`, and `ras_depth` fields, but for an encoded image it must be explicitly stored in order to be available without decoding the image itself. Note that the length of the header and of the possibly empty colormap values are not included in the value in the `ras_length` field; it is only the image data length. For historical reasons, files of type `RT_OLD` will usually have a 0 in the `ras_length` field, and software expecting to encounter such files should be prepared to compute the actual image data length if it is needed. The `ras_maptype` and `ras_maplength` fields contain the type and length in bytes of the colormap values, respectively.

If the `ras_maptype` is not `RMT_NONE` and the `ras_maplength` is not 0, then the colormap values are the `ras_maplength` bytes immediately after the header. These values are either uninterpreted bytes (usually with the `ras_maptype` set to `RMT_RAW`) or the equal length red, green, and blue vectors, in that order (when the `ras_maptype` is `RMT_EQUAL_RGB`). In the latter case, the `ras_maplength` must be three times the size in bytes of any one of the vectors.

### 9.3. Writing Parts of a Raster File

The following routines are available for writing the various parts of a raster file. Many of these routines are used to implement `pr_dump`. First, the raster file header and the colormap can be written by calling `pr_dump_header`.

#### Write Header to Raster File

```

int pr_dump_header(output, rh, colormap)
FILE *output;
struct rasterfile *rh;
colormap_t *colormap;

```

`pr_dump_header` returns `PIX_ERR` if there is a problem writing the header or the colormap, otherwise it returns 0. If the colormap is `NULL`, no colormap values are written.

**Initialize Raster File Header**

```
Pixrect *pr_dump_init(input_pr, rh, colormap,
 type, copy_flag)
Pixrect *input_pr;
struct rasterfile *rh;
colormap_t *colormap;
int type, copy_flag;
```

For clients that do not want to explicitly initialize the rasterfile struct the following routine can be used to set up the arguments for `pr_dump_header`. The arguments to `pr_dump_init` correspond to the arguments to `pr_dump`. However, `pr_dump_init` returns the pixrect to write, rather than actually writing it, and initializes the structure pointed to by `rh` rather than writing it. If `colormap` is NULL, the `ras_maptype` and `ras_maplength` fields of `rh` will be set to `RMT_NONE` and 0, respectively.

If any error is detected by `pr_dump_init`, the returned pixrect is NULL. If there is no error, the `copy_flag` is zero, and the input pixrect is suitable for direct dumping (a primary memory pixrect), the returned pixrect is simply `input_pr`. However, if `copy_flag` is non-zero, or the input pixrect cannot be dumped directly, the returned pixrect is dynamically allocated and the caller is responsible for deallocating it with `pr_destroy` when it is no longer needed.

**Write Image Data to Raster File**

```
int pr_dump_image(pr, output, rh)
Pixrect *pr;
FILE *output;
struct rasterfile *rh;
```

The actual image data can be output via a call to `pr_dump_image`. This routine returns 0 unless there is an error, in which case it is `PIX_ERR`. It cannot write the image in a non-standard (filtered) format, since by the time it is called the raster file header has already been written.

Since these routines sequentially advance the output file's write pointer, `pr_dump_image` must be called after `pr_dump_header`.

**9.4. Reading Parts of a Raster File**

The following routines are available for reading the various parts of a raster file. Many of these routines are used to implement `pr_load`. Since these routines sequentially advance the input file's read pointer, rather than doing random seeks in the input file, they should be called in the order presented below.

**Read Header from Raster File**

```
int pr_load_header(input, rh)
FILE *input;
struct rasterfile *rh;
```

The raster file header can be read by calling `pr_load_header`. This routine reads the header from the specified input, checks it for validity, and initializes the specified `rasterfile` structure from the header. The return value is 0 unless there is an error, in which case it is `PIX_ERR`.



**Read Colormap from Raster File**

```
int pr_load_colormap(input, rh, colormap)
FILE *input;
struct rasterfile *rh;
colormap_t *colormap;
```

If the header indicates that there is a non-empty set of colormap values, they can be read by calling `pr_load_colormap`. If the specified colormap is `NULL`, this routine will skip over the colormap values by reading and discarding them. If the type and length values in `colormap` do match the input file, `pr_load_colormap` will allocate space for the colormap with `malloc`, read the colormap, and modify `colormap` before returning. If this occurs, the space allocated can be released with a `free(colormap->map[0])`.

The return value is 0 unless there is an error, in which case it is `PIX_ERR`.

**Read Image from Raster File**

```
Pixrect *pr_load_image(input, rh, colormap)
FILE *input;
struct rasterfile *rh;
colormap_t *colormap;
```

An image can be read by calling `pr_load_image`. If the input is a standard raster file type, this routine reads in the image directly. Otherwise, it writes the header, colormap, and image into the appropriate filter and then reads the output of the filter. In this case, both the rasterfile and the colormap structures will be modified as a side-effect of calling this routine. In either case, a `pixrect` is dynamically allocated to contain the image, the image is read into the `pixrect`, and the `pixrect` is returned as the result of calling the routine. If there is an error, the return value is `NULL`.

**Read Standard Raster File**

```
Pixrect *pr_load_std_image(input, rh, colormap)
FILE *input;
struct rasterfile *rh;
colormap_t colormap;
```

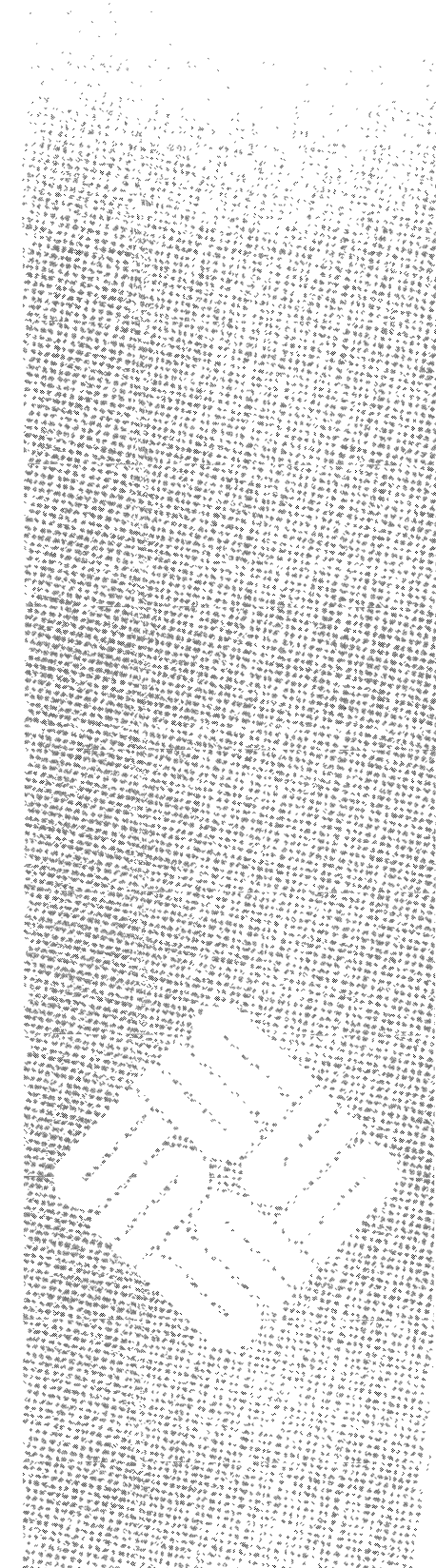
If it is known that the image is from a standard raster file type, then it can be read in by calling `pr_load_std_image`. This routine is identical to `pr_load_image`, except that it will not invoke a filter on non-standard raster file types.



---

## Windows and Window Based Tools: Beginner's Guide

|                                                                                       |     |
|---------------------------------------------------------------------------------------|-----|
| Windows and Window Based Tools: Beginner's Guide .....                                | 165 |
| 10.1. Errata and Addenda for Windows and Window Based Tools<br>Beginner's Guide ..... | 165 |
| cmdtool Improved .....                                                                | 165 |
| “Quadruple-Clicking” in Text Subwindows .....                                         | 165 |





---

## Windows and Window Based Tools: Beginner's Guide

### 10.1. Errata and Addenda for Windows and Window Based Tools Beginner's Guide

The following are errata and addenda for the Sun Release 3.0 version of the *Windows and Window Based Tools: Beginner's Guide*. These are in addition to the errata and addenda printed on Pages 114-118 of the *Release 3.4 Manual*. They do not cover all the new features of SunView (and in particular, `cmdtool`) added since the beginner's guide was published.

#### `cmdtool` Improved

The new functionality added to `cmdtool` (see Section 6.2.1, *Improvements to cmdtool*, above), has made the cautions for using `cmdtool` in Chapter 10, *The Command Facility*, unnecessary. On Page 123, delete the entire section titled “vi, more, and su: Cbreak or Raw Mode.”

#### “Quadruple-Clicking” in Text Subwindows

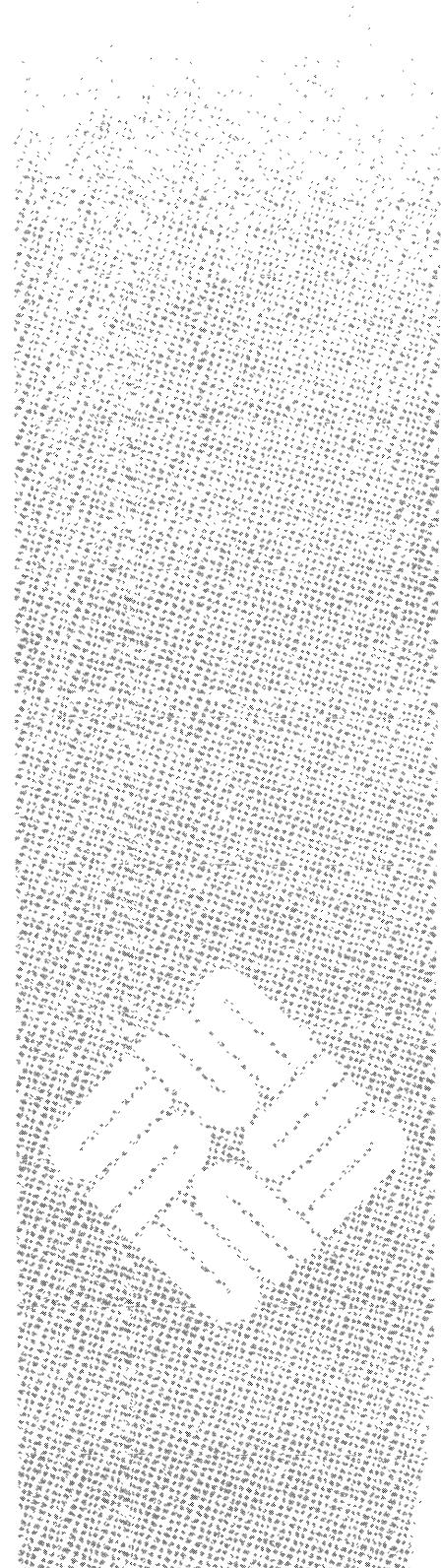
Add the following to Page 103 in Chapter 9, *Advanced Editing*, at the end of the first paragraph:

To select the entire contents of the text subwindow, point anywhere in the window and click the mouse four times in rapid succession.



## Working with SunView

|                                                                           |     |
|---------------------------------------------------------------------------|-----|
| Working with SunView .....                                                | 169 |
| 11.1. Working With SunView .....                                          | 169 |
| 11.2. General Tips .....                                                  | 169 |
| Other Sources of Information .....                                        | 169 |
| Where are the Function Keys? .....                                        | 169 |
| 11.3. Common defaultsedit Changes .....                                   | 170 |
| Walking Menus .....                                                       | 170 |
| SunView Default Font .....                                                | 170 |
| Input Preferences .....                                                   | 171 |
| Miscellaneous defaultsedit Settings .....                                 | 171 |
| 11.4. Working with Selections .....                                       | 171 |
| Pending-Delete Selections .....                                           | 171 |
| Making Selections .....                                                   | 172 |
| Useful Operations on the Selection Using <code>get_selection</code> ..... | 172 |
| Problems with <code>get_selection</code> .....                            | 172 |
| 11.5. Secondary Get .....                                                 | 173 |
| The Basic Way to Copy Text .....                                          | 173 |
| The Fast Way .....                                                        | 173 |
| Move the Cursor Back .....                                                | 173 |
| <i>Click_to_Type</i> Mode .....                                           | 173 |
| 11.6. Other Common Text Subwindow Solutions .....                         | 174 |
| How Do I Read in a File Using <code>mailtool</code> ? .....               | 174 |
| Scanning Digests .....                                                    | 174 |



|                                                                      |     |
|----------------------------------------------------------------------|-----|
| Putting > in Front of Included Text (quick search and replace) ..... | 175 |
| 11.7. Souping Up Your Root Menu .....                                | 176 |
| Add Local Applications .....                                         | 176 |
| Operations on the Selection .....                                    | 176 |
| Standard Tools .....                                                 | 177 |
| User Interface .....                                                 | 178 |
| 11.8. The Power of .textswrc .....                                   | 178 |
| Getting Started with .textswrc Filters .....                         | 178 |
| How Filters Work .....                                               | 178 |
| Pending-Delete or Not .....                                          | 179 |
| Useful .textswrc Filters .....                                       | 179 |
| Multiple Means to the Same End .....                                 | 179 |
| Keyboard “Macros” .....                                              | 179 |
| Short Macros .....                                                   | 179 |
| Long Macros .....                                                    | 179 |
| A Scrap/Test Filter .....                                            | 180 |
| Running Any Command .....                                            | 180 |
| From the Bourne Shell .....                                          | 180 |
| From the C-Shell .....                                               | 180 |
| Suggested Uses .....                                                 | 181 |
| Running the Selection through Any Command .....                      | 181 |
| Running the Selection through a “Run a Command” Filter .....         | 182 |
| Summary (an Example) .....                                           | 182 |
| 11.9. Using cmdtool .....                                            | 183 |
| Function Key Mappings .....                                          | 184 |
| Macros with mapi .....                                               | 184 |
| Escape Sequences Using mapo .....                                    | 184 |
| Sample ~/.cshrc .....                                                | 184 |
| 11.10. Performance .....                                             | 186 |
| Reduce CPU Overhead .....                                            | 186 |
| Schedule Fewer Tasks for the CPU .....                               | 186 |
| Don’t Run comsat .....                                               | 187 |
| Don’t Make Other Tools Repaint .....                                 | 187 |
| Optimize Memory Usage .....                                          | 187 |



|                                    |     |
|------------------------------------|-----|
| Use System Fonts .....             | 187 |
| Merging Applications .....         | 187 |
| Reconfigure Your Kernel .....      | 188 |
| Improve Interactive Response ..... | 188 |
| Use Retained Subwindows .....      | 188 |
| Learn Keyboard Accelerators .....  | 188 |
| Increase Mouse "Gain" .....        | 188 |
| 11.11. Miscellaneous .....         | 188 |
| Macintosh Users .....              | 188 |
| Editing Operations .....           | 189 |
| Unpacking Shell Archives .....     | 189 |



---

## Working with SunView

### 11.1. Working With SunView

This describes various aspects of the SunView environment. This chapter is an addendum to the *Beginner's Guide*. It provides tips and techniques that should suggest ways to put what you have learned together with some other SunView features and commands to make your work in SunView more productive.

The sample files listed in here are not for you to type in *as is* into your home directory; rather they are to encourage you to set up your own personal SunView work environment.

### 11.2. General Tips

#### Other Sources of Information

Besides the other chapters in this *Beginner's Guide* (and in particular, the reference cards inserted in it), there is much useful information in the `suntools(1)` and `textedit(1)` manual pages.

You can get help about editing in `textedit` by typing

```
% textedit -EH
```

Also, you can get help about the frame command line arguments that most SunView tools respond to by typing

```
% textedit -WH
```

(any tool name will do, besides `textedit`).

#### Where are the Function Keys?

This Chapter constantly refers to the SunView function keys. These are a group of function keys that have predefined meanings in standard SunView windows; in particular, many operations in text subwindows can only be performed by pressing these function keys.<sup>7</sup>

**Note:** If you are left-handed, you should try setting your machine up for left-handed function key assignment before sticking the key tops

---

<sup>7</sup> The version of SunView scheduled for the next major SunOS release has menu items for all the operations bound to function keys.

on.

Your should have a set of adhesive key tops that you can put on the function keys. Should you not have these stickers, or if you use a machine that does not have them, you can use the following mnemonics to help remember what key does what in the two columns of function keys:

Table 11-1 *SunView Function Key Mnemonics*

| <i>Outer Keys</i><br>( <b>L1</b> ) -<br>( <b>L9</b> ) |                 | <i>Inner Keys</i><br>( <b>L2</b> ) -<br>( <b>L10</b> ) |                 |
|-------------------------------------------------------|-----------------|--------------------------------------------------------|-----------------|
| <i>Key</i>                                            | <i>Mnemonic</i> | <i>Key</i>                                             | <i>Mnemonic</i> |
| Stop                                                  | Stop            | Again                                                  | All             |
| Props                                                 | Preparing       | Undo                                                   | UNIX            |
| Expose                                                | Expensive       | Put                                                    | People          |
| Open                                                  | Opulent         | Get                                                    | Get             |
| Find                                                  | Food            | Delete                                                 | Drunk           |

The **Expose** and **Open** keys work in most windows. The **Stop** key will reset the Selection Service if it gets confused; it also halts a find operation in the text subwindow, and halts other application-dependent behavior.

In addition, **F1** key acts as a **Caps Lock** key on a window-by-window basis; pressing it in most subwindows sets caps lock in that subwindow only.

### 11.3. Common defaultsedit Changes

The `defaultsedit(1)` program is a key element in personalizing your SunView environment. You should take the time to browse through its categories by clicking the left mouse button on each item's label. This will display information on each item.

**NOTE** *In general, you must quit tools and restart them to notice changes made using defaultsedit, and you have to exit suntools and restart it to see the effects of some of the options in the "SunView" category.*

### Walking Menus

One change you should immediately make is to enable *Walking Menus* in the *SunView* category. The default stacking menu style is *only* provided for backwards compatibility of the user interface with previous releases. When you change to walking menus, you can set many useful parameters in the *Menu* category, such as *Menu/Center\_string\_items*, *Menu/Font*, *Menu/Shadow*, *Menu/Stay\_up*, and *Menu/Initial\_Selection*. The *Release 3.4 Manual* describes *Menu/Center\_string\_items*, *Menu/Stay\_up*, and *Menu/Items\_in\_column\_major* in detail.

### SunView Default Font

Pick a default font that is easy to read on your display from the fonts in `/usr/lib/fonts/fixedwidthfonts`, and set it for *SunView/Font*.

**NOTE** *You can improve interactive performance by using a "built-in" SunView font; see "Use System Fonts", in Section 11.10 below.*

## Input Preferences

If you have a Sun-2 or Sun-3 machine, you should definitely turn off *Input/Jitter\_Filter* and *Input/Speed\_Enforced*; these “clamps” on mouse motion are only appropriate for earlier machines. The program `input_from_defaults` sets the system to your *Input* preferences immediately.

Miscellaneous  
defaultsedit Settings

Other preference changes usually take effect as you start up new tools. Here are some suggestions you may want to try.

- If you keep clicking in the scrollbar when you try to select characters near the beginning of a line, you can increase the left margin by changing *Text/Left\_Margin*, or you might want to try moving the scrollbar to the trailing edge of text lines by setting *Scrollbar/Vertical\_bar\_placement* to *East*. This is more useful if you “tile” your windows so they don’t overlap, meaning that their right edges are usually visible.
- To set your erase, word-erase, and line kill characters to something other than `(Delete)`, `(Control-W)`, and `(Control-U)`, set *Edit\_back\_char*, *Edit\_back\_word* and *Edit\_back\_line* in the *Text* category. To specify `(Control-A)`, for example, type `\^A`.

11.4. Working with  
Selections

When you click the left and middle mouse buttons on text, you make a selection.<sup>8</sup> Selections are very powerful:

- They work between panels, text subwindows and tty subwindows.
- You can store them on the shelf with `(Put)`
- You can delete the selection in panels and text subwindows with `(Delete)`.
- You must first make a selection to use many of the items in the text menu, such as ‘Load file’, ‘Get from file’, ‘Select line #’, and so on.
- You can use the `get_selection(1)` command to print the selection or send it to a UNIX program.

## Pending-Delete Selections

There are two kinds of selections, ordinary (appears black) and pending-delete (appears grayed out). The difference is that the next character(s) to appear *replaces* a pending-delete selection. Normally the Select and Extend (usually the left and right) mouse buttons make an ordinary selection, but if you hold down the `(Control)` key while selecting or adjusting, then the selection becomes pending-delete.<sup>9</sup>

<sup>8</sup> Actually, there is more than one kind of selection. What is described above is the most common kind, a *primary selection*; if you hold down the `(Get)`, `(Put)`, or `(Delete)` keys, then you are making a secondary selection; and the contents of the shelf are the third-level selection. All three are managed by the `selection_svc(1)` program on behalf of other tools; this program is started automatically when you bring up `suntools(1)`.

<sup>9</sup> If you are used to the selection always being pending-delete or you simply prefer this, then you can make this the default by setting *Adjust\_is\_pending\_delete* in the *Text* category of `defaultsedit`.

## Making Selections

Here are some miscellaneous tips:

- If there is no selection in a text subwindow, you can just “wipe” with the middle mouse button to make a selection.
- You can double-, triple-, and quadruple-click in text subwindows to select words, lines, or the entire text subwindow’s contents, respectively.
- You can adjust the sensitivity of text subwindows to multi-clicking using the *Text/Multi\_click\_space* and *Text/Multi\_click\_timeout* default `sedit` settings.

## Useful Operations on the Selection Using `get_selection`

The `get_selection(1)` program provides the most *ad hoc* method of working with the selection. Once you’ve made a selection, you can use `get_selection` to:

- save the selection in a file:

```
% get_selection > somefile
```

This is especially useful when you want to save part of a mail message in a file.

- pipe the selection to some other program, such as `lpr(1)`:

```
% get_selection | lpr -J' some title for the selection'
```

... or `spell(1)`:

```
% get_selection | spell
```

(this is more useful in a `cmdtool` where you can scroll through the misspelled words).

- use the selection’s contents directly on the command line; for example, if you want to look at a file using `more(1)` and its name is displayed somewhere on the screen, then you can select the filename and type

```
% more `get_selection`
```

## Problems with `get_selection`

One problem with `get_selection` is that if you type `get_selection` in the window with the selection, the selection goes away. Another is that `get_selection` is a lot to type.

A solution to both problems is to put the common operations you perform on selections in your `suntools` root menu; see *Souping Up Your Root Menu* below.

When using `get_selection` in a command, you must wait until the command completes before making another selection.

## 11.5. Secondary Get

The various secondary selection modes are somewhat intimidating. It makes it easier to learn and remember them if you work through the various operations in Chapter 9, *Advanced Editing*, applying them as you go to a text sample in `textedit`.

However, secondary Get is undoubtedly the most useful editing operation in SunView, so it is revisited here. The most common operation you need to perform in SunView is to grab some text elsewhere on the screen and insert it in the current window. First we give the basic way, then the fast way using secondary Get.

### The Basic Way to Copy Text

The basic way to do this is:

1. Select the desired text.
2. Press `[Put]`.
3. Move to the window you want the text to go into (if necessary) and set the insertion point where you want the text to go (if necessary).
4. Press `[Get]` in that window.

### The Fast Way

Secondary get is much faster; simply:

1. Hold down the `[Get]` key .
2. Make a selection anywhere on the screen .
3. Let up on the `[Get]` key (in any window).

The text is instantly copied.

### Move the Cursor Back

One pitfall with this is that after the operation the cursor is often in another window. If you want to repeatedly grab stuff to go into one window, you have to move the cursor back into that window each time. Should you forget, you can click in the `suntools` “root gray” background pattern; this clears the secondary selection, so you can safely let up on the `[Get]` key.

### Click\_to\_Type Mode

If you are in click-to-type mode, the *keyboard focus* stays put in one window, so you don't run into this problem. You can grab text from all over the screen and leave the mouse anywhere, while the keys you press on the keyboard are directed to that one window throughout. Read the `swin(1)` manual page for more information. You can set `Click_to_Type` in the *SunView* category of `defaultsedit` to *Enabled* if you want to use click-to-type mode by default.

However, many users find it confusing to have the cursor in one window while keystrokes appear in another. You can go into click-to-type mode by typing

```
% swin -c
```

and return to the default focus-follows-cursor mode by typing

```
% swin -m
```

Or, better still, you can add entries to your `rootmenu` to jump from one mode

to the other; see *User Interface* below.

## 11.6. Other Common Text Subwindow Solutions

### How Do I Read in a File Using mailtool?

If you use some other editor than `textedit(1)`, you probably have many such questions. First, understand that the `textedit` program is just a veneer over the SunView text subwindow package. If you don't understand some things about the text subwindow, you won't be able to use `mailtool`, `cmdtool`, or other programs that use the text subwindow, even if you choose not to use `textedit`.

In this particular case, use the general mechanism that the text subwindow package provides for inserting a file:

1. Type in the filename you want to include.
2. Hold down the **Control** key and select the filename with the left and middle mouse buttons. The filename will be grayed out, indicating that the next chunk of text that appears in the window will wipe it out: this is called a "pending-delete" selection, since it's about to go away.<sup>10</sup>
3. Choose 'Get from file' from the text subwindow menu. Hey presto, the file itself appears in place of its name.

If you want people to know the name of the file, type

```
here's a file called ----> name_of_file
[blank line]
```

Select the *name\_of\_file*, extend the selection to the blank line, and choose 'Get from file' from the menu. This way, the file appears on a new line after its name.

### Scanning Digests

If you are fortunate enough to be connected to the USENET news network, you may well receive digests of the form

<sup>10</sup> Apple Macintosh users who are used to this kind of selection can make it be the default by setting `TextAdjust_is_pending_delete` to `True` in `default.sedit`; see *Macintosh Users* below for other tips for ex-Macintosh users.



```

Subject: Info-Graphics Digest
Info-Graphics Digest Sun May 10 03:00:26 PDT 1987
...
Today's Topics:

Distributed animation summary
Looking for PHIGS and GKS-3D standards
IBM Graphics Cards and Monitors
...

...
Subject: Distributed animation summary

(text of message)
...

```

What you can do is view the list of subjects near the top, select an interesting one, and then choose Find (from the text menu, or press **F****ind** — usually **L9**) to locate it, then start reading.

To return back to the list of subjects, you can either click at the top of the scrollbar or press **F****ind** again.

Or, you can use the 'Split view' command in the text subwindow menu to create two views of the digest, and keep the list of subjects in the top view window while you read the stories of interest in the bottom window.

### Putting > in Front of Included Text (quick search and replace)

This should really be done with a permanent filter or an on-the-fly one (see Section 11.8, *The Power of .textswrc*, below). But here's another approach that's effective for small inclusions. It uses the "Again" feature of text subwindows. **A****gain** (usually **L2**) repeats everything that happened since you last clicked the mouse to set the caret, including forward and backward character/word/line deletes, and Put, Get, and Find operations. This allows you to repeat complex operations; in this case:

1. Select the newline at the end of the line *two* lines above the first line you want to indent with > .
2. Press **F****ind** (usually **L9**) or choose 'Find' from the text subwindow menu.
3. This should select the next newline, positioning the caret at the start of the first line to > over.
4. Type > **S****pace**.
5. Press **A****gain** (usually **L2**). This repeats the Find operation and bumps the next line over.
6. Press **A****gain** again and again ...

If you change a line you didn't want to bump over, then press **U****ndo** (usually **L4**).

Again works in many other common editing situations as well. You have to figure out in advance what the sequence of operations you want to repeat, but there's always the **U****ndo** key. Note that character, word and line erase in both

directions (usually **Delete**, **Control-W**, **Control-U**, and their **Shift**ed variants) are also repeated by **Again**, so that you can, for example, replace a particular word, the word before it and the two characters after it, then repeat the operation by pressing **Again**.

## 11.7. Souping Up Your Root Menu

The `suntools` root menu (it appears when you hold down the right mouse button over the "root gray" background) is very powerful. It's expressly designed for you to alter to suit your requirements. It provides an easy route to a system-wide set of command and control shortcuts. Chapter 15, *Modifying Your Root Menu*, explains how to change your root menu and gives some examples. Basically, set the `SunView/Rootmenu_filename` entry in `defaultsedit` to `~/rootmenu` (the code that creates the root menu understands the convention that `~` represents your home directory), and create the file `~/rootmenu`. The next time you start `suntools` you will get the new menu, and thereafter changes you make to `~/rootmenu` will immediately be visible the next time you bring up your root menu. Here are some more suggestions for customizing your `~/rootmenu` file.

**NOTE** *Entries in your `.rootmenu` file must be on one line; however, here many have been printed on several lines separated by "\" symbols for clarity. If you type these in, enter them on one line and don't type the \s. No line in your `.rootmenu` file should be more than 255 characters long.*

### Add Local Applications

The obvious additions to your `~/rootmenu` file are any applications you use that are not in the default root menu. These need not necessarily be different programs; it is useful to have some programs start up with particular command line options (window size, font, and so forth).

```
...
rolotool /usr/local/rolo
test ~/bin/SunViewtest
NeWS cmdtool -Wi -Wl NeWS /bin/csh -c "cd ~/NeWS; start"
tinytool cmdtool -Wt /usr/lib/fonts/fixedwidthfonts/sail.r.6
"24 x 80" cmdtool -Ww 80 -Wh 24 -Wl "24 x 80"
"Printer Queue" lpq
```

Note that commands that write to standard output, such as `lpq(1)` above, will write to the console if run from the root menu. You should have a `cmdtool` window as your console to take advantage of this.

### Operations on the Selection

A menu of the common things you do with the selection (See *Useful Operations on the Selection Using `get_selection`* above), such as printing it, spelling it, saving it to a scratch file, using it as a filename to edit or a machine name, etc.

The following example shows a pull-right menu for common actions on the selection: running the selection through `spell`, editing the selected filename, adding the selection to a scrap file, running `perfmeter` on the selected machine, and running a selected command through the shell.

**NOTE** *Watch for the use of `sh -c` in the examples below to run complex commands. The `.rootmenu` code only knows how to run a single command; it doesn't*

know about pipes, sequential program execution etc. If you want a menu item to invoke a complex command sequence that uses these features, you must either make it a separate shell script or explicitly run it from a shell. You do the latter by putting your complex command in quotes and passing it to `sh -c`.

Some of these commands take a while to complete, during which time you shouldn't make a new selection.

```
...
Selection MENU
"<.> Spell" sh -c "echo '..Spelling started:.'; \
get_selection | spell; echo '..Spelling done.^G'"
"<.> Edit" sh -c "textedit `get_selection`"
"<.> Save" sh -c "get_selection>>$HOME/Scrap; \
echo 'Scrap saved.^G'"
"<.> Meter" sh -c "perfmeter `get_selection`"
"<.> Do it" sh -c "get_selection | sh"
Selection END
```

Here's a menu that supplies a number of printing options.<sup>11</sup> You can quadruple-click in a text subwindow then choose one of these items to print a text subwindow's contents.

```
Printing MENU
"lpr it" sh -c "get_selection | lpr -h; \
echo 'Selection printed.^G'"
"Courier 8 Point" sh -c "get_selection | \
enscript -F Courier-Bold12 -f Courier8; \
echo 'Selection printed.^G'"
"Two-up Landscape" sh -c "get_selection | \
enscript -2r;echo 'Selection printed.^G'"
"Times 32 Point" sh -c "get_selection | \
enscript -F Courier-Bold12 -f Times-Roman32; \
echo 'Selection printed.^G'"
"Versatec Dump" sh -c "screendump | rastrepl | \
lpr -v -Pversatec; echo 'Screen dumped to Versatec.^G'"
"Laser Dump" sh -c "screendump | pssun -S 8 | \
lpr; echo 'Screen dumped to LaserWriter.^G'"
Printing END
```

## Standard Tools

It is straightforward to add the standard root menu of SunView tools in `/usr/lib/.rootmenu` to your own custom root menu as a pull-right menu:

```
"Std. Tools" MENU /usr/lib/.rootmenu
```

<sup>11</sup> These particular examples assume you have a Sun LaserWriter and the TranScript that supports it.

## User Interface

User interface tweaks such as changing between click-to-type and focus-follows-mouse mode.

```

Services MENU
 "ReDisplay All" REFRESH
 "Click to type" swin -c
 "Follow cursor" swin -m
 "Reset Input" input_from_defaults
Services END

```

## 11.8. The Power of .textswrc

In *Useful Operations on the Selection Using get\_selection* above, we discussed useful operations that you can perform on selections, using

```
get_selection | some_program
```

Then in *User Interface* we showed how you can perform some of these operations from your root menu. You can also use the text subwindow *FILTER* mechanism to operate on the selection, but with a new twist: the results of the operation (if any) appear in the text subwindow.

You do this by adding filters to your `~/ .textswrc` file; the filter is associated with a particular function key.

Sample filters for `.textswrc` and `.ttyswrc` are in `/usr/lib`. They are somewhat documented in Chapter 16, *Modifying your Environment*. Copy these over to your home directory, play around, make changes, start up a new `textedit` to see how they work, play around, make changes, ...

### Getting Started with .textswrc Filters

You should start by copying the sample `.textswrc` file in `/usr/lib` to your home directory; from the C-shell, type

```
% cp /usr/lib/.textswrc ~
```

Now try some of the filters in it: type

```
% textedit ~/.textswrc
```

Select parts of the file, and try out the various filters.

### How Filters Work

Filters are very simple. You press a function key, the filter is run; the filter's standard input (what is fed into it) is whatever is selected at the time (in any window, not just the text subwindow). The filter's standard output (what it sends out) appears where the caret is in the window in which you pressed the function key.

**NOTE** *Another way of looking at filters: Filters are just a way to hook up the stdin and stdout of a command to a selection and the caret, respectively.*

Some filters don't do anything with standard input, for example the "macro" keys described below; others don't generate any output.

**Pending-Delete or Not**

Most of the time you want the output from the filter to replace the selected text: for example, if you use the `capitalize` filter (mapped to `R4`) in the `.textswrc` file in `/usr/lib`, then you want to replace the selected words with the capitalized versions. If the selection is "ordinary" (appears black), then the output from `capitalize` will appear before or after the selected text, depending on the direction in which you adjusted the selection, but it won't replace it. However, if you make the selection pending-delete (appears grayed-out) (by holding down the `Control` key while selecting or adjusting), then the output from the `capitalize` filter will wipe out what you fed into it. See Section 11.4, *Working with Selections*, above for more information on pending-delete selections and selections in general.

**Useful `.textswrc` Filters****Multiple Means to the Same End**

All the operations on the selection that you can perform with `get_selection` (see *Useful Operations on the Selection Using `get_selection`* above) or from your `.rootmenu` (see *Souping Up Your Root Menu* above) can also be done using `.textswrc` filters.

You should create `FILTER` entries in your `~/ .textswrc` file for all the common activities you perform in text subwindows that you do not want to perform on selections made in other windows. The default set in `/usr/lib/ .textswrc` are oriented towards `troff(1)` users. Here are some other suggested filters you can try out.

**Keyboard "Macros"**

To save typing long key sequences, you can assign them to a function key in your `~/ .textswrc` file.

**Short Macros**

If the key sequence is a single line, you can use the `echo` command; for example, my brief signature is

```
R3 FILTER
/bin/echo -n S Page
```

(the `-n` option to `echo` prevents a newline after the text.

**Long Macros**

If the key sequence is longer, or is on multiple lines, the filter can just `cat(1)` a file; for example, my long signature is

```
R3 FILTER
/bin/cat /usr/tutorial/ssp/.sig1
```

where `/usr/tutorial/ssp/.sig1` contains

```
=S Page Tech Pubs (windows) spage@sun.COM 354-4688 M/S A4-49
{ucbvax, decwrl}!sun!spage
```

## A Scrap/Test Filter

Often you don't want to dedicate a key to a particular filter. One approach is to have one key mapped to `test`:

```
R11 FILTER
 /usr/tutorial/ssp/bin/test
```

If you change this file, *even while a program is running*, you will run the new version when you next press **R11**.

You can make such a filter even more useful if you have another filter that saves the current selection as the new `test` script:

```
R3 FILTER
 cat >! ~/bin/test; chmod 755 ~/bin/test
```

(The `!` forces the test file to be overwritten, which is necessary if you have `noclobber` set in your `~/ .cshrc` file.)

With this filter, you can type in a shell program on the fly, select it (do not make it pending-delete, since the filter doesn't output anything), press **R3** to save it to `~/bin/test`, press **Undo** to erase it from the text subwindow (it's saved to `~/bin/test` already), and the program can now be run by pressing **R11**. For example, if you need to change a `/usr/local` prefix in many lines scattered around a text file to `/proto/src`, you could create the `sed(1)` script

```
#!/bin/sh
sed -e 's;/usr/local;/proto/src;g'
```

## Running Any Command

An even better way of running an command as a filter is to have a filter that runs whatever command you select. You can do this by making a filter which executes the selection from a chosen shell.

### From the Bourne Shell

The Bourne shell, `/bin/sh` takes its standard input, executes it, and puts the result on the standard output. Thus, if you have a filter along the lines of

```
R6 FILTER
 /bin/sh
```

you can type in any sequence of commands compatible with the Bourne shell, select them, press **R6** and the commands will be run under the Bourne shell, with the output displayed in the text subwindow.

### From the C-Shell

For the C-shell, `/bin/csh`, the filter is slightly more complex:

```
R5 FILTER
 apply_filter
```

where `apply_filter` is

```
#!/bin/csh
eval $<
```

### Suggested Uses

The uses of such “run command” filters are limitless. You can type in a shell command line in a text subwindow and its result appears in the same text subwindow. Hence you can do most of the things you can do in a `cmdtool` simply by typing commands, selecting them pending-delete, and pressing your filter’s function key. For example, if you typed `date` and did this, today’s date from the `date(1)` command would appear; if you typed `grep $USER /etc/passwd` and did this, then your password file entry (if it exists on the local machine as well as in the Yellow Pages) will appear. You may well find it useful to write shell scripts and simple programs expressly for generating and inserting useful strings into text subwindows.

Here are some other uses.

#### Listing Files

To get a list of files, type `ls -C somedir/filespec`, or some such `ls` command, select it pending-delete, and press your filter’s function key. The `-C` option forces `ls` to output multi-column output.

#### Editing Multiple Files

Type, for example, `ls -C src/*.c` in `textedit`’s scratch subwindow, select the line pending-delete, and press your filter’s function key to generate a list of files for editing; then to edit a new file select the desired file in the scratch window, and choose ‘Load file’ in the main text subwindow.

### Running the Selection through Any Command

Often you want to run the selection through a particular command or sequence of shell commands. You can dedicate a function key to the particular command(s), but often you want to run code through such things on the fly.

You can achieve this effect by selecting the text you want to send to the command(s), then type

```
% get_selection | commands > /tmp/scrap
```

in a `cmdtool`. Then you can delete the selected text in the original text subwindow, select “`/tmp/scrap`” in the `cmdtool`, and choose ‘Get from file’ in the original text subwindow.

As usual, there are more straightforward ways to do this. The following shell script, `~/bin/exec.filter`, executes the first line of the primary selection, passing it the rest of the primary selection as data.

```

#! /bin/csh -f
echo "#! /bin/csh -f" > /tmp/exec.$$
echo $< >> /tmp/exec.$$
chmod u+x /tmp/exec.$$
/tmp/exec.$$
rm -f /tmp/exec.$$

```

### Running the Selection through a "Run a Command" Filter

There is a really tricky way to achieve the same effect using the "run a command" filters described in the *Running Any Command* above. Both the C-shell and Bourne shell can pass their standard input on to the program they are running as its standard input, using the << notation. Thus, if you want to format the lines in part of a text subwindow to be 60 characters long using `fmt(1)`:

1. Type `fmt -61 <<\anystring` (where *anystring* is some sequence of characters that doesn't appear anywhere in the text, for example `EOF`) on its own line just above the text you want to format.
2. Select this command and the text you want to run through it pending-delete.
3. Press your "run a command" function-key.

The rest of the selection after the first line is redirected to the `fmt` command until the characters *anystring* are reached or until the end of the selection. The `\` in front of *anystring* prevents the selected characters from being interpreted by the shell as they are run through `fmt` — read the `sh(1)` or `csh(1)` manual page for details. There is a limit to the number of characters that can be fed into a "run a command" filter this way, and you may run into problems with buffering on large selections or selections that do not end in a newline.

### Summary (an Example)

Say you want to indent lines in a text subwindow with `>>`. There are many ways to do this:

1. You can do it by repeating a find-replace cycle using the `Again` key, as explained above in *Putting > in Front of Included Text (quick search and replace)*. This is cumbersome for large files.
2. You could select the lines, type

```
% get_selection | sed -e 's/^/>> /' > /tmp/sed.tmp
```

in a `cmdtool` or `shelltool` to create an indented copy of the lines in `/tmp/sed.tmp`, then press `Delete` to remove the original lines, then select `/tmp/sed.tmp` in the shell window, then choose 'Get from file' in the original window to paste in the altered lines.

3. If you have

```
R13 FILTER
sed -e 's/^/>> /'
```

in your `.textswrc` file, just select the lines pending-delete and press



**(R13)**.

## 4. If you have

```
R11 FILTER
~/bin/test
```

in your `.textswrc` file, just change the shell script `~/bin/test` to

```
#!/bin/sh
sed -e 's/^/>> /'
```

select the lines in the original window pending-delete, and press **(R11)**.

## 5. If you have

```
R3 FILTER
cat >! ~/bin/test; chmod 755 ~/bin/test
```

in your `.textsrc` file, you can load up the `~/bin/test` filter from the above example with the `sed` shell script by typing it into any text subwindow, selecting it pending-delete, and pressing **(R3)**; then select the original text pending-delete and press **(R11)** as before.

## 6. Finally, if you have the Bourne shell “run a command” filter

```
R6 FILTER
/bin/sh
```

or the corresponding C-shell `apply_filter` in your `.textswrc` file, type

```
sed -e 's/^/>> /' <<EOF
```

above the lines you wish to indent, select pending-delete this command and the lines you wish to indent, and press **(R6)**.

## 11.9. Using `cmdtool`

`cmdtool` is the equivalent to a terminal, except that

- it keeps a scrollable log of your dialog with the computer
- you can change the size of the “screen”
- you can split the “screen”
- you can edit the command you are typing (most of the time)
- when necessary, it gives an application “fullscreen access”

The latter feature was added to `cmdtool` in Release 3.4. You should read the SunView enhancements section of that manual to learn more about how it works.

## Function Key Mappings

The `.ttyswrc` file allows you to map the behavior of function keys in `cmdtool`<sup>12</sup> and `shelltool`. It is described in Chapter 16, *Modifying your Environment*, which also suggests a standard version; unfortunately, some of its keys are on the left keypad, and SunView uses these keys for its own operations.

The `.ttyswrc` file lets you assign strings to function keys, but the strings are fixed and it can't run commands. There are two ways to do this, `mapi` functions that act as if you are typing the characters into the shell yourself, and `mapo` functions that act as if the shell is displaying the characters.

## Macros with `mapi`

You can easily duplicate the “macro” keys from `.textswrc` using the `mapi` function, and you can use `termcap(5)`-style notation for special characters. For example, the equivalents for the signature lines above are:

```
Signatures
mapi R1 =S Page Tech Pubs (windows) \
 354-4688 M/S A4-49 ssp@polar
mapi R2 =S Page Tech Pubs (windows) \
 spage@sun.COM (415)354-4688 M/S A4-49\n \
```

These are one-line commands that have been broken up onto several lines here for clarity, but note that the `R2` mapping has `\n` in the middle of it to generate an actual newline when the `R2` function key is pressed.

## Escape Sequences Using `mapo`

If you use `mapo` for macros, you will quickly discover that the characters appear in the `cmdtool`, but the shell never sees the characters. This is because `mapo` makes `cmdtool` only display the characters, instead of pretending that you typed them. This is only useful because `cmdtool` and `shelltool` respond to certain escape sequences when they “display” them by changing the frame header, resizing their frame, etc. The entire list of escape sequences is listed on pages 160-161 of the ; the next section lists a monster `.cshrc` file that uses these escape sequences to pack the frame header with useful information.

## Sample `~/cshrc`

You usually run a shell in each `cmdtool` or `shelltool` which you start up. In the C-shell, you can use the powerful `alias` facility (read `csh(1)` for more information) to create commands that use the escape sequences recognized by `cmdtool`.

Here is part of a “monster” `.cshrc` file that sets up some powerful aliases, and also demonstrates some other useful concepts relating to using shells in the Sun-View environment. (This was originally published in the *Software Technical Bulletin*, sent to customers on software support contracts.)

```
#!/bin/csh
@(#)cshrc 2.1
monster cshrc -- everything you might ever want to do when you
use csh.
```

<sup>12</sup> `cmdtool` picks up function key settings from both `.textswrc` and `.ttyswrc`, with the latter taking

```

if we are running a script, don't source .cshrc for speed.
prompt is set if we are interactive.
term is set if we have a tty attached (needed for at)
if (! $?prompt) exit
if (! $?term) exit

..

quick pushd/popd -- pushd should really be +, but that takes requires shift
~ is shorthand for pushd $HOME, which tends to happen often
alias - popd
alias + pushd
alias ~ pushd ~

..

quick sunview compiling
alias ccsv "cc \!* -lsuntool -lsunwindow -lpixrect"
alias ccsvg "cc \!* -g -lsuntool -lsunwindow -lpixrect"

..

This is complex. What this does is set things up so if you are running
SunView the hostname and directory stack are in the window stripe.
if you aren't running SunView, then it places the current directory and
hostname in the prompt.
set tty = `tty`
set hostname = `hostname`
set console = '<< CONSOLE >>'
set promptchar = "%"
set lastdir = $cwd
WINDOW_PARENT active if you are in sunview
if ($?WINDOW_PARENT) then
 if ($?NEWSSERVER) then # NEWS is running
 goto getout
 end
command string to set the frame header
alias hdr `echo -n "`[l\!*^\`\"`
command string to set the icon label
alias ihdr `echo -n "`[L\!*^\`\"`
setbar actually does the work
alias setbar `hdr "${hostname}: `dirs`"; ihdr $cwd:t`
for rlogin -- if the hostname of the window isn't the hostname of
hostname for my machine, then put the hostname in the icon label
instead of the directory. Note that the default hostname is hardcoded,
so this fails if you run SunView on multiple machines, but
I don't know a way around that.
if (${hostname} != plaid) alias setbar `hdr "${hostname}: `dirs`"; ihdr ${hostname}
HACK ALERT. If the window is on tty0, assume it is the console, and
flag it as such. This assumes (1) that you start your console window

```

preference in case both files map a function key.

```

first in your .suntools file, (2) that nothing else is taking up tty0
when you start the window system, and (3) that nothing else unusual is
going on. There is, unfortunately, no way to find out if a given window
is a console short of looking in kmem, so we have to guess. Believe it
or not, this hack works most of the time.
 if (${tty} == "/dev/tty0" && ${hostname} == myhostname)
 alias setbar 'hdr "${console} `dirs`"'
 if (${tty} == "/dev/tty0") echo -n "`[]Lconsole`[\a"
 set prompt = "${hostname}$promptchar "
else
if you aren't in windows, use the prompt.
 alias setbar 'set prompt = "${hostname}:$cwd$promptchar "'
endif

All of the above just set things up. These aliases do the work.
. resets the window stripe if it needs it.
cd, pushd, and popd are the only commands that change the working
directory and directory stack, so intercept them and make sure things
get updated. Finally, run setbar to initialize things for the first time.
alias . 'dirs;setbar;jobs'
alias cd 'set lastdir = $cwd; cd \!* ; setbar'
alias popd 'set lastdir = $cwd; popd \!* ; setbar'
alias pushd 'set lastdir = $cwd; pushd \!* ; setbar'
alias c- 'set x=$cwd; cd $lastdir; setbar; set lastdir=$x'
setbar
getout:
 exit 0

```

## 11.10. Performance

Sun's fastest workstation is ten times as fast as the slowest one and SunView runs on all Sun workstations. If you are running SunView on a Sun 2/50, without extra memory and with an overloaded fileserver, you may wish that you could improve SunView's performance. The following paragraphs show you how.

You can customize your working environment to improve the interactive performance of SunView. The improvements fall into three categories:

- reducing CPU overhead
- optimizing memory usage
- improving interactive response

### Reduce CPU Overhead

You want to reduce the number of tasks your workstation's CPU (Central Processing Unit) has to perform so more of its cycles are available.

### Schedule Fewer Tasks for the CPU

Various options include: turning off the second hand in `clock(1)`, turning off the blinking caret in text subwindows (set `Text/Blink_caret` to `False` in `defaultsedit`), reducing the sampling rate in `perfmeter(1)` (specify a higher sample time using its `-s sample-time` option), and avoiding running programs such as `perfmom` and `traffic` that update the display continuously.

**Don't Run `comsat`**

The `comsat(8C)` daemon is a server process which listens for reports of incoming mail. If you comment out the line

```
comsat udp /usr/etc/in.comsat
```

in your machine's `/etc/servers` file, SunView performance will be enhanced. You will not be able to run the `biff(1)` program to notify you of incoming mail, but you can use the C-shell variable `mail` or the Bourne shell variables `MAIL`, `MAILPATH`, and `MAILCHECK` to get notified when mail arrives.

**Don't Make Other Tools Repaint**

When tools overlap, every time you close, hide or move one, its neighbors must repaint themselves. If you can arrange the tools in your `.suntools` file so that they do not overlap, these operations will take less time, since only the background needs to be repainted.

**Optimize Memory Usage**

Your Sun workstation can run a mix of programs that together require much more memory than the physical memory (RAM) it has available. However, in sharing the physical memory between applications, chunks of programs that aren't in use must be copied to and from the disk, which slows down performance. This is why a tool that you do not interact with for a while may be sluggish when you move the cursor into it — much of the program may need to be copied in from disk. The more efficiently you use physical memory, the less this copying (swapping and paging) needs to take place.

**Use System Fonts**

Some fonts are pre-compiled into the SunView libraries. If you specify `screen.r.13`, `screen.b.12`, `screen.r.7`, or `screen.r.11` as your font then the system can share the font among applications. These fonts are loaded into the text segment already and are sharable.

You don't need to give the full `/usr/lib/fonts/fixedwidthfonts` path to these "built-in" fonts; just specify the name alone to `defaultsedit` (in *SunView/Font* and/or *Text/Font*) or the command-line argument `-Wt fontname`

**Merging Applications**

If you type

```
% ls -l /usr/bin | more
```

you can see that all the common SunView programs are links to one of two so-called *toolmerge* programs, `suntools` and `othertools`. If you only run programs from one merge, only one toolmerge's text segment needs to be in memory, and performance is improved. As soon as you run a program that is not in the same merge, two large programs must contend for the limited amount of physical memory in your machine, often leading to excessive paging. The split of programs between `suntools` and `othertools` reflects Sun's guess as to which tools users commonly run together; for example, you tend to run `textedit`, `cmdtool` and `mailtool` simultaneously on a regular basis, but tend not to run `defaultsedit` frequently and don't use `lockscreen` at the same time as other programs.

You can make a better judgement as to the split and build your own custom *tool-merge*. Moreover, if you run your own custom tools alongside the standard set of SunView applications, you should *strongly* consider merging them into a *tool-merge* if possible. Section 4.8 of the *Release 3.2 Manual, SunView Enhancements*, has instructions on how to create your own custom set of merged programs.

#### Reconfigure Your Kernel

The kernel (`/vmunix`) is not paged or swapped, so the larger it is, the more it makes part of your physical memory permanently unavailable to applications. The default `GENERIC` kernel includes code to support many devices and options, consequently it is very large. As recommended in the installation guide for every release, you should *always* build your own kernel that excludes unused devices, whether you run SunView or not.

Read the chapter called *Reconfiguring Your Kernel* in your most recent *Release 3.x Manual* or *Installing UNIX* manual for information on how to reconfigure your kernel.

#### Improve Interactive Response

Taking the steps above will improve the performance of your system, but there are other steps you can take that will make your system “feel” more responsive.

#### Use Retained Subwindows

Try making your text subwindows and tty subwindows retained (set *Text/Retained* and *Tty/Retained* to *Yes* in `defaultsedit`) and see if this helps. If a window is retained, the system keeps an offscreen copy of the window in memory, so if the window on-screen needs to be repainted, the off-screen window is copied onto it; otherwise, the contents of the window have to be recalculated and redisplayed. You are more likely to notice an improvement if you have a monochrome machine and have spare physical memory, but there are no definite rules for this — try it on your workstation and see.

#### Learn Keyboard Accelerators

It takes less time and CPU overhead for you to press a key than to bring up a menu. Learn to use the keyboard accelerators such as `Expose` (usually `L5`) and `Close` (usually `L7`); these are summarized on the *SunView Quick Reference* cards in the *Windows and Window-Based Tools: Beginner's Guide* manual.

#### Increase Mouse “Gain”

By adjusting the mouse motion ceilings and scalings in the *Input* category of `defaultsedit`, you can increase the responsiveness of your mouse and reduce the wrist motions you need to make to position the mouse.

### 11.11. Miscellaneous

#### Macintosh Users

If you are familiar with the Apple Macintosh's style of user interface, there are some differences between it and SunView, some of which you can eliminate.

## Editing Operations

Terminology is different in SunView:

|           |   |        |
|-----------|---|--------|
| Copy      | ⇒ | Put    |
| Paste     | ⇒ | Get    |
| Cut       | ⇒ | Delete |
| Clipboard | ⇒ | shelf  |

Many Macintosh applications let you display the Macintosh clipboard in a window; in SunView you can use `get_selection 3` to see the shelf's contents.

The selection is always pending-delete on the Macintosh — whatever you type replaces it. You can set `Text/Adjust_is_pending_delete` to `True` to get similar behavior in SunView.

## Unpacking Shell Archives

If you get a message containing a shell archive (`shar`) file, instead of editing it, saving the `shar` part, running it through `/bin/sh`, etc., in `mailtool` you can just select the desired part and send it to `/bin/sh` using `get_selection(1)`:

1. Select the start of the `shar` file (`#!`).
2. Move to the end of the file (middle-click near the bottom of the scrollbar's bar, or press `Control-Return`).
3. Extend the selection (middle button) to the end of the `shar` file.
4. In a shell window type

```
% cd desired_directory
% get_selection | /bin/sh
```

Be careful not to change the selection while the command is running.





## SunView Programmer's Guide

|                                                                          |     |
|--------------------------------------------------------------------------|-----|
| SunView Programmer's Guide .....                                         | 193 |
| 12.1. Errata and Addenda for the SunView Programmer's Guide .....        | 193 |
| Maximum Attribute-Value List Size .....                                  | 193 |
| Fitting Frames Around Subwindows .....                                   | 193 |
| Restrictions on Pop Up Windows .....                                     | 193 |
| Window Positioning .....                                                 | 194 |
| WIN_EXTEND_TO_EDGE Invalid for Frames .....                              | 194 |
| Zero WIN_X .....                                                         | 194 |
| Changing Subwindow Size .....                                            | 194 |
| Attribute Ordering .....                                                 | 194 |
| File Descriptor Usage Table .....                                        | 194 |
| Canvas Resizing .....                                                    | 195 |
| LOC_TRAJECTORY Events .....                                              | 195 |
| TEXTSW_INSERT_MAKES_VISIBLE Does Not Affect<br>textsw_insert() .....     | 196 |
| Cautions for Default Panel Item Layout .....                             | 196 |
| Menu Generate Proc Attributes .....                                      | 196 |
| Nonexistent WIN_ICON Attribute .....                                     | 197 |
| Creating Scrollbars .....                                                | 197 |
| Differences Between notify_set_signal_func() and<br>signal(3) .....      | 197 |
| Correct PANEL_ITEM_Y_GAP Value .....                                     | 198 |
| Undocumented PANEL_MARK_IMAGE and<br>PANEL_NOMARK_IMAGE Attributes ..... | 198 |

|                                          |     |
|------------------------------------------|-----|
| Correct TEXTSW_BLINK_CARET Default ..... | 199 |
| Revised typein.c Example Program .....   | 199 |

---

## SunView Programmer's Guide

### 12.1. Errata and Addenda for the SunView Programmer's Guide

The following pages contain errata and addenda for the Sun Release 3.2 version of the *SunView Programmer's Guide* (Part Number: 800-1345-10).

#### Maximum Attribute-Value List Size

Add to Page 28 in Chapter 3, *Interface Outline*, under "Attribute-based Functions:"

The maximum length of attribute-value lists supported by the SunView packages (see `ATTR_STANDARD_SIZE` in `<sunwindow/attr.h>`) is 250. If the number of attributes in a list you pass to SunView exceeds this size, the attribute package prints

```
Number of attributes(nnn) in the attr list exceeds
the maximum number(nnn) specified. Exit!
```

on standard output and exits with exit status 1.

#### Fitting Frames Around Subwindows

Many 3.0 SunView programs do not explicitly size their subwindows — they rely on getting the default window size.

Add the following to Page 38 in Chapter 4, *Windows*, after the paragraphs on `window_fit()`:

Since Release 3.2, if you use `window_fit()` or its variants for sizing the width and height of a frame, you need to be careful that the subwindows have some specified size, or they will be shrunk very small by the `window_fit()` call. Usually subwindows have a fixed size in one or both dimensions, or they are sized to be a percentage of the frame's size. The default size of a frame is that it encloses an area 34-rows by 80-columns in its default font.

#### Restrictions on Pop Up Windows

Add the following note to Page 45 in Chapter 4, *Windows*, at the end of "Popup Confirmer."

There are some restrictions on popup frames displayed using `window_loop()`:

- You can only have one subwindow in the popup frame.
- The only subwindow types that work properly are canvases and panels.

These limitations do not apply to non-blocking popups.

## Window Positioning

### WIN\_EXTEND\_TO\_EDGE Invalid for Frames

Add the following footnote to Page 46 in Chapter 4, *Windows*, after “Specifying Subwindow Sizes and Positions.”

It is meaningless to set the width or height of a frame to WIN\_EXTEND\_TO\_EDGE, and it will interfere with subwindow behavior.

### Zero WIN\_X

Add the following to Page 48 in Chapter 4, *Windows*, in “Specifying Subwindow Sizes and Positions:”

The origin of the window system coordinate space is always in the upper left corner, so change ‘right’ to ‘left’ in the following sentence:

The computations for these attributes take the borders and header of the frame into account, so that specifying WIN\_X and WIN\_Y of 0 will result in the subwindow being placed correctly at the upper right corner of the frame.

### Changing Subwindow Size

Add the following section to Page 48 in Chapter 4, *Windows*, after “Specifying Subwindow Sizes and Positions:”

If you programmatically change the size or position of subwindows after you create them, you must explicitly re-specify the origin of any subwindows that are below or to the right of the altered subwindows, even if you specified the positions of these other subwindows using relative position attributes, such as WIN\_BELOW. This is necessary because subwindows are not automatically laid out again when the positions and sizes of other subwindows are changed. They are only laid out again if the frame changes size. When re-specifying the layout of the other subwindows, you *can* use relative position attributes such as WIN\_BELOW.

### Attribute Ordering

Make these changes to Page 51 in Chapter 4, *Windows*, under “Command-line Arguments:”

In the two examples, the order of the arguments to FRAME\_ARGS should be reversed, so that argc comes *before* argv, instead of argv, argc as indicated.

### File Descriptor Usage Table

Page 54 in Chapter 4, *Windows*, Table 4-2, *SunView File Descriptor Usage*, is missing some items. Add the following:

|        |   |                                         |
|--------|---|-----------------------------------------|
| PANEL  | 1 | 1 for the window                        |
| MENU   | 0 | fullscreen access uses the window's fd  |
| CURSOR | 0 | Most cursors are managed by the kernel. |

## Canvas Resizing

Add the following to the end of Section 5.6 on Page 65 in Chapter 5, *Canvases*:

Note that, as described in Section 4.8, "Attribute Ordering," the canvas attributes are evaluated before the generic window attributes. This means that if you want to set the window size, then disable automatic sizing of the canvas, you must *first* set the window size, and *then*, in a separate `window_set()` call, disable `CANVAS_AUTO_SHRINK` and/or `CANVAS_AUTO_EXPAND`. If you do both in the same call, the auto-sizing will be turned off before the window size is set, so that the canvas size will not match the window size you specify. Here is an example of how to do it correctly:

```
canvas = window_create(frame, CANVAS,
 WIN_HEIGHT, 400,
 WIN_WIDTH, 600,
 0);

window_set(canvas,
 CANVAS_AUTO_SHRINK, FALSE,
 CANVAS_AUTO_EXPAND, FALSE,
 0);
```

## LOC\_TRAJECTORY Events

In Section 6.2, "Events," of Chapter 6, *Handling Input*, the documentation is muddled on the distinction between `LOC_MOVE` and `LOC_TRAJECTORY` events. As stated on Page 77 in "Locator Motion Events," `LOC_MOVE` and `LOC_DRAG` events are collapsed (so that clients receive the most recent mouse location) unless you enable `LOC_TRAJECTORY` events. However, the rest of the description on Page 78 is wrong. When you enable `LOC_TRAJECTORY` events, clients do not receive any `LOC_MOVE` events, they get `LOC_TRAJECTORY` events instead, together with `LOC_DRAG` events if you have enabled them..

Thus, Table 6-1, *Event Codes*, on Page 76 is wrong. In it, change the description of `LOC_TRAJECTORY` events from

Causes consecutive `LOC_MOVES` to be collapsed

to

Inhibits the collapse of mouse motions; clients receive `LOC_TRAJECTORY` events for every locator motion the window system detects.

Also, on Page 78, change

Note that when you ask for `LOC_TRAJECTORY` events, you still only get `LOC_MOVE`'s or `LOC_DRAGS`'s in the input stream — however, you get more of them.

to

Note that when you ask for `LOC_TRAJECTORY` events, you get (many!) `LOC_TRAJECTORY` events in place of `LOC_MOVE`'s, but you still get `LOC_DRAG` events if you have enabled them.

**TEXTSW\_INSERT\_MAKES\_VISIBLE Does Not Affect**  
`textsw_insert()`

Add the following note to Page 118 in Chapter 8, *Text Subwindows*, in Section 8.2, and to Page 322 in Chapter 18, *SunView Interface Summary*, in Table 18-25, *Text Subwindow Attributes*:

`TEXTSW_INSERT_MAKES_VISIBLE` applies only to text inserted by the user typing from the keyboard; `textsw_insert()` never scrolls the window. You should use `textsw_possibly_normalize()` to ensure that the desired section is visible, for example:

```
textsw_possibly_normalize(mytextsw,
 (Textsw_index) window_get(mytextsw, TEXTSW_INSERTION_POINT);
```

**Cautions for Default Panel Item Layout**

Add the following comment and caution to Page 141 in Chapter 9, *Panels*, in Section 9.2, after "Default Item Positioning:"

The default position for the next item is computed after an item is created. But if a client calls `panel_set()` after creating an item in such a way that the enclosing rectangle of the item is altered, the default position for the next item will *not* be recomputed. So, for example,

```
item = panel_create(panel, PANEL_MESSAGE, 0);
panel_set(item, PANEL_LABEL_STRING, "Hi", 0);

item1 = panel_create(panel, PANEL_MESSAGE,
 PANEL_LABEL_STRING, "There",
 0);
```

will result in "There" overlapping "Hi."

**CAUTION**

**Choice items currently have problems with item "creep."** Each time the label of a choice item is set, the position of the item will be evaluated. If the value's position has not been fixed (with `VALUE_X/Y`), the value is positioned after the label. The problem is that the label is baseline-adjusted for a choice item. If the item position is not given when the label is set, the choice item will creep down because of the baseline adjustment.

**Menu Generate Proc Attributes**

On Page 203 in Chapter 11, *Menus*, under "Generate Procedures"

There is no `MENU_ITEM_GEN_PROC` attribute; this should be `MENU_GEN_PROC_ITEM` instead.

On Page 207 in Chapter 11, *Menus*, under "Pullright Generate Procedure"

There is no `MENU_GEN_PULLRIGHT_PROC` attribute; this is the `MENU_GEN_PULLRIGHT` attribute referred to elsewhere. The sample call to specify a menu item's pullright generate procedure should be

```
Menu my_pullright_gen();

menu_set(menu_item, MENU_GEN_PULLRIGHT, my_pullright_gen, 0);
```

**Nonexistent WIN\_ICON Attribute**

On Page 227 in Chapter 13, *Icons*,  
WIN\_ICON should be FRAME\_ICON.

**Creating Scrollbars**

On Page 235 in Chapter 14, *Scrollbars*, in Section 14.3, “Creating, Destroying and Modifying Scrollbars,” add the following after the example of a non-default scrollbar:

If you set the SCROLL\_THICKNESS attribute then you *must* also set the SCROLL\_DIRECTION of the scrollbar, since the dimension of the scrollbar that is altered by SCROLL\_THICKNESS depends on the orientation of the scrollbar.

**Differences Between**

notify\_set\_signal\_func()  
and signal(3)

On Page 253 in Chapter 16, *The Notifier*, add the following information to “A signal() Replacement for Notifier Compatibility:”

The Notifier routine notify\_set\_signal\_func() does not fully emulate the signal(3) function. It should deal with error states in a similar manner as signal(3). Errors from signal(3) are indicated by a -1 return value, and the value of errno is set to EINVAL.

The errors for notify\_set\_signal\_func() are not communicated back to the caller, but error messages are printed. For example, if the signal number is not valid, the Notifier prints

```
Bad signal number
```

but its return value indicates success; the signal(3) system call does not print a message, but returns -1 and sets errno to EINVAL. As another example, if SIGKILL or SIGSTOP are ignored or a handler supplied, the Notifier prints

```
Notifier assertion botched: Unexpected error: sigvec
```

but its return value indicates success, while signal(3) does not print a message, returns value of -1, and sets errno to EINVAL.

The workaround is to use the following replacement function for signal(3) instead:

```

#include <sunwindow/notify.h>
#include <errno.h>

int (*
signal(sig,func))()
int sig, (*func) ();
{
 if ((sig < 1 || sig > NSIG) ||
 (sig == SIGKILL || sig == SIGSTOP)) {
 errno = EINVAL;
 return(BADSIG);
 }
 if (sig == SIGCONT && func == SIG_IGN) {
 errno = EINVAL;
 return(BADSIG);
 }
 return ((int(*)())notify_set_signal_func(sig,func,
 sig,NOTIFY_ASYNC));
}

```

**Correct**  
**PANEL\_ITEM\_Y\_GAP Value**

Change the following on Page 301 in Chapter 18, *SunView Interface Summary*, in Table 18-15, *Panel Attributes*.

The correct value for the default `PANEL_ITEM_Y_GAP` is 5 pixels, not 10. (The *Panels* chapter is correct, the table is wrong.)

**Undocumented**  
**PANEL\_MARK\_IMAGE and**  
**PANEL\_NOMARK\_IMAGE**  
**Attributes**

The panel choice attributes `PANEL_MARK_IMAGE` and `PANEL_NOMARK_IMAGE` are attributes corresponding to `PANEL_MARK_IMAGES` and `PANEL_NOMARK_IMAGES`. They let clients set the selected and unselected images, respectively, for an individual mark. Modify Table 18-17, *Choice and Toggle Item Attributes*, on Page 304 in Chapter 18, *SunView Interface Summary*, as follows:

|                                 |                |                                                                                                                                              |
|---------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <code>PANEL_MARK_IMAGE</code>   | int, Pixrect * | Image to mark choice specified by the first argument when it is selected. Default is push-button image:<br><images/panel_choice_on.pr>.      |
| <code>PANEL_NOMARK_IMAGE</code> | int, Pixrect * | Image to mark choice specified by the first argument when it is not selected. Default is push-button image:<br><images/panel_choice_off.pr>. |



|                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Correct</b><br>TEXTSW_BLINK_CARET    | Change the following on Page 321 in Chapter 18, <i>SunView Interface Summary</i> , in Table 18-25, <i>Text Subwindow Attributes</i> .                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Default</b>                          | The text subwindow caret blinks by default, so the default for TEXTSW_BLINK_CARET is TRUE, not FALSE.                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Revised typein.c Example Program</b> | On Page 381 in Appendix A, <i>Example Programs</i> .<br>The new attributes for tty subwindows (described in Section 6.2.6, <i>New Tty Subwindow Attributes</i> , above) make the sample program <i>typein</i> supplied in 3.2 needlessly complex, and eliminate the requirement for a dummy program, <i>loop-back</i> , running in its tty subwindow. Here is a revised version of <i>typein</i> that uses the new TTY_TTY_FD attribute and TTY_ARGV_DO_NOT_FORK value for TTY_ARGV. The new version is included in the optional software for 3.4. |

```

/*****/
#ifdef lint
static char sccsid[] = "@(#)typein.c 1.5 87/01/07 Copyr 1986 Sun Micro";
#endif
/*****/

#include <stdio.h>
#include <suntool/sunview.h>
#include <suntool/canvas.h>
#include <suntool/tty.h>
#include <ctype.h>

static Frame frame;
static Canvas canvas;
static Tty tty;
static Pixwin *pw;

static Notify_client my_client;

#define STDIN_FD 0
#define STDOUT_FD 1
#define BUFSIZE 1000

main(argc, argv)
int argc;
char **argv;
{
 static Notify_value read_input();
 int tty_fd;

 frame = window_create(NULL, FRAME,
 FRAME_ARGS, argc, argv,
 WIN_ERROR_MSG, "Cannot create frame",
 FRAME_LABEL, "typein",
 0);

 tty = window_create(frame, TTY,
 WIN_PERCENT_HEIGHT, 50,

```

```

 TTY_ARGV, TTY_ARGV_DO_NOT_FORK,
 0);

tty_fd = (int>window_get(tty, TTY_TTY_FD);
dup2(tty_fd, STDOUT_FD);
dup2(tty_fd, STDIN_FD);

canvas = window_create(frame, CANVAS,
 0);
pw = canvas_pixwin(canvas);

/*
 * Set up a notify proc so that whenever there is input to read on
 * stdin (fd 0), we are called to read it.
 * Notifier needs a unique handle: give it the address of tty.
 */
my_client = (Notify_client) &tty;
notify_set_input_func(my_client, read_input, STDIN_FD);

printf("Enter first coordinate:\nx? ");

window_main_loop(frame);
exit(0);
}

/*
 * This section implements a simple application which writes prompts to
 * stdin and reads coordinates from stdout, drawing vectors with the
 * supplied coordinates. It uses a state machine to keep track of what
 * number to read next.
 */
#define GET_X_1 0
#define GET_Y_1 1
#define GET_X_2 2
#define GET_Y_2 3
int state = GET_X_1;
int x1, y1, x2, y2;

/* ARGSUSED */
static Notify_value
read_input(client, in_fd)
Notify_client client; /* unused since this must be from ttysw */
int in_fd; /* unused since this is stdin */
{
 char buf[BUFSIZE];
 char *ptr, *gets();

 ptr = gets(buf); /* read one line per call so that we
 don't ever block */
 /* ***** does this matter any more?? */

 /* handle end of file */
 if (ptr==NULL) {

```

```

/* Note: could have been a read error */
window_set(frame, FRAME_NO_CONFIRM, TRUE, 0);
window_done(tty);
} else {
 switch (state) {
 case GET_X_1:
 if (sscanf(buf, "%d", &x1) != 1) {
 printf("Illegal value!\nx? ");
 fflush(stdout);
 } else {
 printf("y? ");
 fflush(stdout);
 state++;
 }
 break;
 case GET_Y_1:
 if (sscanf(buf, "%d", &y1) != 1) {
 printf("Illegal value!\ny? ");
 fflush(stdout);
 } else {
 printf("Enter second coordinate:\nx? ");
 fflush(stdout);
 state++;
 }
 break;
 case GET_X_2:
 if (sscanf(buf, "%d", &x2) != 1) {
 printf("Illegal value!\nx? ");
 fflush(stdout);
 } else {
 printf("y? ");
 fflush(stdout);
 state++;
 }
 break;
 case GET_Y_2:
 if (sscanf(buf, "%d", &y2) != 1) {
 printf("Illegal value!\ny? ");
 fflush(stdout);
 } else {
 printf("Vector from (%d, %d) to (%d, %d)\n",
 x1, y1, x2, y2);
 pw_vector(pw, x1, y1, x2, y2, PIX_SET, 1);
 printf("\nEnter first coordinate:\nx? ");
 fflush(stdout);
 state = GET_X_1;
 }
 break;
 }
}
return(NOTIFY_DONE);
}

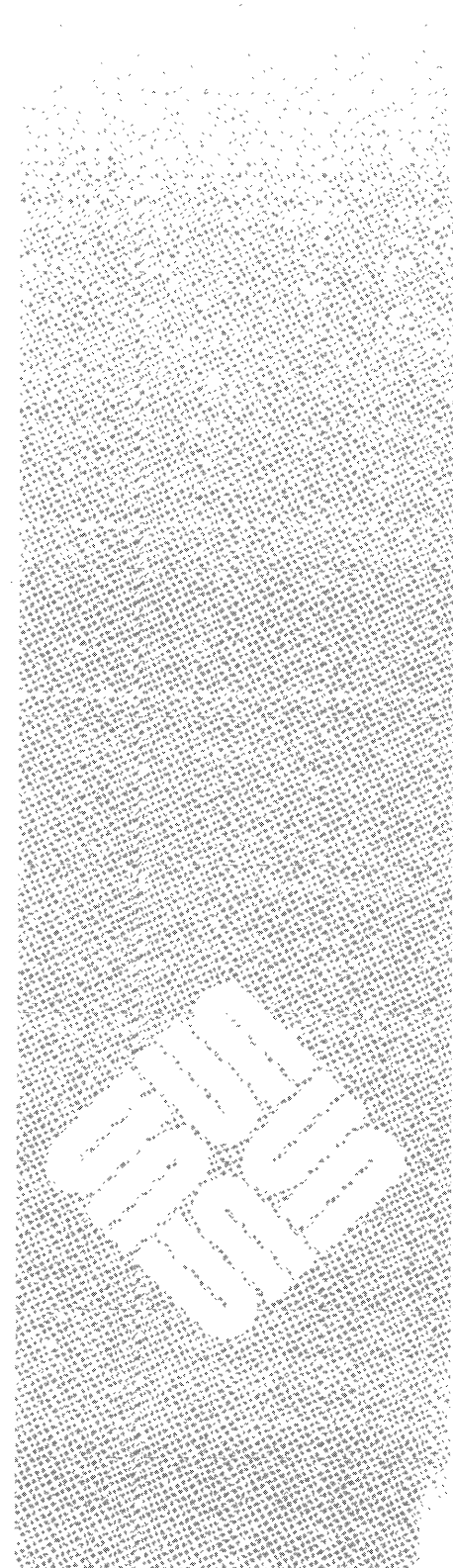
```



---

## SunView System Programmer's Guide

|                                                                             |     |
|-----------------------------------------------------------------------------|-----|
| SunView System Programmer's Guide .....                                     | 205 |
| 13.1. Errata and Addenda for the SunView System Programmer's<br>Guide ..... | 205 |
| Scrollbar Chapter .....                                                     | 205 |





---

## SunView System Programmer's Guide

### 13.1. Errata and Addenda for the SunView System Programmer's Guide

The following pages contain errata and addenda for the Sun Release 3.2 version of the *SunView System Programmer's Guide* (Part Number: 800-1342-10).

#### Scrollbar Chapter

Page 187 in Chapter 15, *Scrollbars*, the third sentence from the top,

For canvases and text subwindows...

should read

For canvases and panels...

In the same chapter, on Page 191

... defaults to 4.

should read:

... defaults to 4 pixels.





---

## System Administration Manual

|                                                                                                 |     |
|-------------------------------------------------------------------------------------------------|-----|
| System Administration Manual .....                                                              | 209 |
| 14.1. System Administration Manual Addenda .....                                                | 209 |
| /etc/ttys File .....                                                                            | 209 |
| Run <code>catman</code> to Access Online Manual .....                                           | 210 |
| Hooking Up a Serial ASCII Printer .....                                                         | 210 |
| Hooking Up a Printer to a VPC-2200 Multibus Board .....                                         | 210 |
| Major Changes in Disk and Directory Layout in Release 3.0<br>and Later Releases .....           | 211 |
| Preparing a Previously-Used Client Partition .....                                              | 211 |
| <code>dump</code> Cannot Be Used to Dump an NFS-mounted File System .....                       | 212 |
| How to Use <code>yppasswd</code> Versus <code>passwd</code> .....                               | 212 |
| Print Filter Hook .....                                                                         | 212 |
| <code>fstab</code> Entry for a Sun-2/3Com Diskless Client Mounting<br>from a Sun-3 Server ..... | 214 |
| How to Add a New <code>yp</code> Slave Server Not in the Original Set .....                     | 214 |
| <code>uucp</code> <code>L.cmds</code> File .....                                                | 215 |



---

## System Administration Manual

### 14.1. System Administration Manual Addenda

#### /etc/ttys File

The following pages are Errata and Addenda for Sun Release 3.0 version of the *System Administration for the Sun Workstation* manual (Part Number: 800-1323).

#### Pages 146-147

The sample /etc/ttys file on Page 146 is incorrect and should read as follows:

```
12console
12ttya
02ttyb
12ttys0
1fttys1
.
.
.
```

The last line of the example should read `1fttys1` instead of `14ttys1`. The change of 4 to f matches the changes to the /etc/gettytab entries below; that is, the `1fttys1` line indicates `ttys1` should be enabled for logins at 1200-baud.

Change the sample /etc/gettytab file on Page 147 to read as follows:

```
c|std.300|300-baud: :nd#1:cd#1:sp#300:
f|std.1200|1200-baud: :fd#1:sp#1200:
2|std.9600|9600-baud: :sp#9600:
```

The corresponding sentence in the paragraph following the /etc/gettytab file excerpt on Page 147 should read as follows:

Therefore, a 'c' as the second character on the line would set the rate to 300, an 'f' would set the rate to 1200, and a '2' would set the rate to 9600.

### Run `catman` to Access Online Manual

After installing your system, you need to run `catman -w` while in the `/usr/man` directory. This creates the `whatis` database. The `whatis` database must exist for you to be able to use either the `man -k` command (which prints out one-line summaries from the `whatis` database) or the `whatis` command (which describes what a command is).

Note that it usually takes about two hours for the `whatis` database to be generated.

### Hooking Up a Serial ASCII Printer

#### Page 160

In the *Editing the `printcap` File* section, add the following sentence to the end of the bulleted description of `lp`:

This file should be owned by `daemon`, belong to the `daemon` group, and have permissions 660.

#### Page 161

In the *Other File System Modifications* section under the first bulleted item, the first sentence should read as follows:

Check to make sure the proper permissions and ownerships exist on the files `/usr/lib/lpd`, `/usr/ucb/lpr`, `/dev/ttya`, and on the directory `/usr/spool/lpd`.

Add the following line to the example:

```
ls -lg /dev/ttya
crw-rw---- 1 daemon daemon 12, 0 Oct 21 11:57 /dev/ttya
```

### Hooking Up a Printer to a VPC-2200 Multibus Board

#### Page 165

In the section *Editing the `/etc/printcap` File*, add the following sentence to the end of the bulleted item on `lp`:

This file should be owned by `daemon`, belong to the `daemon` group, and have permissions 660.

In the *Other File System Modifications* section under the first bulleted item, the first sentence should read as follows:

Check to make sure the proper permissions and ownerships exist on the files `/usr/lib/lpd`, `/usr/ucb/lpr`, `/dev/vpc0`, `/dev/lp0`, and on the directory `/usr/spool/lpd`.

Add the following lines to the example:

```
ls -lg /dev/vpc0 /dev/lp0
crw-rw---- 1 daemon daemon 28, 1 Oct 21 11:57 /dev/lp0
crw-rw---- 1 daemon daemon 28, 0 Oct 21 11:57 /dev/vpc0
```

Page 164

Add the following sentences to the section *Using MAKEDEV to Create Special Files*:

`/dev/vpc0` is for the Versatec printer/plotter interface of the Systech VPC-2200 board. `/dev/lp0` is for the Centronics/Dataproducts interface of the same board. Once you have created the `/dev` entries, change the permissions with the following commands:

```
chown daemon /dev/vp0 /dev/vpc0 /dev/lp0
chgrp daemon /dev/vp0 /dev/vpc0 /dev/lp0
chmod 660 /dev/vp0 /dev/vpc0 /dev/lp0
```

## Major Changes in Disk and Directory Layout in Release 3.0 and Later Releases

The changes in disk and directory layout in Release 3.0 are not described in the *System Administration for the Sun Workstation* manual for Release 3.0. There is no `/usr2` partition in Release 3.0; instead there is `/usr/server_name/hostname`. This is important because if you put user directories in `/usr2`, you will end up with full root file systems almost immediately.

The changes in disk and directory layout are described on Page 4 of the *Release 3.0 Change Notes*.

## Preparing a Previously-Used Client Partition

Page 64

Under the section *Preparing a Previously-Used Client Partition*, users are referred to *Installing UNIX on the Sun Workstation* manual for instructions on how to prepare a client partition “by hand.” There are no such instructions in the *Installing UNIX on the Sun Workstation* manual.

To prepare a client partition “by hand,” refer to Appendix G of the *Installing UNIX on the Sun Workstation* manual for a listing of the contents of the distribution tape. On the first distribution tape is a raw client image in `tar` format, so it is no longer necessary to dump a raw partition to tape when installing your system; instead you can just `tar` off the client image and use that for your client partition.

Then refer to Appendix A in the *Installing UNIX on the Sun Workstation* manual for a list of files affected on a client. These are the files that the *Setup* program would have created for you, so make sure that these files exist and alter them accordingly for the new client. Then install any site-specific files. Then finish setting up the new client partition as you have other clients on your network.

### dump Cannot Be Used to Dump an NFS-mounted File System

Note that you cannot use `dump` to dump an NFS-mounted file system. You can only dump hard partitions (`/dev/rxy0a` through `/rxy0h`) or soft partitions (`/dev/rnd10`, `/dev/rnd11`, and so forth.) For example, if you wanted to dump the home directories in the `/usr` partition, you could type the following:

```
/etc/dump 9uf /dev/nrmt0 /dev/rxy0d
```

to do a level 9 dump of the home directories in `/usr`. If you wanted to dump the clients' root partitions you might type:

```
/etc/dump 9uf /dev/nrmt0 /dev/rnd10
/etc/dump 9uf /dev/nrmt0 /dev/rnd11
/etc/dump 9uf /dev/nrmt0 /dev/rnd12
```

This would dump the clients' root (soft) partitions, which are located on the hard partition `/dev/xy0c`. You could also dump a file system onto a remote system's tape drive from a machine where you are logged in as superuser by using the command:

```
/etc/rdump 9uf remote_machine_name:/dev/rmt0 /dev/xy0d
```

This would dump the `d` hard partition where the home directories are located. Use the `rdump` command when doing remote dumps over the Ethernet. You cannot, however, dump an NFS-mounted file system.

### How to Use `yppasswd` Versus `passwd`

Use the `passwd` command to change (or create) your local `/etc/passwd` file. Use `yppasswd` to change your password in the `yp` database. The `yppasswd` command does not affect your local password file, or any of the password files on any remote machines on which you have accounts; it only affects the password in the `yp` database. Thus the password in the `yp` database may be different from the one on your own machine.

Note that only the owner of the name or the superuser can change a password.

`yppasswd`, the yellow pages password daemon, must be running on your `yp` server for the new password to take effect.

For more information, refer to the `passwd(8)`, `yppasswd(1)`, and `passwd(5)` manual pages.

### Print Filter Hook

Page 167

In the *Adding Hardware to your System* chapter, the subsection entitled *Output Filter Specifications* describes the interface for print filters. The line printer daemon (`lpd`) uses a hook for switching from one filter to another, and introduces special characters that the filter can detect and thus switch to the other filter. If you are using more than one print filter, you need to use the following print filter hook:

```

/*
 * This example filter demonstrates how to handle the ^Y^A sequence lpd
 * sends for switching from this 'of' filter to another. This action is
 * required when using 'of' filters only for filtering or printing banner
 * pages and this 'of' filter is stopped to use another filter, e.g., 'if'.
 *
 * This filter highlights the lpd-generated banner page by overstriking and
 * is intended to work with an 'if' filter entry in the printcap file.
 *
 * The banner text can be processed in other ways, e.g., parsing the short
 * banner line (see 'sb' in printcap(5)) and incorporating this information
 * in your own banner page.
 */

#include <stdio.h>
#include <signal.h>

main(argc, argv)
int argc;
char *argv[];
{
 register char *cp;
 register int ch;

 /* 'of' filters are only passed width and length arguments from lpd */
 while (--argc) {
 if (*(cp = ++argv) == '-') {
 switch (cp[1]) {
 case 'w':
 /* this filter does not use width argument */
 break;

 case 'l':
 /* this filter does not use length argument */
 break;
 }
 }
 }

 while ((ch = getchar()) != EOF) {
 switch (ch) {
 case ' 31':
 /*
 * lpd needs to use a different filter to
 * print data so stop what we are doing and
 * wait for lpd to restart us.
 */
 if ((ch = getchar()) == ' 01') {
 (void) fflush(stdout);
 (void) kill(getpid(), SIGSTOP);
 break;
 } else {
 (void) ungetc(ch, stdin);

```

```

 ch=' 31';
 }

 default:
 putchar(ch);
 putchar('\\b');
 putchar(ch);
 break;
 }
}
exit(0);
}

```

### **fstab Entry for a Sun-2/3Com Diskless Client Mounting from a Sun-3 Server**

If a diskless Sun-2/3-Com client NFS-mounts a file system from a Sun-3 server, extra entries need to be added to the `/etc/fstab` and `/etc/nd.local` files to make it work successfully. Otherwise, the Sun-3 server pumps out packets so fast that the Sun-2 cannot keep up. The NFS buffer size and `nd maxpack` parameter have to be limited, or else you will get too many re-transmissions causing *server not responding* messages.

If the Sun-2/3-Com mounts an NFS file system from a Sun-3, you need to have the Sun-2/3-Com specify "rsize=2048" in `/etc/fstab` for this mount. This makes sure NFS reads by the Sun-2/3-Com NFS client from the Sun-3 server are reduced to a size that the 3-Com can handle.

If a Sun-3 mounts an NFS file system from a Sun-2/3-Com, you need to have the Sun-3 specify "wsize=2048" for this mount. This makes sure that NFS writes by the Sun-3 client to the Sun-2/3-Com NFS server are reduced to a size that the 3-Com can handle.

If you have a Sun-2/3-Com diskless client being served by a Sun-3, you must similarly adjust the `nd` software to prevent this overrun of the 3-Com. This is done by adding a "2" at the end of each Sun-2/3-Com client line in `/etc/nd.local` file on the Sun-3 server.

The reverse situation, that is, having a Sun-3 diskless client being served by a Sun-2/3Com server, cannot be compensated for by changing `/etc/nd.local` and is not recommended.

### **How to Add a New yp Slave Server Not in the Original Set**

Page 45

The following section replaces the section currently on Page 45 of the *System Administration for the Sun Workstation* manual:

To add a new `yp` slave server, start by modifying some maps on the `yp` master server. If the new server is a host that has not been a `yp` server before, you must add the host's name to the `ypservers` map in the default domain. To add a server named `ypslave` to `domain_name`, do the following:



```

ypmaster# cd /usr/etc/yp/domain_name
ypmaster# ../makedbm -u ypservers > /tmp/temp_file
ypmaster# vi /tmp/temp_file
 <add the new hostname to the temporary file>
ypmaster# ../makedbm /tmp/temp_file ypservers

```

Running the `makedbm` command with the `-u` option undoes `ypservers`; that is, it converts it from `dbm` format into text file format temporarily so that you can add the new hostname to the temporary file `temp_file`. Then running the `makedbm` command with `temp_file` as the input file and `ypservers` as the output file converts `ypservers` back into `dbm` format.

You can then set up the new yp slave server's databases by copying the databases from the yp master server `ypmaster`. To do this, remote log in to the new yp slave server, and run the `ypinit(8)` command as follows:

```

ypslave# cd /usr/etc/yp
ypslave# ypinit -s ypmaster

```

To verify that the `ypservers` file is correct (because there is no ASCII file for the `ypservers` map), do the following:

```

ypslave# cd /usr/etc/yp/domain_name
ypslave# ../makedbm -u ypservers

```

**Note:** If a host name is not in the `ypservers` map it will not be notified when updates are made to the yp database.

Then complete the steps described in the section *How To Set Up A Slave yp Server* above.

#### uucp L.cmds File

Each line of the `L.cmds` file contains the name of a command. Avoid leaving any white space that is not part of the name of a command, either at the beginning or end of the line. If you leave white space in the `L.cmds` file, `uucp` incorrectly parses the file and sends out the error message

```

PERMISSION DENIED to LOGFILE

```

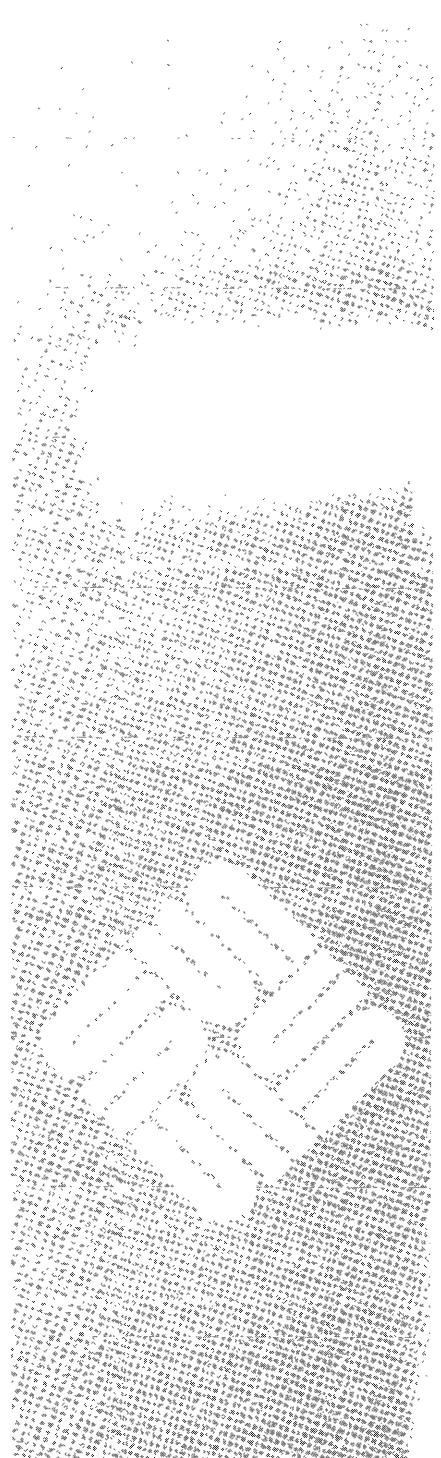


# A

---

## Reference Manual Pages

Reference Manual Pages ..... 219





## Reference Manual Pages

A number of reference manual pages have been reprinted to include corrections or enhancements made in Sun Release 3.4 and 3.5. The reprints include the correct facing pages, so that they can replace existing pages, or be inserted at the proper page locations, in the Sun Release 3.2 reference manuals.

The following manual pages have been reprinted, and are included as an attachment to this release manual.

*NOTE* `rasfilter8tol(1G)` is moved to `rasfilter8tol(1)`.

Table A-1 *Reprinted Reference Manual Pages*

| Reprinted Pages for Insertion into Reference Manuals |                                                                                                                                       |
|------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Manual Page                                          | Description                                                                                                                           |
| <i>Commands Reference Manual</i>                     |                                                                                                                                       |
| <i>Section 1</i>                                     |                                                                                                                                       |
| <code>clear_colormap(1)</code>                       | Documents enhancements to Sun Release 3.4.                                                                                            |
| <code>cmdtool(1)</code>                              | Documents enhancements to Sun Release 3.5.                                                                                            |
| <code>cp(1)</code>                                   | Now warns that the <code>-r</code> option does not preserve symbolic links.                                                           |
| <code>defaultsedit(1)</code>                         | Documents new environment variable, <code>DEFAULTS_FILE</code> .                                                                      |
| <code>ex(1)</code>                                   | Documents <code>EXINIT</code> environment variable and <code>.exrc</code> files.                                                      |
| <code>grep(1V)</code>                                | Now notes that <code>-n</code> behavior is position-dependent when concatenated with other options.                                   |
| <code>id(1V)</code>                                  | Now classified correctly as System V optional software.                                                                               |
| <code>indent(1)</code>                               | Documents enhancements for Sun Release 3.4.                                                                                           |
| <code>login(1)</code>                                | Manual page was missing from printed manual, omitted mention of <code>/etc/securetty</code> root-login terminal access file.          |
| <code>lpr(1)</code>                                  | Documents <code>PRINTER</code> environment variable and the default-printer <code>printcap</code> entry <code>lpr</code> supplies.    |
| <code>on(1C)</code>                                  | Added a <code>BUGS</code> entry: a <code>CTRL-Z</code> in an NFS-mounted directory hangs the window from which the signal originates. |
| <code>overview(1)</code>                             | Corrected erroneous view-surface names.                                                                                               |
| <code>pack(1)</code>                                 | Corrected vertical page alignment.                                                                                                    |
| <code>ps(1)</code>                                   | Documents the fact that the reported size of process segments depends on options selected.                                            |
| <code>rasfilter8tol(1)</code>                        | Documents enhancements for Sun Release 3.4.                                                                                           |
| <code>rastrepl(1)</code>                             | Documents enhancements for Sun Release 3.4.                                                                                           |

Table A-1 Reprinted Reference Manual Pages—Continued

| Reprinted Pages for Insertion into Reference Manuals |                                                                                                                                                                                                                                                                                                                 |
|------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Manual Page                                          | Description                                                                                                                                                                                                                                                                                                     |
| rcp(1C)                                              | Documents <code>-p</code> (preserve access and mod. times) option.                                                                                                                                                                                                                                              |
| rdist(1)                                             | Dollar-sign characters (\$) were misprinted as percent signs (%).                                                                                                                                                                                                                                               |
| screendump(1)                                        | Documents enhancements for Sun Release 3.4.                                                                                                                                                                                                                                                                     |
| screenload(1)                                        | Documents enhancements for Sun Release 3.4.                                                                                                                                                                                                                                                                     |
| sh(1)                                                | Dollar-sign characters (\$) were misprinted as percent signs (%).                                                                                                                                                                                                                                               |
| suntools(1)                                          | Documents new environment variable, <code>DEFAULTS_FILE</code> .                                                                                                                                                                                                                                                |
| symoder(1)                                           | Error in synopsis.                                                                                                                                                                                                                                                                                              |
| <i>Section 6</i>                                     |                                                                                                                                                                                                                                                                                                                 |
| graphics_demos(6)                                    | Documents enhancements to <code>goban</code> for Sun Release 3.4.                                                                                                                                                                                                                                               |
| <i>Section 7</i>                                     |                                                                                                                                                                                                                                                                                                                 |
| ascii(7)                                             | The octal chart was mislabeled as decimal.                                                                                                                                                                                                                                                                      |
| man(7)                                               | Documents enhancements for Sun Release 3.5.                                                                                                                                                                                                                                                                     |
| <i>Section 8</i>                                     |                                                                                                                                                                                                                                                                                                                 |
| catman(8)                                            | Documents harmless but disturbing warning messages.                                                                                                                                                                                                                                                             |
| dump(8)                                              | Now notes that multi-file tapes are not supported.                                                                                                                                                                                                                                                              |
| edquota(8)                                           | Documents requirement for a <code>/etc/quotas</code> file in order to activate quotas.                                                                                                                                                                                                                          |
| fingerd(8C)                                          | Reference Page missing from Release 3.2.                                                                                                                                                                                                                                                                        |
| ifconfig(8C)                                         | Documents enhancements for Sun Release 3.5.                                                                                                                                                                                                                                                                     |
| kadb(8S)                                             | Additions have been made to text.                                                                                                                                                                                                                                                                               |
| monitor(8S)                                          | Registers <code>CA</code> , <code>CC</code> and <code>CX</code> added to the <code>R</code> command.                                                                                                                                                                                                            |
| mount(8)                                             | Incorrectly indicated that the default <code>retry</code> value for <code>nfs</code> file systems was 1, when it is 10000.                                                                                                                                                                                      |
| setup(8)                                             | Formatting error, index entry was embedded in text.                                                                                                                                                                                                                                                             |
| sticky(8)                                            | Documents Sun Release 3.4 enhancement, sticky directories. When sticky bit set for a directory, files can only be deleted by their respective owners.                                                                                                                                                           |
| <i>UNIX Interface Reference Manual</i>               |                                                                                                                                                                                                                                                                                                                 |
| <i>Section 2</i>                                     |                                                                                                                                                                                                                                                                                                                 |
| chmod(2)                                             | Documents sticky bit for directories.                                                                                                                                                                                                                                                                           |
| getrlimit(2)                                         | Now returns <code>EINVAL</code> for invalid parameters.                                                                                                                                                                                                                                                         |
| gettimeofday(2)                                      | Declared <code>timeval</code> structure incorrectly, was unclear that <code>tz_dsttime</code> indicates that daylight savings time <i>could</i> apply (not whether it is currently in effect), omitted mention that nonzero return values are defined in the header file <code>/usr/include/sys/time.h</code> . |
| read(2V)                                             | Now notes that you must use <code>readdir</code> or <code>getdirentries</code> over NFS.                                                                                                                                                                                                                        |
| shmop(2)                                             | Spurious "EINVAL" appears in wrong paragraph.                                                                                                                                                                                                                                                                   |
| socket(2)                                            | Omitted <code>EPROTOTYPE</code> error return.                                                                                                                                                                                                                                                                   |
| <i>Section 3</i>                                     |                                                                                                                                                                                                                                                                                                                 |
| intro(3)                                             | Documents enhancements for Sun Release 3.5.                                                                                                                                                                                                                                                                     |
| directory(3)                                         | Documents enhancement for Sun Release 3.5.                                                                                                                                                                                                                                                                      |
| frexp(3)                                             | Documents 4.3 BSD enhancements made in Sun Release 3.2.                                                                                                                                                                                                                                                         |
| monitor(3)                                           | Documents enhancements for Sun Release 3.5.                                                                                                                                                                                                                                                                     |

Table A-1 *Reprinted Reference Manual Pages— Continued*

| Reprinted Pages for Insertion into Reference Manuals |                                                                                                            |
|------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| Manual Page                                          | Description                                                                                                |
| vlimit(3C)                                           | Documents enhancements for Sun Release 3.5.                                                                |
| fopen(3S)                                            | Notes NULL returns, other corrections.                                                                     |
| scanf(3S)                                            | Changed to reflect corrected handling of errors.                                                           |
| scanf(3V)                                            | Changed to better reflect the System V behavior.                                                           |
| intro(3X)                                            | Documents enhancements for Sun Release 3.5.                                                                |
| <i>Section 4</i>                                     |                                                                                                            |
| bwone(4S)                                            | Documents enhancements to the Sun Release 3.4 device driver.                                               |
| bwtwo(4S)                                            | Documents enhancements to the Sun Release 3.4 device driver.                                               |
| cgfour(4S)                                           | Documents enhancements to the Sun Release 3.4 device driver.                                               |
| fbio(4S)                                             | Documents enhancements for Sun Release 3.4.                                                                |
| gpone(4S)                                            | Documents enhancements for Sun Release 3.5.                                                                |
| mcp(4S)                                              | Documents enhancements for Sun Release 3.5.                                                                |
| xy(4S)                                               | Documents enhancements for Sun Release 3.5.                                                                |
| <i>Section 5</i>                                     |                                                                                                            |
| aliases(5)                                           | The <code>.forward</code> file must be readable by all, home directory searchable by all, to forward mail. |

The on-line sources to the following manual pages have been updated to correct typographical and other minor errors.

Table A-2 *Corrected On-Line Reference Pages*

| Online Manual Pages with Minor Corrections |                                                                                                              |
|--------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| Manual Page                                | Description                                                                                                  |
| ar(1)                                      | Undocumented member-name length restriction, 15 characters; longer names are truncated in the library entry. |
| biff(1)                                    | Must own terminal to use, but the super-user owns terminal windows; workaround noted.                        |
| binmail(1)                                 | -i option is invalid.                                                                                        |
| bstring(3)                                 | ffs returns 0 for passed value of 0.                                                                         |
| creat(2)                                   | Missing reference to open(2), which supercedes creat.                                                        |
| csh_builtins(1)                            | Bad ".so" reference.                                                                                         |
| cuserid(3)                                 | Typographical error.                                                                                         |
| etherfind(8C)                              | Spurious reference to { and } removed.                                                                       |
| find(1)                                    | Typographical error.                                                                                         |
| getgrent(3)                                | Formatting error.                                                                                            |
| getrusage(2)                               | Formatting error                                                                                             |
| indxbib(1)                                 | Does not allow pathnames in <i>database</i> argument, only files in current directory.                       |
| mntent(5)                                  | Spelling error.                                                                                              |
| monitor(8s)                                | Minor change to register names in the R command.                                                             |
| more(1)                                    | Typographical error.                                                                                         |
| putc(3S)                                   | Minor clarifications, returns integer written.                                                               |

Table A-2 *Corrected On-Line Reference Pages—Continued*

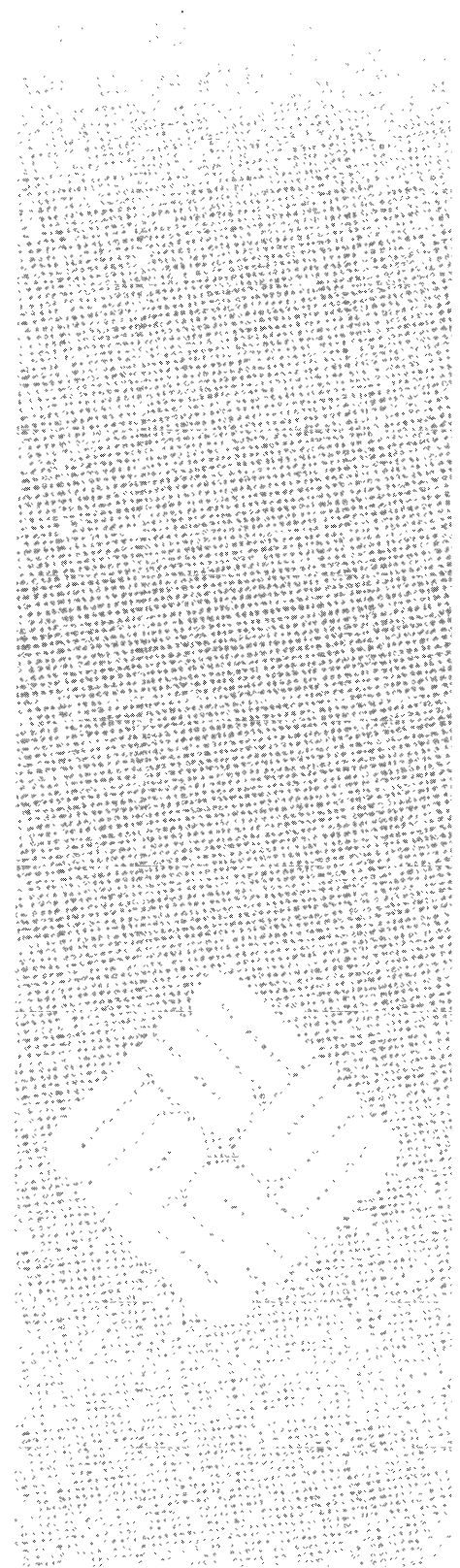
| Online Manual Pages with Minor Corrections |                                                                                                           |
|--------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| Manual Page                                | Description                                                                                               |
| printf(3)                                  | Formatting error.                                                                                         |
| readdir(3)                                 | Formatting error.                                                                                         |
| rm(1)                                      | Fails to note that <code>rm</code> checks real UID before removing a directory.                           |
| select(2)                                  | The structure <code>timeval</code> should be declared as type <code>*timeout</code> .                     |
| swapon(8)                                  | Typographical error.                                                                                      |
| textedit(1)                                | FILES entry <code>/tmp/Text*</code> should be <code>/tmp/EtHost*</code> .                                 |
| vi(1)                                      | Did not document <code>EXINIT</code> environment variable. (See attached reprint of <code>ex(1)</code> .) |
| vacation(1)                                | Must use pathname <code>/usr/ucb/vacation</code> in the <code>.forward</code> file                        |
| vpc(4S)                                    | FILES entry <code>/dev/vp0</code> should be <code>/dev/vpc0</code> .                                      |
| wall(1)                                    | <i>file</i> argument not documented, names a file containing text of broadcast message.                   |
| ypfiles(5)                                 | Minor clarification to description of YP database.                                                        |



# B

## Installing UNIX on Tapeless Workstations

|                                                                            |     |
|----------------------------------------------------------------------------|-----|
| Installing UNIX on Tapeless Workstations .....                             | 225 |
| B.1. Overview of the Installation Procedure .....                          | 225 |
| B.2. Configuring the Remote Host as a Network Disk .....                   | 226 |
| B.3. Configuring the Remote Host as a Network Disk Server .....            | 226 |
| B.4. Configuring the Remote Host .....                                     | 227 |
| B.5. Determining Network Information .....                                 | 230 |
| B.6. Loading the Mini UNIX System on the Remote Host .....                 | 231 |
| B.7. Using <i>diag</i> to Format and Label the Target Machine's Disk ..... | 232 |
| B.8. Loading the Mini UNIX System .....                                    | 232 |
| B.9. Booting the Mini UNIX System .....                                    | 233 |





# B

## Installing UNIX on Tapeless Workstations

This chapter describes how to install UNIX on a standalone workstation or a file server which does not have a resident tape drive. To do this, you use the tape drive on another, fully installed machine. This machine is your tape server — which we call your **remote host** — and perform the installation on your **target machine** across the Ethernet.

### NOTE

1. The remote host **MUST** be running Release 3.5. The remote host machine must either be configured as a server or a standalone machine; it may not be a client. If the machine is configured as a standalone, you must make it 'look' like a network disk server for remote installation. To do this, there is one primary requirement: the machine's kernel must have been generated from a system configuration file which includes the device description lines:

```
pseudo-device ether
pseudo-device nd
```

2. The remote host must have at least 5 megabyte of free disk space. If the remote host is a file server, this space must be in the */pub* partition and if the remote host is a standalone machine, this space must be in the */usr* partition.

### B.1. Overview of the Installation Procedure

A "remote installation" is very similar to standard UNIX installation; steps are:

1. Complete UNIX installation on your remote host. See chapters 1 through 7 of this document for procedures. As noted above, the remote host must either be configured as a server or a standalone machine, and must have the "pseudo-device nd" and "pseudo-device ether" lines included in its system configuration file.
2. If your remote host is configured as a standalone system, you must enable it as a server and turn its */usr* file system into a public network disk. If your remote host is configured as a server, this step is unnecessary.
3. Make the ethernet addresses and Internet addresses for both the remote host and the target machine available to the networking software.
4. Start the reverse-ARP daemon.

5. Edit */rhosts*.
6. Determine the unit number of the public partition (*pub#*).
7. Link to the proper *tftpboot* file.
8. Determine the network information necessary for installation.
9. Load the mini UNIX system onto the public partition of your remote host from the distribution tape.
10. Boot *diag* over the network; run *diag* to format (if necessary) and label your disk.
11. Boot the standalone *copy* program over the network. Run *copy* to copy a mini UNIX system over the network into the swap area on your disk.
12. Boot the mini UNIX system.
13. Run the *Setup* program to install UNIX from the tape and initialize the network files.
14. Boot the full UNIX system.

## B.2. Configuring the Remote Host as a Network Disk

*NOTE* Follow the procedures in this next section only if you are using a standalone machine rather than a server as your remote host. You can proceed to the next section if your remote host is already configured as a server.

## B.3. Configuring the Remote Host as a Network Disk Server

If the remote host machine is not configured as a network disk server, you need to turn the host's */usr* file system into a public network disk so that your target machine can access the files necessary for remote booting. Do the following on the remote host:

1. Edit the */etc/nd.local* file and add the following two lines at the end of the file.

```
user 0 0 /dev/usr_partition 0 -1 -1
son
```

*usr\_partition* refers to the disk controller abbreviation, unit number and partition letter of the */usr* partition. The disk controller abbreviations are *xy* for a Xylogics disk controller, *sd* for a SCSI disk controller, and *ip* for an Interphase disk controller. On most standalone machines the */usr* partition is on disk unit zero and on partition *g*. For example, for a standalone machine with a Xylogics disk controller the commands are:

```
user 0 0 /dev/xy0g 0 -1 -1
son
```

2. Enable the network disk server by typing the following (the argument to **MAKEDEV** below is three alpha characters "ndl" followed by 'zero')

numeric):

```
host# cd /dev
host# MAKEDEV nd10
host# /etc/nd - < /etc/nd.local
```

The device description lines mentioned above must be included in the host machine's system configuration file for this to work.

## B.4. Configuring the Remote Host

Now you must configure the remote host to allow it to communicate with the target machine.

1. You must insure that the ethernet addresses and Internet addresses for both the target machine and the remote host are known to the networking software.

If you are using the yellow pages, this information must be added to the yellow pages databases on the yellow pages master server.

If you are not using the yellow pages, the Internet addresses are added to the file */etc/hosts* and the ethernet addresses are added to the file */etc/ethers*.

For example, if you are not using the yellow pages and you have the following configurations:

|                | Host name  | Internet Addr | Ethernet Addr |
|----------------|------------|---------------|---------------|
| Remote host    | tapeserver | 192.9.200.100 | 8:0:20:0:0:1  |
| Target Machine | tapeless   | 192.9.200.101 | 8:0:20:0:0:2  |

make sure */etc/hosts* contains these lines:

```
192.9.200.100 tapeserver
192.9.200.101 tapeless
```

make sure */etc/ethers* contains these lines:

```
8:0:20:0:0:1 tapeserver
8:0:20:0:0:2 tapeless
```

2. Start the reverse-ARP daemon. Its arguments are the name of the ethernet interface and the host name. If you have a Sun ethernet board, the ethernet interface is *ie0*. If you have a 3COM ethernet board, the ethernet interface is *ec0*. For example, if the remote host's name is *tapeserver* and it has a Sun ethernet board the command is

```
host# /usr/etc/rarpd ie0 tapeserver
```

3. Add the target machine's host name to the file */.rhosts* file. This enables the target machine to execute remote commands on the remote host machine.
4. You must determine the unit number of the public partition that is being used for remote booting. This number will be used in the next step and will be referred to as *pub#*.

If the remote host is a standalone machine, the *pub#* is unit zero.

If the remote host is a file server, then you must look in the */etc/nd.local* file to find the unit number. If the file server is serving only one architecture, look for a line beginning with **user 0** and the unit number will be the number immediately following **user 0**. In the following example the unit number is one

```
user 0 1 /dev/xy0f 0 -1 -1
```

If the file server is serving multiple architectures, then there will be more than one line beginning with **user 0**. It is best to use the public partition that serves the same architecture as the target machine. Use */etc/mount* to determine the device name associated with each */pub.\** file system and then look for a line beginning with **user 0** that is on the appropriate device name. The unit number will be the number immediately following **user 0**. For example if the output of the */etc/mount* command is

```
host# /etc/mount
/dev/xy0e on /pub.MC68010 type 4.2 (rw)
/dev/xy0f on /pub.MC68020 type 4.2 (rw)
```

and the */etc/nd.local* file contains the following lines

```
user 0 0 /dev/xy0e 0 -1 -1
user 0 1 /dev/xy0f 0 -1 -1
```

then the unit number for the MC68010 (Sun2) public partition is 0 and the unit number MC68020 (Sun3) public partition is 1.

5. A link to the proper tftpbooting file in the directory */tftpboot* must be created. If the architectures of the target machine and the remote host differ, the remote host will not have the proper tftpbooting files. For example, if the remote host is a Sun2 and the target machine is a Sun3, then the remote host will have tftpbooting files for Sun2s only. In this case, you must obtain Sun3 tftpbooting files from another Sun3 or from the Sun3 installation tapes. If there is another Sun3 on the network that is running Release 3.0, issue the following command:

```
host# rcp machine:/tftpboot/ndboot.* /tftpboot
```

If there is not another Sun3, then the files must be obtained from the installation tapes (see *Appendix G*). Position the tape to the *pub files* and use the *tar* command to extract the */tftpboot* directory. Remember that to position a tape to file *n*, you must forward space the tape *n-1* files. For example, if the *pub files* is the eighth tape file, issue the following commands:

```
host# cd /
host# mt -f /dev/nrtape0 rew
host# mt -f /dev/nrtape0 fsf 7
host# tar xfvbp /dev/nrtape0 126 ./tftpboot
```

Now the link must be created. The name of the link is the target machine's Internet address converted to hexadecimal and using all capital letters. *adb(1)* can be used to convert the Internet address to hexadecimal. For example, if the Internet address is 192.9.200.100:

```

host% adb
0t192=X
 c0
0t9=X
 9
0t200=X
 c8
0t100=X
 64
^D
host%

```

Therefore, the file name for Internet address 192.9.200.100 is *C009C864*. The link must be to a file that is appropriate for the architecture of the target machine. If the target machine is a Sun3, the link should be made to *ndboot.sun3.pub0* or *ndboot.sun3.pub1* and for a Sun2, the link should be made to *ndboot.sun2.pub0* or *ndboot.sun2.pub1*. If *pub#* is zero, use the file ending with *pub0* and if *pub#* is one, use the file ending with *pub1*. For example, if the target machine is a Sun3 and *pub#* is zero, the link command would be:

```

host# cd /tftpboot
host# ln -s ndboot.sun3.pub0 C009C864

```

6. The files required for remote booting must be installed on the public partition that is being used for booting. There are several scenarios here. The remote host may be either a file server or a standalone machine and the architectures of the target machine and remote host may or may not be the same. In the following sequences of commands */pubarch* refers to the directory name of the public partition that is being used for booting, (typically */pub.MC68010* or */pub.MC68020*). *machine* refers to another machine of the same architecture as the target machine. If there is not another machine on the network with the same architecture as the target machine, the files must be obtained from the installation tapes. The previous subsection explains how to obtain files from the installation tapes. *pub\_partition* refers to the disk controller abbreviation, disk unit number and partition letter of the partition that is being used for booting (for example, *xy0d*). */bootfile* refers to the file */pub.boot* if the target machine is a MC68010 (Sun2) and the file */boot* if the target machine is a MC68020 (Sun3).

If the remote host is a file server and the architecture being served by the public partition is the same as the target machine's architecture, then all the files are in place and you may skip the rest of this subsection.

If the remote host is a file server and the architectures differ, then the files are in place but they are for the wrong architecture. The files must be moved and replaced by the correct ones. The correct files can be obtained from another machine of the appropriate architecture on the network; or they may be obtained from the installation tapes. The example shows how to obtain them from another machine:

```

host# cd /pubarch
host# mv stand stand.orig
host# mv boot boot.orig
host# mkdir stand
host# rcp machine:/stand/diag stand
host# rcp machine:/stand/copy stand
host# rcp machine:/bootfile /pubarch/boot
host# rcp machine:/usr/mdec/bootnd /tmp
host# cd /usr/mdec
host# installboot /tmp/bootnd /dev/pub_partition
host# sync
host#

```

If the remote host is a standalone machine and its architecture is the same as the target machine's architecture, issue the following commands.

```

host# cp /stand/diag /usr/stand
host# cp /stand/copy /usr/stand
host# ln -s /usr /pub
host# cp /bootfile /pub/boot
host# cd /usr/mdec
host# installboot bootnd /dev/pub_partition
host# sync
host#

```

If the remote host is a standalone machine and its architecture differs from the target machine's architecture, you must obtain files that are appropriate for the target machine. The correct files can be obtained from another machine of the appropriate architecture on the network; or they may be obtained from the installation tapes. The example shows how to obtain them from another machine:

```

host# cd /usr
host# mv stand stand.orig
host# mkdir stand
host# rcp machine:/stand/diag stand
host# rcp machine:/stand/copy stand
host# ln -s /usr /pub
host# rcp machine:/bootfile /pub/boot
host# rcp machine:/usr/mdec/bootnd /tmp
host# cd /usr/mdec
host# installboot /tmp/bootnd /dev/pub_partition
host# sync
host#

```

## B.5. Determining Network Information

For later phases of remote installation, you need to know the remote host's host number and the target's hardware Ethernet address. You must obtain this information now.

1. To determine the remote host's hexadecimal host number, find its entry in its own */etc/hosts* file. As you remember, entries consist of a machine's full Internet address (network number followed by host number) and name, for



example:

```
192.9.200.48 augustus
192.9.200.50 julius
192.9.200.52 claudius
```

Here, julius' Internet address is 192.9.200.50; its network number is 192.9.200, and its host number (in decimal) is 50. Be sure to remember the Internet address because you will need it during *Setup*. If you are using the yellow pages, you can find the Internet address with `ypmatch julius hosts`.

Since host numbers in */etc/hosts* are in decimal, and you need the remote host's host number in hexadecimal, you will need to convert. You can use *adb* for this if you wish:

```
host% adb
0thost_number_in_decimal = x
host_number_in_hex
^D
host%
```

2. To obtain the hardware Ethernet address of the target, power up the target workstation. You will see the PROM Monitor's power-up banner — which includes the hardware Ethernet address — and then the machine will start to auto boot. Stop the auto boot immediately by typing the appropriate abort sequence (if you don't know the abort sequence for the target machine, please see Chapter 2, *Abort Procedure*):

```
Self Test completed successfully.

Sun Workstation, [model type], [keyboard type]
ROM Rev N, some_number_MBytes memory installed
Serial #some_number, Ethernet address xx:xx:xx:xx:xx:xx

Auto-boot in progress . . .

[abort by typing the appropriate abort sequence here]

Abort at some address
>
```

Copy down the displayed Ethernet address.

## B.6. Loading the Mini UNIX System on the Remote Host

Next, you copy the mini UNIX file system from the distribution tape to the remote host's public partition.

1. Load distribution tape 1. If you have any questions about loading the tape, see the previous chapter, *Loading the Bootstrap Program*.
2. Type the following on the remote host. Remember to replace *tape* with *mt* for the nine-track tape, *ar* for the Archive quarter-inch tape, or *st* for the SCSI tape controller. Replace *pub* with **pub** if the remote host is a standalone machine and either **pub.MC68010** or **pub.MC68020** if the remote host is a file server. If you are using a nine-track half-inch tape, use **20** for *blk\_factor* ('bs=20b'); use **126** for a 1/4-inch tapes ('bs=126b'):

```

host# mt -f /dev/nrtape 0 rew
host# mt -f /dev/nrtape 0 fsf 5
host# dd if=/dev/nrtape 0 of=/pub/minifs bs=blk_factor b
host# sync
host#

```

This takes about seven minutes using a 1/2" tape, and fifteen minutes using a 1/4" cartridge.

### B.7. Using *diag* to Format and Label the Target Machine's Disk

Now, you start to work on your target machine, and install UNIX from the remote host. The first step is to format your target's disk(s) with the *diag* utility.

Procedures for using *diag* in this remote installation are identical to those for a standard UNIX installation with one exception: during a standard installation you boot *diag* from the distribution tape; here, you boot from your remote host.

Boot *diag* from the remote host with the following boot command. Remember to replace *e\_interface* with the appropriate abbreviation for your Ethernet controller (ec for the 3COM Ethernet Controller, ie for the Sun-2 Ethernet Controller, or le for the Sun3/50); replace *host\_number* with the remote host's hexadecimal host number (obtained earlier). If you have more than one Ethernet Controller Board in your system, and you are booting from the second, third, etc., replace the first "0" in the command with the controller's address on the Multibus (in hex).

```
> b e_interface (0,host_number,0) stand/diag
```

When you type this, the monitor boots *diag* from the network disk server. When *diag* starts up, it displays a sign on message:

```

Version sccs version_number and date
Disk Initialization and Diagnosis

When asked if you are sure, respond with 'y' or 'Y'

```

### B.8. Loading the Mini UNIX System

When you're done formatting and labeling your disk(s), you're ready to load the mini UNIX system from the remote host to your disk. To do this, you use the standalone *copy* program which you boot from the remote host. Proceed as follows.

1. Boot the standalone *copy* program from the remote host with the following commands. Replace *e\_interface* with the proper device abbreviation for your Ethernet controller, and replace *host\_number* with the remote host's host number (in hex). The new *Setup* only works from the first Ethernet board. Also, if you are not booting from the first Ethernet Controller Board in your system, use the board's Multibus address (in hexadecimal) rather than "0" in the boot command:

```

> b e_interface (0,host_number,0)stand/copy
Boot: e_interface (0,host_number,0)stand/copy
Load: e_interface (0,host_number,0)boot

Boot: e_interface (0,host_number,0)stand/copy
[...messages displaying sizes of copy program...]

```

2. The *copy* program prompts you for the source (From:) and destination (To:) of the copy. When you respond to its queries, remember to substitute the correct device abbreviations for *e\_interface* and *disk*:

```
Standalone Copy
From: e_interface (0,host_number,0)minifs
To: disk (0,0,1)
```

**NOTE** *If trying to boot a Sun-3 machine from a Sun-2 server, the partition number has to be 1.*

Copying in the mini UNIX system takes about seven minutes using a half-inch tape, and about fifteen minutes using a quarter-inch cartridge. This process loads the mini UNIX file system into the swap area on the disk. It takes about six minutes to do the copy over the Net. When it completes, the *copy* program returns control to the monitor:

```
Copy completed - 524880 bytes
Standalone Copy
From:
>
```

Now abort your system.

## B.9. Booting the Mini UNIX System

The next step is to boot UNIX in single-user state and specify the location of its root file system.

1. First, bring in the main boot program:

```
> b e_interface (0,host_number,0)boot -a
```

2. Now you can tell the the bootstrap program to boot the mini UNIX system from your own disk. Because this boot is an unusual one, you must specify the *-a* (for ask me) option on the boot command, and also the *-s* (come up single user) option, as follows:

```
Boot: disk (0,0,1)vmunix -as
Size: 366592+61440+98828 bytes [numbers vary with system level]
Sun UNIX 4.2 Release 3.0 (GENERIC) #145: Mon Feb 17 20:35:1
Copyright (c) 1985 by Sun Microsystems, Inc.

[...about a dozen lines of configuration messages...]

root device?
```

3. As the mini UNIX system comes up, it displays some messages about the configuration of the system on which it is running, and finally queries you, asking for its root file system. The root file system at this stage is “*disk 0\**”, which has a special meaning to the mini UNIX system. Since this notation looks ambiguous, let me specify: if you have a Xylogics disk controller, your root device is *xy0\**; and if you have a SCSI disk controller, it is *sd0\** — the asterisk is part of the device name:

**NOTE** *If you have an Interphase controller, it is ip0\*.*

```
root device? disk0*
```

Depending upon your hardware, you may be asked to set the date at this point:

```
using number buffers containing number bytes of main memory
WARNING: no tod clock -- CHECK AND RESET THE DATE!
Single user boot - - fsck not done
```

If so, continue with the section *Setting the Date* in Chapter 2, and return to this point when you've finished that subsection; otherwise, continue with the next section.

At this point, proceed to chapters 3-5 of this manual to invoke the *Setup* program.

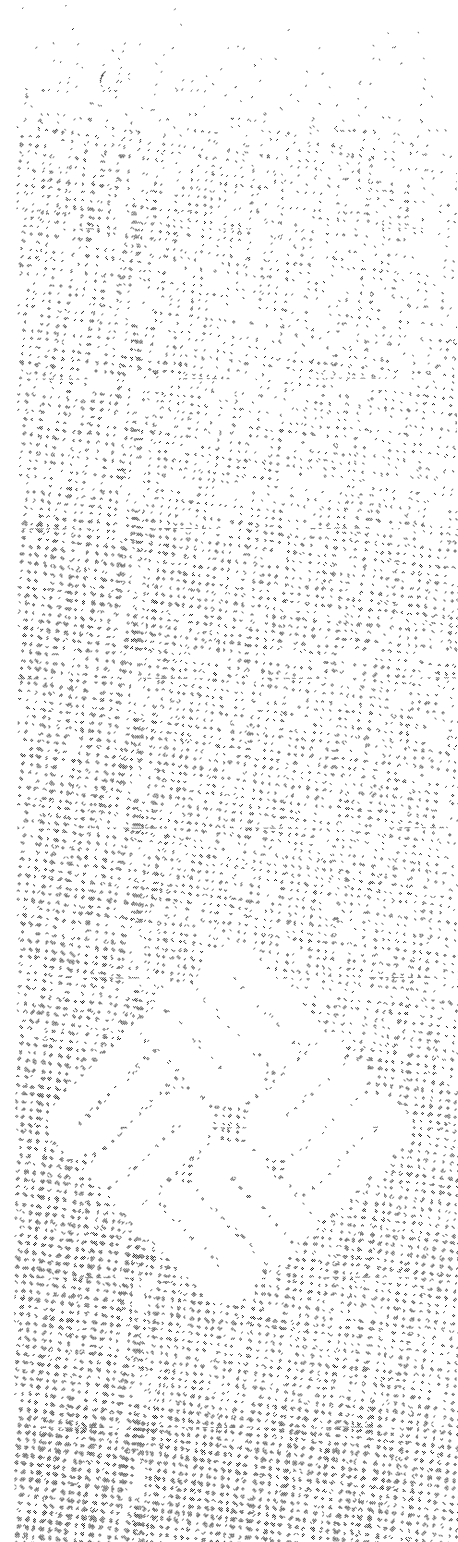
If your remote host is a file server and if earlier you obtained booting files from another machine or the installation tapes, you should now reinstall the original files. In the following sequence of commands, */pubarch* refers to the directory name that was being used for booting (typically */pub.MC68010* or */pub.MC68020*). *pub\_partition* refers to the disk controller abbreviation, disk unit number and partition letter of the partition that was being used for booting (for example, *xy0d*).

```
host# cd /pubarch
host# rm -rf stand
host# mv stand.orig stand
host# rm boot
host# mv boot.orig boot
host# cd /usr/mdec
host# installboot bootnd /dev/pub_partition
host# sync
host#
```

---

## Installing SunPro

|                                      |     |
|--------------------------------------|-----|
| Installing SunPro .....              | 237 |
| C.1. Installation Instructions ..... | 237 |





## Installing SunPro

This appendix explains how to finish installing the SunPro optional software. Before following the instructions in this section, you should have upgraded your system to 3.5 and selected the SunPro option during the installation process.

### C.1. Installation Instructions

Follow these instructions to install SunPro:

1. Enter

```
% cd /usr/sunpro
```

When you select SunPro during the 3.5 upgrade process, the operating system makes the directory `/usr/sunpro`, which contains the following new SunPro software:

```
compile default.mk filemerge.1 ld make
cpp filemerge install_sunpro make.1
```

`install_sunpro` is the shell script that you run to complete SunPro installation.

2. Enter

```
% install_sunpro
```

During the installation process, `install_sunpro` does the following:

- creates the directory `/usr/sunpro/3.2`, and copies pre-SunPro versions of `compile`, `/lib/cpp`, `/bin/ld`, `/usr/bin/m4`, `/bin/make`, and `/usr/man/man1/make1` into it.
- moves the files in `/usr/sunpro` to their appropriate places within the file system, overwriting older versions of the software with the new SunPro files.

**Note:** If you have a heterogeneous server, running this script installs SunPro on all of the server's architecture.

After you complete these steps, the system will automatically run SunPro versions of such files as `make` and `ld` when you invoke them. If you need more file space, you may want to delete the older files in `/usr/sunpro/3.2`.

For information about `filemerge`, refer to the `filemerge` manual page in the manual page package included with this release. The *make Manual*, also included in this release, explains the `make` tool available with SunPro.





# D

---

## Files to Be Saved

|                                           |     |
|-------------------------------------------|-----|
| Files to Be Saved .....                   | 241 |
| D.1. Standalone Systems .....             | 241 |
| D.2. Servers .....                        | 241 |
| D.3. Diskless Clients .....               | 242 |
| D.4. YP Masters and Non YP Machines ..... | 242 |



# D

---

## Files to Be Saved

### D.1. Standalone Systems

```
/.login
/.cshrc
/.rhosts
/exports
/etc/passwd
/etc/mtab
/etc/magic
/etc/printcap
/etc/fstab
/etc/hosts.equiv
/etc/dumpdates
/etc/hosts
/etc/remote
/etc/gettytab
/etc/rc.local
/etc/rc.boot
/etc/ttytype
/etc/ttys
/usr/lib/crontab
/usr/lib/sendmail.cf
```

### D.2. Servers

```
/.login
/.cshrc
/.rhosts
/etc/exports
/etc/remote
/etc/passwd
/etc/mtab
/etc/magic
/etc/printcap
/etc/fstab
/etc/hosts.equiv
/etc/dumpdates
/etc/hosts
/etc/gettytab
/etc/nd.local
```

```
/etc/rc.boot
/etc/rc.local
/etc/ttytype
/etc/ttys
/usr/hosts/MAKEHOSTS
/private/usr/lib/crontab
/private/usr/lib/sendmail.cf
```

### D.3. Diskless Clients

```
/.login
/.cshrc
/.rhosts
/etc/gettytab
/etc/remote
/etc/mtab
/etc/magic
/etc/passwd
/etc/printcap
/etc/fstab
/etc/hosts.equiv
/etc/hosts
/etc/rc.local
/etc/rc.boot
/private/usr/lib/crontab
/private/usr/lib/sendmail.cf
```

In addition to the files listed above, the following files will be saved if your system is a YP master or a non YP machine.

### D.4. YP Masters and Non YP Machines

```
/etc/ethers
/etc/netgroup
/etc/services
/etc/protocols
/etc/servers
/etc/group
/etc/networks
/etc/rpc
/usr/lib/aliases
yp/usr/etc/yp/domain/*(for
```

If your system is a YP slave server, all the YP maps under /usr/etc/yp/domain/ will be saved.

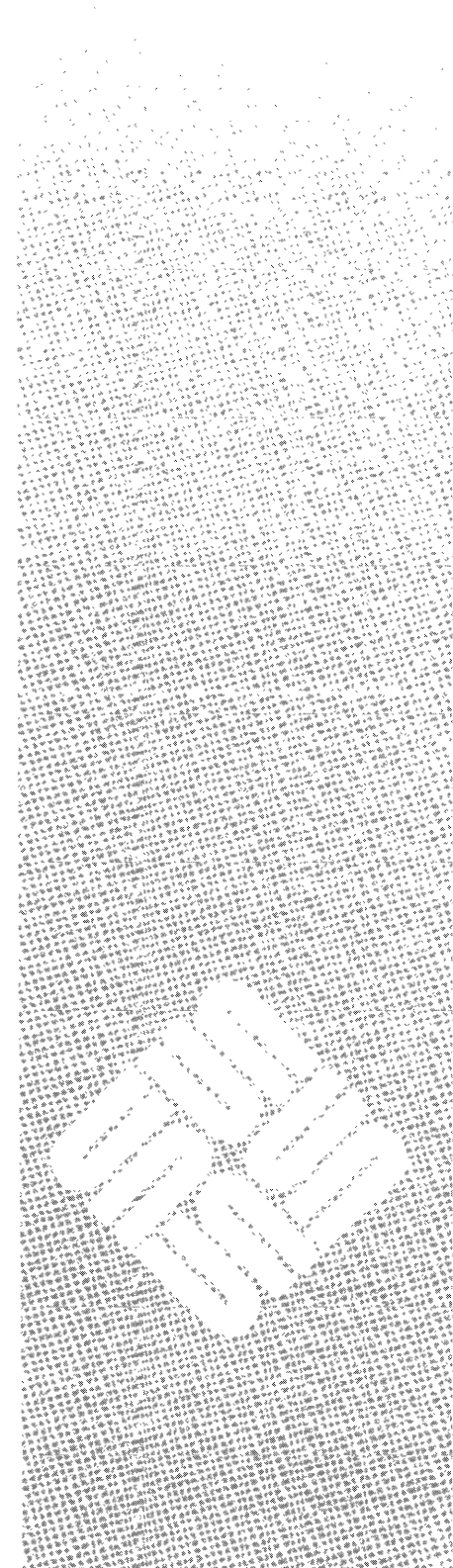
```
/usr/etc/yp/domain/*
```

# E

---

## Release 3.3 Change Notes

|                                                  |     |
|--------------------------------------------------|-----|
| Release 3.3 Change Notes .....                   | 245 |
| E.1. disconnect/reconnect .....                  | 245 |
| Product Description .....                        | 245 |
| Advantages to the User .....                     | 247 |
| E.2. Subnets .....                               | 247 |
| The Sun System Release 3.3 Implementation .....  | 247 |
| Enabling Subnets .....                           | 248 |
| Setting Up a Subnet .....                        | 249 |
| Examples of Subnets .....                        | 249 |
| E.3. Extensions to the Pascal Compiler .....     | 250 |
| char Data Type .....                             | 250 |
| Public or Private Declarations .....             | 250 |
| GETFILE Function .....                           | 250 |
| E.4. Bug Fixes .....                             | 250 |
| Locking Bugs .....                               | 251 |
| tbl Layout Processor Bug .....                   | 251 |
| Sun-3 ECC Error Message Bug .....                | 251 |
| Problems with 1/2 Inch Tapes on Sun 3/280s ..... | 251 |
| MT02 Tape Appending Problems .....               | 252 |
| MT02 and VPC-2000 Tape Hangs .....               | 252 |
| Interphase Controller Reference .....            | 252 |





---

## Release 3.3 Change Notes

This chapter describes the changes and new features available with the Sun System Release 3.3, including

- disconnect/reconnect
- subnets
- extensions to the Pascal compiler
- general bug fixes and documentation updates

### E.1. disconnect/reconnect

The disconnect/reconnect facility available with the Sun System Release 3.3 can be used on all Sun-3 models with the following:

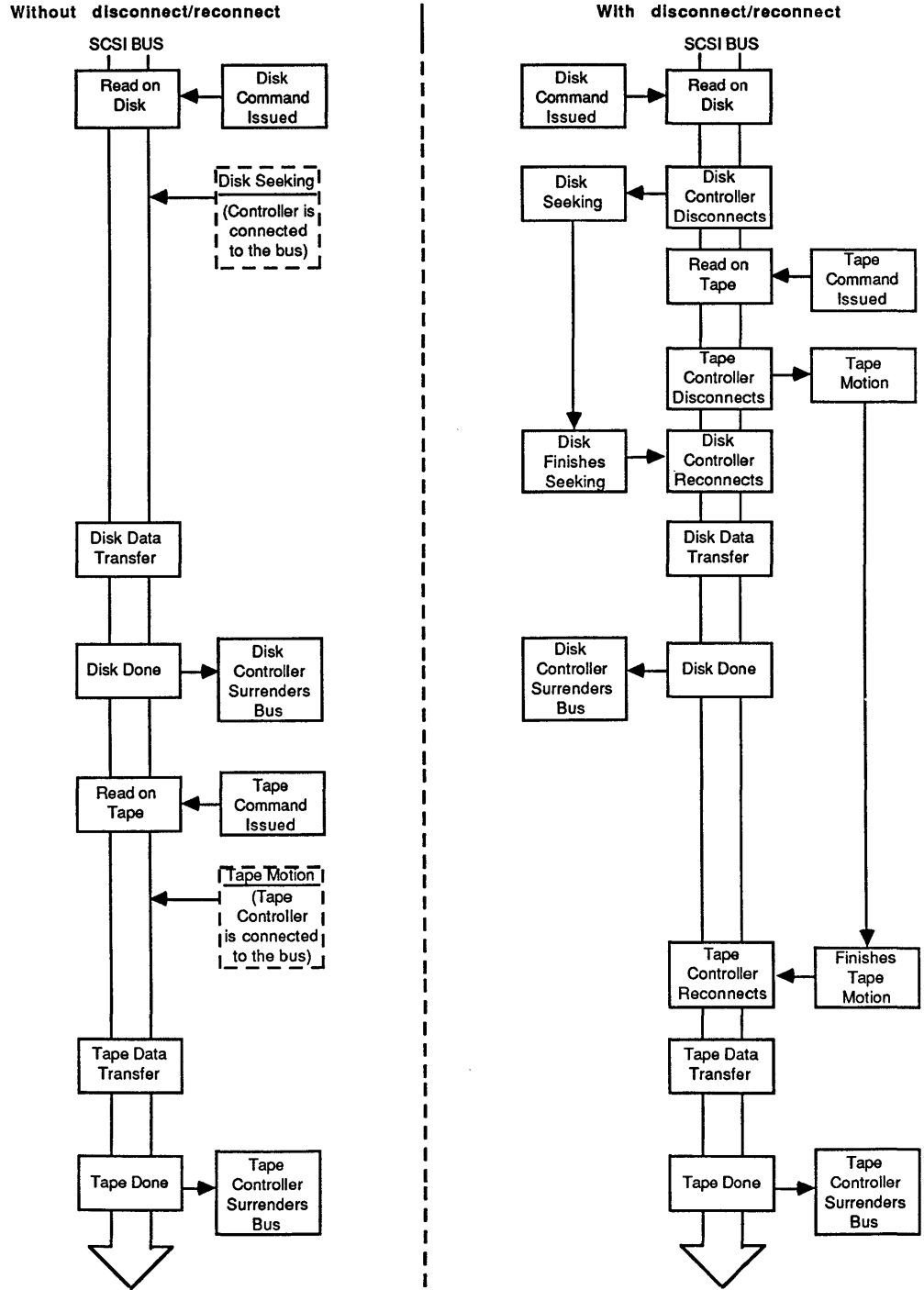
- SCSI-3 type host adapter
- SCSI disk controller(s)
- MT-02 tape controller(s)

### Product Description

disconnect/reconnect is a software facility that enables a disconnect-reconnect capable tape or disk controller to surrender access to (disconnect from) a SCSI bus while performing an operation that does not require the bus's data transfer capabilities. The controller can then regain access to (reconnect) the bus when it needs to resume data transfer. This improves data transfer efficiency over systems without disconnect/reconnect, since the controller no longer is attached to the SCSI bus for an entire "read" or "write" operation. The bus is free to handle incoming commands intended for other controllers, while the disconnected controller performs operations that do not require access to the bus, such as disk seeks or tape movement.

The following chart gives a general picture of the interaction among controllers and a SCSI bus on a system that uses disconnect/reconnect and on one that does not.

Figure E-1 *SCSI Bus Usage with and without disconnect/reconnect*





As you can see, controllers that do not implement disconnect/reconnect tie up the bus. Once a controller attaches itself to the bus, it locks out all other processing, until data transfer has completed. This is usually not noticeable during disk reads. However, tape operations are significantly slower and noticeably “freeze” the system during the operation, because they prevent disk accesses, such as those that occur during swapping.

With disconnect/reconnect, however, the controller disconnects from the bus during the relatively time-consuming seeking process. Therefore, the bus can handle requests from other processes while the controller seeks to its desired location.

## Advantages to the User

disconnect/reconnect may improve performance on systems that have overlapped I/O operations—the typical UNIX-based system. It is not truly necessary for systems that perform all peripheral operations serially. The most noticeable improvement on a system with disconnect/reconnect is that the system does not seem to pause during a tape operation such as a rewind. Since the tape device no longer takes over the bus for the duration of a tape operation, the system does not “lock out” during that tape operation.

## E.2. Subnets

Sun System Release 3.3 includes limited support for Internet standard subnets. These subnets are compatible with all SunLink products, particularly the SunLink inter-network router. You can use them with all Sun architectures; however, to perform a tapeless install on a subnet, you must use a tape server running Release 3.0 or later releases.

Subnets allow more flexibility in the assignment of network addresses. Normally, two bits of the 32 bit Internet Protocol (IP) address tell how much is a network number. There are 127 Class A networks with 24 bit host fields, 16,383 Class B networks with 16 bit host fields, and over two million Class C networks with eight bit host fields.

The problem is that routing can get very complicated as the number of networks grows. For example, a small organization might give each Ethernet a Class C number. As the organization grows, the administration of network numbers could get out of hand. A better idea is to allocate a few Class B network numbers: one for Engineering, one for Operations, one for Support, one for Sales, and so on. Then, divide each Class B network into physical Ethernets using subnets. In this way, machines are isolated from topology changes in remote parts of the organization.

## The Sun System Release 3.3 Implementation

When setting up your network, a network-wide *network mask* must be selected. The network mask determines which bits will be the subnet number, and the rest is the host within the subnet. For example, an organization could be one Class B network, with each Ethernet (or SunLink connection) assigned a subnet number within that network. The 16 bits could be allocated as eight for subnet and eight for host, or nine for subnet and seven for host, and so on. However, this decision would be transparent to everyone outside that organization.

You can express network masks as a single hexadecimal number, or as a sequence of four decimal numbers in the IP “dotted notation.” The default is a mask of 0xFF000000 (255.0.0.0) for Class A networks, 0xFFFF0000

(255.255.0.0) for Class B networks, and 0xFFFFFFFF (255.255.255.0) for Class C networks. Network masks must only be explicitly specified when they are “wider” (that is, have more one-bits) than the default values. One common case is a Class C mask on a Class B network.

Previously, the check for matching networks always applied the default masks. The check now allows you to use one non-standard mask when the network matches a saved value. Thus, Release 3.3 has one important limitation in its support of subnets: all interfaces with non-default subnet masks must be on the same IP network (but may be on different subnets). In other words, you cannot have more than one subnetted network interfaced to any machine. Usually, a workstation will be on only one subnet; a server will be a gateway between subnets of the same net, and possibly other non-subnetted networks. These cases are supported by 3.3. This temporary restriction should be removed in a future release.

Sun System Release 3.3 also fixes the broadcast problem. Some implementations of IP send broadcasts with a normal network (or subnet) field, but with a host field of all ones. This is correct, according to the IP specification, but may cause earlier Sun software to essentially bring down the network, by broadcasting ARP requests for Host 255. Note that all-one host numbers (for example, x.y.z.255 for Class C networks) work in previous Sun System releases, even though they are not allowed by the IP specification. Starting with this release, all-one host numbers are treated properly, as broadcast.

## Enabling Subnets

In addition to the kernel modules needed to support subnets, two programs are involved. The `/etc/ifconfig` program has a new option to set the network mask, and a new `/etc/in.routed` is needed to dynamically maintain the routing tables.

Normally, you should edit the `/etc/rc.local` file on diskfull machines to add an `/etc/ifconfig netmask` command, or else edit the `/etc/rc.boot` file to add the `netmask` option to the `ifconfig` command. Workstations that are booted over the network should then set their masks from their file servers using a network protocol. You can also manually set the kernel variables `subnet_net` and `subnet_mask` using `adb`, although this is discouraged.

For example, consider the Class B network 128.32 with an eight-bit wide subnet field (and, therefore, an eight-bit wide host field), with a server that is Host 1 on Subnet 37, Host 100 on Subnet 3, and is also a gateway to Network 10. The `/etc/rc.boot` file could be as follows:

```
/etc/ifconfig ie0 128.32.37.1 netmask 255.255.255.0 -trailers up
/etc/ifconfig ie1 128.32.3.100 netmask 255.255.255.0 -trailers up
/etc/ifconfig imp0 10.2.0.78
```

Symbolic names for subnet addresses can be defined in the `/etc/hosts` file. These subnet names can be used instead of the numeric addresses as parameters to commands.

## Setting Up a Subnet

Follow these steps to set up a subnet.

1. Decide upon the new subnet topology, including considerations for subnet gateways and locations of hosts on the subnets.
2. Assign all subnet and host addresses.
3. Edit `rc.local` if the host is a diskfull server or standalone workstation.
4. Change hosts addresses.
5. Edit `/etc/hosts` files to change host address. If `yp` is running, change the `yp` database.
6. ND servers should be sure the `rarpd` runs on the interface connected to the ND client's subnet. The `rarpd` entry is in the `rc.local` file. Change the IP address of ND clients in the `/etc/tftpboot` directory.
7. `yp` servers should also follow the setup procedure described in *System Administration for the Sun Workstation* (800-1323-03).
8. Connect the physical networks and reboot all machines.

## Examples of Subnets

The following examples show network installations where subnets are (and are not) in use:

```

128.32.0.0 Berkeley class B network (subnetted) netmask 255.255.255.0
36.0.0.0 Stanford class A network (subnetted) netmask 255.255.0.0
10.0.0.0 ARPAnet class A network (non-subnetted) netmask 255.0.0.0

```

All of the University of California at Berkeley is assigned the network number 128.32.0.0, so that any outside user only needs to know one route to access Berkeley. Within the campus, a class C subnet mask is used to give each individual Ethernet a subnet number, with 254 hosts on each of the 254 possible subnets. (Zero and all ones are reserved.) Stanford University uses a class A network number with a class B network mask, for 254 subnets of 65534 hosts each. The ARPAnet itself is a class A network without subnets; therefore, the default class A netmask is used.

Here are examples showing legal and illegal subnet configurations:

- Configuring a Sun with interface addresses of 128.32.1.1 and 128.32.2.1 is legal for Sun 3.3 (two subnets of same net).
- Configuring a Sun with interface addresses of 128.32.1.1 and 128.32.2.1 and 10.2.0.78 is legal for Sun 3.3 (two subnets of same net and non subnetted network).
- Configuring a Sun with interface addresses of 36.8.0.8, 36.10.0.1, and 10.2.0.11 is legal for Sun 3.3 (two subnets of same net and non subnetted network).
- Configuring a Sun with interface addresses of 128.32.1.1, 128.32.2.1, and 36.8.0.8 is **not** legal for Sun 3.3 (two subnets of the same net and another subnetted network).

### E.3. Extensions to the Pascal Compiler

Release 3.3 includes two extensions to the Pascal compiler (pc). It also includes a new function, GETFILE, which has been added to libpc.a. Note that programs compiled with earlier versions of pc are upward-compatible with the new compiler.

#### char Data Type

The valid range of the char data type is now 0 to 255. Its previous range was 0 to 127.

#### Public or Private Declarations

You can now declare variables, procedures, or functions at the outer block level as either *public* or *private*. For example,

```
var
 total : real;
 quantity : integer;

private var
 score : integer;

private procedure eval(n : integer);
begin
 .
 .
```

The default scope of an outer block variable, procedure, or function is public. If you declare a variable, function, or procedure as private, then its scope is restricted to the current compilation unit.

#### GETFILE Function

The Pascal runtime library, /usr/lib/libpc.a, now contains the GETFILE function, which simplifies using the C standard I/O library from Pascal programs.

To use GETFILE, declare the following from Pascal:

```
type
 iobptr = ^integer;

function GETFILE(var f: text): iobptr;
external c;
```

GETFILE(f) returns a pointer to the C standard I/O descriptor associated with the Pascal file f. The result may be passed to fprintf(), fread(), or other C standard I/O library routines.

### E.4. Bug Fixes

The following bugs have been fixed in Release 3.3:

- record locking bugs
- tbl table layout preprocessor bug
- Sun-3/200 ECC error message bug
- problems with 1/2 inch tape on Sun 3/280s

- MT02 tape appending problems
- MT02 and VP-2200 tape hangs
- documentation update

These changes are described below.

### Locking Bugs

Two file locking bugs files have been fixed.

Using the System V record locking features `lockf` or `fcntl` on a remote file caused the end-of-file mark to be ignored. In such cases, a `read()` past the end-of-file marker always returned successfully. This condition, originally noted in Release 3.2, *Read This First*, has been corrected for Release 3.3.

Another locking-related bug in Release 3.2 caused NFS servers to crash when a client read from a file locked by the System V-compatible `fcntl` or `lockf` routines. The problem only occurred when the client read started at an offset that was not a multiple of four, and the data types crossed a block boundary in the server's file system. This bug has been fixed in Release 3.3.

### tbl Layout Processor Bug

In Release 3.2, `tbl` did not correctly format numeric fields in tables. This has been fixed in Release 3.3.

### Sun-3 ECC Error Message Bug

In Release 3.2, if a Sun-3/200 had more than one memory board and a correctable memory error occurred, the ECC address of the error, as displayed in the error message, was computed incorrectly.

The old message format was

```
mem1: soft ecc addr 20000000 + bfe980 = 20bfe980 syn 97 <S32,
S4, S1, S0, SX>
```

This error has been corrected. The new format of the address is

```
mem1: soft ecc addr 394980 syn 97 <S32, S4, S1, S0, SX> 59 01459
```

### Problems with 1/2 Inch Tapes on Sun 3/280s

Sun-3/280 Workstations configured with GCR 1/2 inch tape drives and 472 Xylogics 1/2 inch tape controllers running 3.2 or previous releases may have experienced failures when executing `tar` or `dump`. These commands failed with the following error message:

```
xt: bad command synchronization
```

This has been fixed in Release 3.3.

### MT02 Tape Appending Problems

MT02 tape controller users sometimes were unable to append new files to the end of a tape because it was difficult to reliably reach the end of recorded media. The tape driver would report an error upon reaching the end of tape when the `mt` command was used. This error has been eliminated.

It is suggested that you use the following procedure for positioning to the end of the recorded media:

```
% mt -f /dev/nrst0 fsf nnn
```

`nnn` represents a large count—at least one more than the actual number of files on the tape. Do not worry about the large `fsf` count because any extra `fsf`'s are ignored by the driver. Note that you must use the no-rewind device `/dev/nrst0`. Otherwise, the tape rewinds when the `mt` command exits, undoing the positioning to the end of recorded media.

Please note that appending data is supported only at the end of recorded media. Writing data in the middle of the tape is not supported, due to hardware limitations.

### MT02 and VPC-2000 Tape Hangs

Systems with both the MT02 tape controller and the Systech VPC-2200 parallel printer/plotter interface were experiencing tape hangs. When a user removed a tape, this left the tape drive in a state that prevented initialization when the user inserted the next tape, resulting in the message:

```
st0 : no cartridge loaded
```

This has been fixed in Release 3.3

### Interphase Controller Reference

Do not use the Interphase Controller reference (`ip0`) on Page 21 of *Installing UNIX on the Sun Workstation* for the 3.2 Release. This controller is no longer supported; therefore, the kernel does not recognize this root device.

---

Notes

---

Notes





**Corporate Headquarters**

Sun Microsystems, Inc.  
2250 Garcia Avenue  
Mountain View, CA 94043  
415 960-1300  
TLX 287815

**For U.S. Sales Office  
locations, call:**  
800 821-4643  
In CA: 800 821-4642

**European Headquarters**

Sun Microsystems Europe, Inc.  
Sun House  
31-41 Pembroke Broadway  
Camberley  
Surrey GU15 3XD  
England  
0276 62111  
TLX 859017

**Australia:** 61-2-436-4699

**Canada:** 416 477-6745

**France:** (1) 46 30 23 24

**Germany:** (089) 95094-0

**Japan:** (03) 221-7021

**The Netherlands:** 02155 24888

**UK:** 0276 62111

**Europe, Middle East, and Africa,  
call European Headquarters:**  
0276 62111

**Elsewhere in the world,  
call Corporate Headquarters:**  
415 960-1300  
Intercontinental Sales

