



---

# SunOS 4.1.2 Release Manual

---

Sun Microsystems, Inc. • 2550 Garcia Avenue • Mountain View, CA 94043 • 415-960-1300

Part No: 800-6645-10  
Revision A of December 1991

The Sun logo, Sun Microsystems, Sun Workstation, NFS, and TOPS are registered trademarks of Sun Microsystems, Inc. Sun, Sun-2, Sun-3, Sun-4, Sun386i, NeWS, NSE, OpenWindows, SunInstall, SunLink, SunNet, SunOS, SunPro, and SunView, are trademarks of Sun Microsystems, Inc.

SPARC is a registered trademark of SPARC International. SPARCstation and SPARCserver are trademarks of SPARC International, licensed exclusively to Sun Microsystems, Inc. UNIX and OPEN LOOK are registered trademarks of UNIX System Laboratories, Inc. Ethernet is a registered trademark of Xerox Corporation, Inc.

X/Open is a Trade Mark of X/Open Company Limited in the UK and other countries. XPG3 Trade Marks are applied to products operating in specific hardware/software environments. Consult the applicable XPG3 Conformance Statement for important details about registered products. The Conformance Statement may be found in Appendix E.



The use of this logo certifies SunOS 4.1 conformance with X/Open Portability Guide Issue 2 (XPG2). This logo is a trademark of the X/Open Company Limited in the UK and other countries, and its use is licensed to Sun Microsystems, Inc.



**LEGAL NOTICE TO USERS:** Yellow Pages is a registered trademark in the United Kingdom of British Telecommunications plc., and may also be a trademark of various telephone companies around the world. Sun will be revising future versions of software and documentation to remove references to Yellow Pages.

All other products or services mentioned in this document are identified by the trademark or service marks of their respective companies or organizations, and Sun Microsystems, Inc. disclaims any responsibility for specifying which marks are owned by which companies or organizations.

Copyright © 1991 Sun Microsystems, Inc. – Printed in U.S.A.

All rights reserved. No part of this work covered by copyright hereon may be reproduced in any form or by any means – graphic, electronic, or mechanical – including photocopying, recording, taping, or storage in an information retrieval system, without the prior written permission of the copyright owner.

**RESTRICTED RIGHTS LEGEND:** Use, duplication, or disclosure by the U.S. government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 (Oct. 1988) and FAR 52.227-19 (c) (June 1987). Sun Microsystems, Inc., 2550 Garcia Avenue, Mountain View, California 94043.

The Sun Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees.

This product is protected by one or more of the following U.S. patents: 4,777,485 4,688,190 4,527,232 4,745,407 4,679,014 4,435,792 4,719,569 4,550,368 in addition to foreign patents and applications pending.

This software and documentation is based in part on the Fourth Berkeley Software Distribution under license from the Regents of the University of California. We acknowledge the following individuals and institutions for their role in its development: The Regents of the University of California, the Electrical Engineering and Computer Sciences Department at the Berkeley Campus of the University of California, and Other Contributors.

# Welcome!

Welcome to SunOS 4.1.2. This preface describes how this manual is organized, with brief descriptions of each section. The manual has five parts:

- **Introduction**
- **Features**
- **Installation and System Configuration**
- **Open Issues**
- **Appendixes**

# Introduction

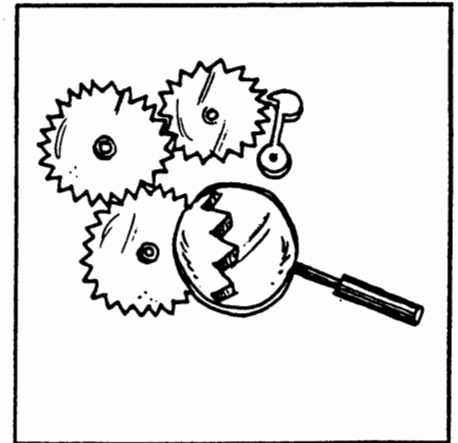
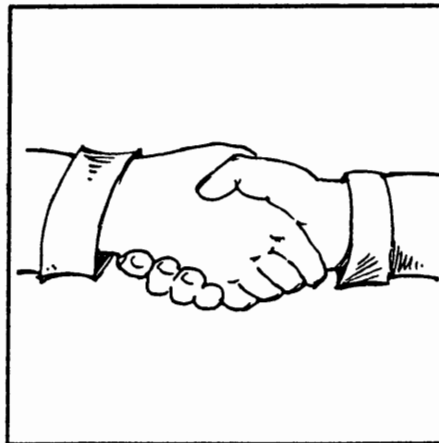
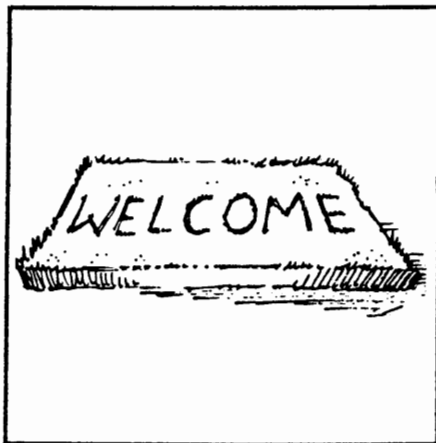
What is in this manual? What should you do before installing this release? Part 1 has one chapter, which contains the following sections:

- **Release Compatibility**
- **Workstation and Machine Architecture Terminology**
- **Getting Help**
- **Guide to Publications**
- **Documentation Conventions**

# Features

What is new in SunOS 4.1.2? Part 2 contains detailed descriptions of all new features and changes. It also describes all bugs that have been fixed since SunOS 4.1.1 Rev B. This part contains the following chapters:

- **New Software Features**
- **About Openwindows**
- **New Hardware Features**
- **Open Boot PROM 2.0 and Hierarchical Device Naming**
- **Bugs Fixed in SunOS 4.1.2**



# Installation and System Configuration

Use Part 3 along with other installation documents when installing SunOS 4.1.2. This part contains the following chapters:

- **Installation Notes and Warnings**

Notes and warnings about installing unbundled products.

- **System Configuration**

Hints for kernel configuration and for improving system performance.

# Open Issues

Part 4 contains the latest information on this SunOS release. Be sure to read this material, which is separate from the *Release Manual* in the Release Minibox, and be sure to insert it behind the *Open Issues* tab. This part contains the following chapters:

- **Late-Breaking News**

Includes the latest news about SunOS 4.1.2

- **Known Problems**

Describes any known problems in the release

# Appendixes

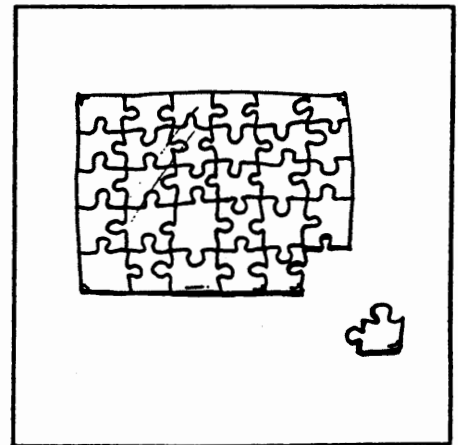
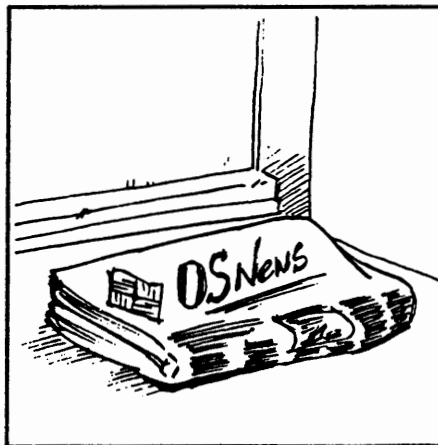
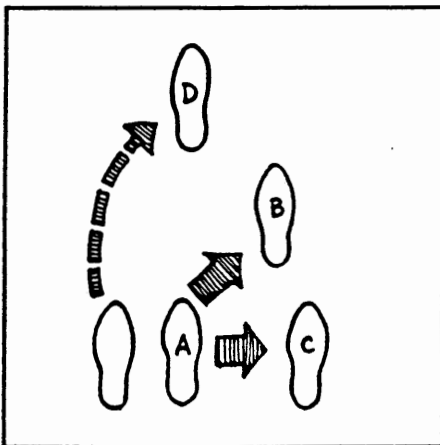
Refer to this part for detailed information about:

- **Documents Available for Use with SunOS 4.1.2**

- **Addendum to Writing Device Drivers**

- **Sundiag Version 2.3 for SunOS 4.1.2**

- **X/Open Conformance Statement for SunOS 4.1 Products Receiving the XPG3 Component Brand**



---

# Contents

<b>Part 1 — Introduction</b> .....	
<b>Chapter 1 Introduction</b> .....	<b>1-1</b>
1.1. Release Compatibility .....	1-1
SunOS 4.1.2 Binary Compatibility with SunOS 4.1 .....	1-1
Binary Compatibility with SunOS 4.0.X .....	1-2
1.2. Workstation and Machine Architecture Terminology .....	1-3
Application and Kernel Architecture .....	1-3
1.3. Getting Help .....	1-4
1.4. Guide to Publications .....	1-5
1.5. Documentation Conventions .....	1-5
<code>.cshrc</code> Could Be <code>.profile</code> .....	1-5
Type Styles .....	1-5
Boxes .....	1-5
<b>Part 2 — Features</b> .....	
<b>Chapter 2 New Software Features</b> .....	<b>2-1</b>
2.1. General Software Changes .....	2-1
Sunupgrade for Upgrading to SunOS 4.1.2 .....	2-1
Graphics Support and Fixes .....	2-1
Quicker File System Checks .....	2-1
Rock Ridge File System Supported on CD-ROM .....	2-2
BSD 4.3 <code>malloc</code> Available .....	2-2

2.2. Link-editor Bug Fixes .....	2-2
2.3. SPARCsystem 600MP Systems .....	2-5
Description of New Features .....	2-5
The SPARCsystem 600MP Multiprocessing Environment .....	2-6
2.4. Writing Device Drivers for SPARCsystem 600MP Systems .....	2-6
<b>Chapter 3 About OpenWindows .....</b>	<b>3-1</b>
3.1. Why Use OpenWindows Version 3? .....	3-1
3.2. General OpenWindows Features .....	3-2
X Window System .....	3-2
NeWS .....	3-2
XView Toolkit .....	3-2
NeWS Toolkit .....	3-2
OPEN LOOK Interface .....	3-2
OPEN LOOK Intrinsic Toolkit .....	3-2
OpenWindows DeskSet Environment .....	3-2
3.3. About OpenWindows Version 2 .....	3-3
Do Not Install Sun IPX Supplement .....	3-3
Keyboard Map File for Japanese International Keyboard .....	3-3
Using SunView for Maximum Security Levels .....	3-3
3.4. Which Default Window System? .....	3-3
New User; OpenWindows Comes Up as Default .....	3-3
New User; SunView Comes Up as Default .....	3-4
New User; No Default Window System .....	3-4
Old User; Previous Window System .....	3-5
3.5. Changing the Window System to be Invoked upon Login .....	3-5
3.6. Online Desktop Tutorial .....	3-5
3.7. OpenWindows Online Tutorial .....	3-5
<b>Chapter 4 New Hardware Features .....</b>	<b>4-1</b>
4.1. Peripheral Support .....	4-1
4.2. Graphics Support .....	4-2
GXplus Graphics Accelerator .....	4-2

GS Graphics Accelerator .....	4-2
GT Graphics Accelerator .....	4-3
VX Graphics Accelerator .....	4-3
TC-Compatible Frame Buffer .....	4-3
4.3. Support for SPARCsystem 600MP Series .....	4-3
Support for Multiple SCSI Buses on SPARCsystem 600MP .....	4-4
Open Boot PROM .....	4-4
Sbus and Mbus Expansion Buses .....	4-4
Improved Network Connections .....	4-4
Enclosure Configurations .....	4-4
CPU Board Bus and Port Options .....	4-4
CPU Board Memory Options .....	4-5
Monitor Options .....	4-5
Graphics Support .....	4-5
Printer Interface Options .....	4-5
4.4. Related SPARCsystem 600MP Documents .....	4-5
<b>Chapter 5 Open Boot PROM 2.0 and Hierarchical Device</b>	
<b>Naming .....</b>	<b>5-1</b>
5.1. Built-in Aliases for Standard Devices .....	5-1
Switching from Restricted Monitor to Forth Monitor .....	5-2
Booting Open Boot PROM 2.0 Systems from the CPU (except 690MP) .....	5-2
Booting a SPARCsystem 690MP from the CPU .....	5-3
Changing Default Boot Device .....	5-3
5.2. Hierarchical Device Naming .....	5-4
5.3. Device Naming Conventions .....	5-4
5.4. Using <code>show-devs</code> Command to Display Device Names .....	5-5
5.5. Using <code>devalias</code> Command to Show and Define Device Aliases .....	5-6
5.6. Open Boot PROM Environment Parameters Stored in NVRAM .....	5-6
<b><code>printenv</code> Command .....</b>	<b>5-6</b>
<b><code>setenv</code> Command .....</b>	<b>5-7</b>
<b><code>set-default</code> and <code>set-defaults</code> Commands .....</b>	<b>5-7</b>

<b>Chapter 6</b> Bugs Fixed in SunOS 4.1.2 .....	<b>6-1</b>
6.1. Kernel .....	6-1
6.2. Network .....	6-3
6.3. SCSI .....	6-4
6.4. Graphics .....	6-5
6.5. Utility .....	6-5
6.6. Library .....	6-6
6.7. Libsuntool .....	6-6
6.8. Sunview .....	6-6
6.9. Compiler/Loader .....	6-6
6.10. C2 .....	6-7
6.11. Documentation .....	6-7
6.12. Install .....	6-7
6.13. CTE Escalations Fixed in SunOS 4.1.2 .....	6-7
6.14. CTE OpenWindows Version 2 Patches .....	6-10

### **Part 3 — Installation and System Configuration .....**

<b>Chapter 7</b> Installation Notes and Warnings .....	<b>7-1</b>
7.1. Upgrade Utility Available .....	7-1
7.2. Do Not Install Earlier Patches to SunOS 4.1.2 .....	7-1
7.3. Installing Unbundled Products on a Server Supporting Multiple Releases .....	7-1
Workaround to Install Language Products for a Client's Release .....	7-2
7.4. OpenWindows Version 2 Restrictions .....	7-3
Do Not Install Sun IPX Supplement .....	7-3
OpenWindows Does Not Fit on 104MB System Disks .....	7-3
8MB Main Memory, 20MB Swap Partition, Required .....	7-3
OpenWindows Requires SunView Libraries In Order to Run .....	7-4
7.5. Graphics: .....	7-4
7.6. Hardware Patches .....	7-4
Do <b>Not</b> Install SunOS 4.1 Patch for Running SPARCprinter on 4.1.2 System .....	7-4



Do Not Install Sun 1.3GB Disk Enhancement .....	7-4
7.7. Prestoserve 2.0 and 2.3 .....	7-5
Error Message During Prestoserve 2.3 Installation .....	7-5
Prestoserve 2.0 Sundiag Problem (1067352) .....	7-5
7.8. NSE 1.2 not Compatible with SunOS 4.1.2 .....	7-5
7.9. Sun DBE 1.0 and 1.1 Not Supported under SunOS 4.1.2 .....	7-5
7.10. Sun Cross Compilers 3.0 Not Supported under SunOS 4.1.2 .....	7-5
7.11. SunTrac Tutorial Date Errors .....	7-5
7.12. SPE 1.1 Problems .....	7-6
SPE Build Failure .....	7-6
Stack Overflows .....	7-6
7.13. SunShield 1.0 on SunOS 4.1.2 .....	7-6
7.14. SunNet/SunLink Networking Products .....	7-7
Installing the Internetwork Router .....	7-8
Exporting SunLink Software for Multiple SunOS Versions .....	7-8
setsid Problems When Running SunLink DNI 6.0 .....	7-9
setsid Problems When Running SunLink X.25 6.0 .....	7-10
Incorrect Instruction in <i>SunLink BSC3270 System</i> <i>Administration Guide</i> (1044933) .....	7-10
7.15. Language Products .....	7-11
Sun C++ 2.1 Patch Installation .....	7-11
FORTRAN 1.4 Patch Installation .....	7-13
Pascal 2.1 Patch Installation .....	7-14
FORTRAN, C, Pascal, Modula-2: Missing Debugging Libraries .....	7-14
FORTRAN 1.2 Problems .....	7-15
7.16. Type-4 Keyboard Compatibility with Unbundled Products .....	7-15
NeWS 1.1 on Type-4 Keyboards .....	7-16
<b>Chapter 8 System Configuration .....</b>	<b>8-1</b>
8.1. About Kernel Configuration .....	8-1
Sun-Supplied Kernel Configuration Files .....	8-2
8.2. Device Configuration on Open Boot PROM Systems .....	8-3
Declaring SCSI Buses, Disks, and Tapes .....	8-4

Booting from IPI Disks on SPARCsystem 600MP Systems .....	8-5
8.3. Customizing the Kernel for a Large Server .....	8-6
Maximum <b>maxusers</b> Values for Sun-4, Sun-4c, Sun-4m Servers .....	8-6
Formula for Determining <b>maxusers</b> .....	8-6
Adjust Number of ALM-2 Ports .....	8-7
Create <code>/dev</code> Entries .....	8-7
Make Entries in <code>/etc/ttytab</code> .....	8-8
Possible Error Messages .....	8-8
Monitoring Performance .....	8-9
8.4. Performance Improvement Hints .....	8-9
Use the <b>tmpfs</b> Filesystem .....	8-9
Use Static Routing When Possible .....	8-9
Do Not Enable Process Accounting .....	8-10
Eliminate Unnecessary Server Processes .....	8-10
Do Not Enable File-System Quotas .....	8-10
8.5. Disk Layout for Systems with 104MB Disks .....	8-10
Desktop SPARCsystem with Two 104MB Disks: Using Second Disk for <code>/home</code> and Additional Swap Space .....	8-10
Using the <code>g</code> Partition on Your Second Disk as <code>/home</code> .....	8-11
Using Your Second Disk for Added Swap Space .....	8-12
Desktop SPARCsystem with Two 104MB Disks: Using Second Disk for OpenWindows .....	8-12
If You Use SunInstall .....	8-13
Systems with SunOS 4.1.X through Preinstallation, Quick Install, or Re-preinstall .....	8-13
Installing OpenWindows Software in <code>sd1g</code> .....	8-15
 <b>Part 4 — Open Issues</b> .....	
 <b>Chapter 9</b> Late-Breaking News .....	<b>9-1</b>
 <b>Chapter 10</b> Known Problems .....	<b>10-1</b>
10.1. System Administration .....	10-2

10.2. Kernel .....	10-4
10.3. Graphics .....	10-5
10.4. Network .....	10-5
10.5. Utilities .....	10-7
10.6. Compiler .....	10-9
10.7. Hardware .....	10-11
10.8. PROM .....	10-14
10.9. Sundiag .....	10-17
10.10. Sun 386i .....	10-18
10.11. Documentation .....	10-18
10.12. CD-ROM .....	10-21
10.13. Library .....	10-22
10.14. Miscellaneous .....	10-22
10.15. OpenWindows Version 2 .....	10-22
10.16. OpenWindows Version 3 .....	10-32

**Part 5 — Appendixes** .....

<b>Appendix A</b> Documents Available for Use with SunOS 4.1.2 .....	<b>A-1</b>
Books: <i>User's Guides</i> .....	A-1
Binder: <i>System and Network Administration</i> .....	A-1
Books: <i>OpenWindows End User Manuals</i> .....	A-1
Binders: <i>SunOS Reference Manual (3 Volumes)</i> .....	A-1
Binder: <i>Global Index</i> .....	A-2
Binder: <i>SunOS Documentation Tools</i> .....	A-2
Binder: <i>Programmer's Guides</i> .....	A-2
Binder: <i>Programmer's Overview Utilities and Libraries</i> .....	A-2
Binder: <i>Network Programming Guide</i> .....	A-2
Binder: <i>Writing Device Drivers/STREAMS Programming</i> .....	A-3
Binder: <i>SunView Programmer's Guide</i> .....	A-3
Binder: <i>SunView 1 System Programmer's Guide</i> .....	A-3

<b>Appendix B</b> Addendum to Writing Device Drivers .....	<b>B-1</b>
--	------------

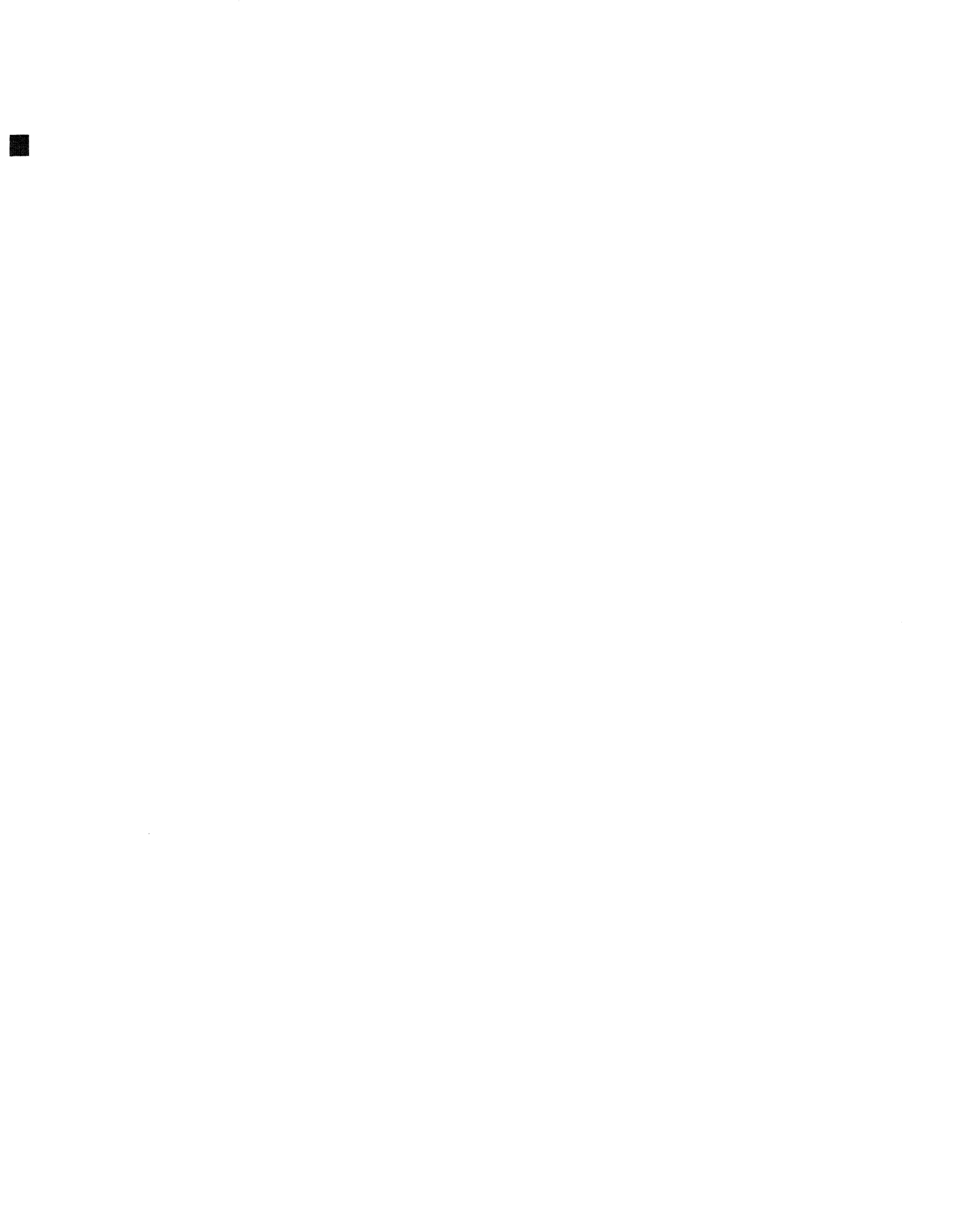
B.1. Sun-4m Device Driver Developer Notes .....	B-1
Compatibility .....	B-2
Interrupts .....	B-3
Write Buffers .....	B-3
Performance Tuning .....	B-3
IOMMU Bypass Mode .....	B-4
SBus Slot Configuration Register .....	B-4
Open Boot PROM .....	B-4
B.2. SunOS 4.1.1 Changes .....	B-5
Changes Since Earlier Releases .....	B-7
<b>Appendix C Sundiag version 2.3 for SunOS 4.1.2 .....</b>	<b>C-1</b>
C.1. Sundiag for the Multiprocessing Chip .....	C-2
Running Sundiag From Window Interface .....	C-2
Configurations .....	C-3
Subtests .....	C-3
Options .....	C-3
Command-line Syntax .....	C-4
C.2. Error .....	C-5
<b>Appendix D X/Open Conformance Statement (XCS-QUE-</b>	
<b>3.106) .....</b>	<b>D-1</b>
<b>Index .....</b>	<b>Index-1</b>

---

## Tables

Table 1-1 System Architectures and Machine Names .....	1-4
Table 5-1 Booting an Open Boot PROM 2.0 System from the CPU .....	5-2
Table 5-2 Booting a SPARCsystem 690MP from the CPU with Open Boot PROM 2.0 .....	5-3
Table 8-1 Sun-Supplied Kernel Configuration Files for sun4c Architectures .....	8-2
Table 8-2 Sun-Supplied Kernel Configuration Files for sun4 Architectures .....	8-3
Table 8-3 Sun-Supplied Kernel Configuration Files for sun4m Architecture .....	8-3
Table 8-4 Maximum <code>maxusers</code> Values for Sun-4, Sun-4c, Sun-4m .....	8-6





---

# Part 1 — Introduction

Read Part 1 to get a quick introduction to the release, including pointers to other parts of the *Release Manual* and to other documentation. Read Part 2, *Features*, for a detailed description of the release, and Parts 3 and 4, *Installation* and *Open Issues*, for the latest information about installing and using the release.

Chapter 1, “Introduction,” includes the following sections:

- Release Compatibility
- Workstation and Machine Architecture Terminology
- Getting Help
- Guide to Publications
- Documentation Conventions







---

# Introduction

Welcome to SunOS™ 4.1.2. This software release includes the following:

- Support for the new SPARCsystem 600MP series of systems
- An incremental software upgrade program (from SunOS 4.1.1 and 4.1.1 Rev B)
- Incorporation of SunOS 4.1.1 Rev B and SunOS 4.1.1 GFX releases
- More than 200 SunOS bug fixes

SunOS 4.1.2 is released on all SPARC™ platforms (CD-ROM only), and is fully compatible with SunOS 4.1.1 and SunOS 4.1.1 Rev B. See “New Software Features” for a detailed description of the new software features and “New Hardware Features” for a detailed description of the newly-supported hardware.

Be sure to read the material in the *Open Issues* package from the SunOS Release Minibox before proceeding with the installation or upgrade; this package, comprised of Chapter 9, “Late-Breaking News,” and Chapter 10, “Known Problems,” may contain material that corrects or supersedes information in this manual. Insert the *Open Issues* package into this manual behind the *Open Issues* tab.

In addition, Desktop SPARCsystem owners may want to consult the appropriate Desktop SPARC *Installation Guide* and other Sun documentation during the installation.

## 1.1. Release Compatibility

SunOS 4.1.2 is fully binary compatible with SunOS 4.1 and subsequent 4.1.X releases; all applications written for SunOS 4.1, SunOS 4.1.1, or SunOS 4.1.1 Rev B will run on the SunOS 4.1.2 release without change.

### SunOS 4.1.2 Binary Compatibility with SunOS 4.1

User executables (including shared libraries), sources, and object files from SunOS 4.1, SunOS 4.1.1, or SunOS 4.1.1 Rev B should move to SunOS 4.1.2 with no change. This includes many Sun and third-party unbundled products. The rare exceptions to this are programs, whether in source, object, or executable form, that depend upon the internal implementation of the kernel and its data structures. Programs that use the *kvm* library are likely to be in this class. Although many of these programs will still run, the data they access through the library may have changed in ways that can affect the functioning of the program.

In addition, programming that modifies the implementation of the system, especially kernel extensions or modifications, may require some conversion. Device drivers that use interface routines published in the *Writing Device Drivers* manual should require no modification. However, all kernel modifications should be inspected to verify that the aspects of system implementation they rely upon have not been altered. See Appendix B for more about device drivers.

Lastly, a bug fix in the `link-editor` provided with 4.1.2 may reveal previously masked errors with incorrectly constructed shared libraries. These will manifest themselves as references to undefined symbols. See “Link-editor Bug Fixes” in Chapter 2 for more detail.

### Binary Compatibility with SunOS 4.0.X

User executables (including shared libraries), sources, and object files from 4.0.X releases should move to SunOS 4.1.2 with no change. This includes many Sun and third party unbundled products. There are exceptions to this that should, in practice, be encountered rarely. These are programs (in any of source, object, or executable forms) that:

- Depend upon the internal implementation of the kernel and its data structures;
- Depend upon the object file format in a way that can be confused by extra information now appearing at the end of some object files in support of `#ident` directives in C source programs;
- Rely upon implementation attributes, such as the location, existence, or format of certain files not documented as part of the system’s programming interface (such as temporary files created by library functions or utilities).

Programs that use the `kvm` library are likely to be in the first class of programs described above; although the programs may still function, the data they access through the library may have changed in ways that will affect the function of the program. Examples of such data include the `proc` and `user` structures of the system.

Further, programming that modifies the implementation of the system, in particular kernel extensions or modifications, may require some conversion. Most drivers should work simply through recompilation, although other kinds of changes should be inspected to verify that whatever aspect of the system implementation they rely upon for their operation has not been altered.

Note that while it is possible, as always, to move earlier programs (their sources, objects, and executables) forward to SunOS 4.1.2, it is **not** possible to move SunOS 4.1, 4.1.1, 4.1.1 Rev B, or 4.1.2 programs back to SunOS 4.0.X or earlier releases. In SunOS 4.1, 4.1.1, 4.1.1 Rev B, and 4.1.2, this extends to newly created filesystems, which incorporate updates from the Berkeley 4.3 BSD “Tahoe” release that change the on-disk format of the filesystem. Such filesystems cannot be directly connected to pre-4.1 systems, though network access to them is unaffected by this change, and tapes made via `dump` and `tar` and similar utilities are also unaffected.

## 1.2. Workstation and Machine Architecture Terminology

Since SunOS 4.0.3, an important distinction has been made between the *application architecture* and the *kernel architecture* of a machine. In this and other documents, the distinction is frequently used in discussing workstation features and installation procedures.

### Application and Kernel Architecture

Every Sun system has an application architecture and a kernel architecture. *Application architecture* refers to the way in which systems interpret the binary code generated by application software. Two systems have the same application architecture if they can both run the same application *binaries*. *Kernel architecture* refers to the hardware-specific portion of a SunOS kernel. Two systems have the same kernel architecture if the same SunOS kernel will run on both of them.

SunOS 4.1.2 supports only the *sun4* application architecture (Sun-4, Sun-4c, Sun-4m). (Application and kernel architectures are identified by “sun” in lower case letters followed without a space by “3” or “4” and, optionally, a letter — “sun4c,” “sun4m,” “sun3x”.) Sun-4, Sun-4c, or Sun-4m servers running 4.1.2 will fully support Sun-3 clients running SunOS 4.1, 4.1.1, or 4.1.1 Rev B.

All systems with *sun4* application architecture are built around Sun’s SPARC processor. All SPARCsystems can run the same application binaries. Similarly, application binaries that run on one *sun3* system will run on all other *sun3s*. However, application binaries generated on a SPARCsystem will not run on a *sun3* system.

Not all systems with *sun4 application architecture* have the same *kernel architecture*. For example, the SunOS kernel that runs on a SPARCstation 1 or SPARCstation 2 will not run on a SPARCstation 470 nor SPARCserver 490, nor will it run on SPARCsystem 600MP series machine. Both the SPARCstation 1 and SPARCstation 2 have *sun4c* kernel architecture. Systems such as the SPARCserver 490 and SPARCstation 470 have *sun4* kernel architecture, and SPARCsystem 600MP series machines have *sun4m* kernel architecture.

There are also two separate kernel architectures corresponding to *sun3* application architecture: *sun3x* kernel architecture and *sun3* kernel architecture.

In commands and screen examples, this and other documents may refer to machine architectures as *a-arch* (application architecture) and *k-arch* (kernel architecture). These are often combined as *a-arch.k-arch* pairs, for example: *sun4.sun4c*. The following table shows application and kernel architectures for Sun systems.

Table 1-1 *System Architectures and Machine Names*

Machine Name	a-arch	k-arch
Sun-4/100 & Sun-4/200 series SPARCsystem 300 and 400 series	sun4	sun4
SPARCstation 1, SPARCstation 1+, SPARCstation 2, SPARCstation IPC, SPARCstation IPX SPARCstation SLC, SPARCstation ELC SPARCserver 1, SPARCserver 1+ SPARCserver 2, SPARCserver IPC, SPARCserver SLC	sun4	sun4c
SPARCsystem 600MP series	sun4	sun4m
Sun-3/50, Sun-3/60, Sun-3/75, Sun-3/100 series, Sun-3/200 series	sun3	sun3
Sun-3/80, Sun-3/400 series	sun3	sun3x

### 1.3. Getting Help

If you have problems installing or using SunOS 4.1.2, call Sun Microsystems with the information outlined below. In the United States you can call 1-800-USA-4-SUN; outside the U.S., contact your local Sun Answer Center or your Sun sales representative for assistance.

#### **You Will Need to Provide the Following Information:**

- Your name and electronic mail address (if any)
- Your company name, address, and phone number
- The model and serial number of your workstation
- Additional information provided by the `showrev(8)` command (described in Chapter 2 of the *Release Manual* and in the `man` page for `showrev(8)`.
  - your system's hostname
  - hostid
  - kernel and application architecture
  - SunOS release number
  - kernel revision
- Any information that may help to diagnose the problem.

Call your sales representative if you have questions about Sun support services or your shipment.

## 1.4. Guide to Publications

If you have a new Sun system or if you have been running a release prior to SunOS 4.1, you should have available the full SunOS 4.1.X set of Sun software manuals. Two boxes in the set are of particular importance for installing and running this release: the SunOS Release Minibox and the OpenWindows End User's Manuals.

Note that the Release Minibox is the only box of the full SunOS 4.1.X set that changed between SunOS 4.1.1 and SunOS 4.1.1 Rev B, and again between SunOS 4.1.1 Rev B and SunOS 4.1.2. In addition to the *Release Manual*, the Release Minibox contains *Installing SunOS 4.1.2 System Software*, with detailed instructions for performing a custom installation of SunOS 4.1.2, a quick installation of a predefined version of SunOS 4.1.2 on a standalone workstation, or an upgrade of a SunOS 4.1.1 or 4.1.1 Rev B system to SunOS 4.1.2.

For a description of the manuals provided in the full SunOS 4.1.X set, other than those mentioned above, see Appendix A.

## 1.5. Documentation Conventions

The following documentation conventions are used in this manual.

### `.cshrc` Could Be `.profile`

In most cases, whenever `.cshrc` is mentioned in this manual, `.profile` may be substituted if you are using the Bourne shell.

### Type Styles

This manual uses different type styles, or fonts, to distinguish between information you type from the keyboard, information the system displays, and variables (items that must be replaced by a value).

- `listing`  
Represents a system response that is displayed on your monitor or terminal screen. Listing font is also used for literal values (such as the names of files or utility programs) and for file listings and session output.
- **bold listing**  
Represents characters you type from the keyboard.
- *italic*  
Represents variables, for which you must substitute values. Also used for emphasis, particularly when introducing new terms.

### Boxes

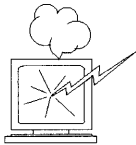
Boxes are used to display information as it appears on your video display.

- Unshaded boxes represent screen displays, system responses, or file listings. They do not contain entries or responses made by a user.

```
standard daemons: update cron.  
starting network daemons: inetd printer.  
Sat Mar 17 13:40:15 PDT 1990  
  
gemini login:
```

- Gray boxes represent a dialog between you and your workstation. Your entries are shown in **bold listing font**, system responses are shown in regular listing font.

```
Enter name of client machine: opusone  
Enter name of server: fourfox
```



- **Exploding Workstation**

Marks information that it is *imperative* for you to read. If you skip over this material, you may damage your system or jeopardize the installation.



- **Pointing Hand**

Points to information that requires close attention.

---

## Part 2 — Features

Part 2 tells you what is new about SunOS 4.1.2. It includes the following chapters:

- Chapter 2, “New Software Features”
- Chapter 3, “About OpenWindows”
- Chapter 4, “New Hardware Features”
- Chapter 5, “Open Boot PROM 2.0 and Hierarchical Device Naming”
- Chapter 6, “Bugs Fixed in SunOS 4.1.2”







---

## New Software Features

SunOS 4.1.2 includes the following software enhancements:

- SPARCsystem 600MP series systems support
- Software upgrade utility (described in *Installing SunOS 4.1.2 System Software*)
- Support for OpenWindows Version 2 and Version 3 (described in Chapter 3)
- Support for a number of graphics devices
- Quicker file system checks during reboot
- Support for Rock Ridge file system on a CD-ROM
- Availability of BSD 4.3 `malloc`
- `link-editor` bug fixes
- 200+ bug fixes, including fixes for graphics devices

### 2.1. General Software Changes

Software changes other than those supporting SPARCsystem 600MP series systems are described below; SPARCsystem 600MP series changes are described in the following section.

#### Sunupgrade for Upgrading to SunOS 4.1.2

SunOS 4.1.2 includes `sunupgrade` for upgrading from SunOS 4.1.1 or SunOS 4.1.1 Rev B. See "Upgrading to SunOS 4.1.2" in *Installing SunOS 4.1.2 System Software* for a complete description of the upgrade procedure.

#### Graphics Support and Fixes

SunOS 4.1.2 incorporates all functionality and fixes from the 4.1.1-GFX Rev.1 and 4.1.1-GFX Rev.2 releases. This includes Sun-4c support for the GS, GT, and GXplus graphics accelerator boards and the TC frame buffer, plus Sun-4 support for the previously-unbundled VX graphics accelerator board, and Sun-4m support for the single-slot GX graphics accelerator and MG1 frame buffer.

#### Quicker File System Checks

SunOS 4.1.2 includes the functionality of Sun QuickCheck 2.0, which was formerly an unbundled product. Basically, `fsck` runs more quickly because the code has been rewritten to speed up file system checking, and because file systems are checked periodically (usually once every 30 seconds), and those marked as "clean" or "stable" are not checked when `fsck` is run in "preen" mode, and during reboot. The superblock of each file system contains a "clean flag" that is set

whenever the file system is known to be in a stable state (following a sync, for example).

See the man pages for `fsck(8)`, `fstab(5)`, `dumpefs(8)`, and `quotacheck(8)` for details.

### Rock Ridge File System Supported on CD-ROM

Sun's `hfsfs` file system (ISO 9660) now supports the Rock Ridge file system extensions. This will allow any Rock Ridge CD-ROM to look like full UNIX file systems, with long file names, case sensitivity, symbolic links, Block and Character devices, and other UNIX file system features. The only file system extension it does not support is writing to the CD-ROM file system itself.

### BSD 4.3 malloc Available

By default, when `malloc()` is invoked, the Sun version, found in `libc`, is run. This version of `malloc()` is more efficient in space than the BSD 4.3 version of the command, but it is not as efficient in speed.

The BSD 4.3 version of `malloc()`, `realloc()` and `free()` which are optimized for speed rather than space, are available in `/usr/lib/libbsdmalloc.a`. For programs where memory usage is less of an issue than execution speed, use the BSD 4.3 versions of `malloc()`, `realloc()` and `free()` by explicitly compiling your programs with the `-Bstatic -libsdmalloc` flags.

The `malloc()`, `realloc()`, and `free()` routines in `libbsdmalloc.a` are defined like the `libc` versions:

```
char *malloc(size)
unsigned size;

int free(ptr)
char *ptr;

char *realloc(ptr, size)
char *ptr;
```



Note the following caveats about `libbsdmalloc`:

- Compiled binaries will not be SCD/ABI-compliant (because `sbrk()`, which is not in the ABI, is called)
- Other `malloc`-related routines defined in 4.X, such as `calloc()`, `memalign`, `mallopt()`, and `mallinfo()`, will not work with `libbsdmalloc`.

## 2.2. Link-editor Bug Fixes

A number of bugs have been fixed in both the static and dynamic link editors. All of these fixes have been previously available in patches to `/usr/bin/ld` and `/usr/lib/ld.so`; the SunOS 4.1.2 editions of these completely supersede all SunOS 4.1 and SunOS 4.1.1 patches. See Chapter 7 for instructions for installing language patches.

In almost all cases, these changes simply cause the loader(s) to perform in accordance with the behaviors specified in the manuals; there are no enhancements or other functional changes as a result of these repairs.

However, one of the repairs corrects a problem (Bug ID 1019004) and reveals bugs in other parts of the system — bugs previously masked by the incorrect behavior of the loader. Another repair (to Bug ID 1052428) restricts the behavior of the dynamic loader library search algorithm for `set-user-` or `set-group-id` programs. Both of these are discussed in more detail below.

As described above, the fix repairing bug 1019004 ("assert definitions can fail to report undefined symbols") can reveal other problems previously masked by the bug. Specifically, this bug allowed the construction of programs that silently permitted inclusion of shared libraries having unresolved references. The bug fix makes the link editing of programs using such libraries now report errors where previously the link editing operation appeared to have completed successfully.

In many cases, particularly in the case of affected libraries in Sun products, the unresolved references were never actually used. Thus no real problem ever appeared, even though such libraries are, strictly speaking, incorrectly constructed. However in other cases, such as for users reporting 1019004, the failure to report the unresolved reference permitted users to believe they had correct programs, only to find out that they were not correct when, during execution, the dynamic link editor would suddenly report the unresolved reference.

This can manifest itself with various Sun libraries in the `-Bdynamic` binding mode (the default), as follows:

- Sun FORTRAN: if a program uses the `libF77` library, but is not built as a FORTRAN program (i.e., not linked with an `f77` command), you can expect to see an undefined symbol reference for `MAIN_`.
- Sun Pascal: if a program uses the Pascal library, but is not built as a Pascal program (i.e., not linked with a `pc` command), you can expect to see an undefined symbol reference for `program`.
- Sun PHIGS: you can expect to receive an undefined symbol reference to `phg_sin_set_no_input_cursor`.
- Sun Modula-2: If a Modula-2 program is linked dynamically it requires that `libtermcap` also be linked.

Note that using a FORTRAN or Pascal `main` program will not produce diagnostics; this will happen only in the special cases of using their libraries from (for example) a C `main` program.

In all cases where the error occurs, you can safely ignore the error if it is the only undefined symbol reported. Your executable will be marked as "executable", but the linker will return a non-zero error, thus causing makefiles to terminate abnormally. A workaround for this problem is to define the undefined symbol in your application. For example, if your C program links with the Fortran library, it will now have an undefined symbol of `MAIN_`. In your C program you could define a subroutine:

```
MAIN_ (
{
} .
```

This would satisfy the undefined external reference. The same could be done for Pascal by defining a routine:

```
program (
{
}
```

If you do not want to use the Sun FORTRAN or Sun Pascal workaround, updated versions of the Fortran and Pascal libraries are supplied with your SunOS 4.1.2 distribution and should be applied after your upgrade (or any new installation or re-installation of 4.1.2 occurs). See Section 7.15 in Chapter 7 for more information.

Undefined external references may also occur with user and third-party supplied shared libraries. In some of these cases, the undefined symbols may have no practical value — as the program may never actually use the code paths that reference the undefined symbol (this is the case with the Sun-supplied libraries.) In this event, the diagnostic from the loader may be ignored and the program executed anyway (unless there is some other diagnostic that prevents completion of link editing.) Note, however, that if a code path accessing any undefined symbol is actually used, the dynamic link editor will abort the program at the time the symbol is referenced (or immediately upon start-up if the reference is from a data structure.)

In the case of bug 1052428 (“ld.so usage of -L options confusing, leads to security probs”), the dynamic loader’s usage of stored search paths specified by “-L” options to ld has been restricted to the use of absolute path names (names beginning with “/”) when running a set-user- or set-group-ID program.

Summary of bugs fixed in /usr/bin/ld:

- 1019004 assert definitions can fail to report undefined symbols
- 1032739 ld core dumps with many libs in multiple directories
- 1034788 -r and -pic do not mix bad secondary magic number error
- 1034833 ld: can’t mixed -r with -Bstatic or -A flag
- 1037879 Cannot create executable with shared object which points to another shared object.
- 1041946 (duplicated by 1065275, below).
- 1042261 ld only recognized first directory in LD\_LIBRARY\_PATH
- 1044524 multiply defined symbols and seg. fault caused by 4.1’s ld
- 1045272 ld -u & -r do not seem to work properly
- 1046462 ld fails with segmentation fault when processing nested #include files
- 1050594 fix uninitialized structure entries.
- 1064820 ld can produce bogus diagnostics when processing cascaded

dependencies

1065275 program dumps core with shared objects having only statics.

Summary of bugs fixed in `/usr/lib/ld.so`:

1033086 Calling `dlopen()` on a file with BSS kills your application.

1043300 `dlclose` prevents access to functions even after further `dlopen`

1045194 `dlsym` returns bad address for uninitialized global variable

1046379 cannot call a function in a `[.so]` from a function in another

1052428 `ld.so` usage of `-L` options confusing, leads to security probs

### 2.3. SPARCsystem 600MP Systems

This section has two subsections: the first subsection lists and describes new SPARCsystem 600MP hardware support and software features; the second subsection briefly describes the SPARCsystem 600MP multiprocessing environment.

#### Description of New Features

- New commands (`mpstat(1)` and `mps(1)`).

The command `mpstat` shows average and per-processor percentage usage data during a time interval. The command `mps` displays the status of current processes on an MP system. See the man pages for these commands for details.

- Extended SCSI device support for SPARCsystem 600MP series

The SunOS 4.1.2 GENERIC sun4m kernel configuration file declares support for five SCSI buses.

- Open boot PROM support for SPARCsystem 600MP series machines

See Chapter 5 for a discussion of the open boot PROM and its use.

- New version of Sundiag

The latest version of Sundiag, version 2.3, is included in SunOS 4.1.2. Sundiag 2.3 features a new multiprocessor test, `mptest`, for validating SPARCsystem 600MP multiprocessing hardware. See Appendix C for a description of the new features and enhancements of Sundiag 2.3.

Note that Sundiag 2.3 is SunView-based; it will not run in OpenWindows. If OpenWindows is running, you must quit the window system. You can then run Sundiag from the monitor prompt (no window system) or you can run Sundiag from the SunView window system.

- New kernel configuration files for SPARCsystem 600MP series systems:

The following kernel configuration files are available for various system configurations:

- DL: diskless Sun-4m system
- NFS: Sun-4m with SCSI disks and tape; boot from server
- SDST: Standalone Sun-4m with SCSI disks and tape
- IDST: Sun-4m with IPI disks and SCSI tape

## The SPARCsystem 600MP Multiprocessing Environment

SunOS 4.1.2 has been modified and tuned to function on multiprocessor (MP) hardware. The changes made to the kernel take advantage of the multiprocessing capability, yet provide complete binary compatibility with SunOS 4.1, 4.1.1, and SunOS 4.1.1 Rev B. The interfaces provided to the user in the form of libraries, system calls, and commands have not changed from SunOS 4.1.1 Rev B; two commands — `mpstat(1)` and `mps(1)` — have been added. Systems will automatically have increased throughput from the MP hardware, under appropriate applications.

### How MP Works

The 600MP series uses a tightly-coupled, shared-memory configuration model. Several CPUs share real memory, but all processes have their own virtual address space and the kernel memory management system controls virtual-to-real address translation. Only one copy of the operating system exists, and is shared by all the processors. Interrupts can be serviced by any processor, and processors can send each other interrupt signals.

### The Operating System

Processors accessing kernel data structures and code do so after acquiring locks that prevent other processors from accessing the same data structures or executing the same code concurrently. Various optimizations have been made to minimize the overhead due to synchronization.

### Application Programs

Application programs running in user mode have their own virtual address space and do not have to worry about shared memory synchronization. Several such applications can run concurrently on separate processors. In these cases additional processors will provide almost linear increases in the system's overall throughput.

Applications that enter the kernel to utilize system resources will have to go through the additional overhead of synchronization; therefore, a mix of applications that spends a large proportion of its time executing in the kernel will have lower throughput than a mix of applications that spends a small proportion of its time in the kernel.

## 2.4. Writing Device Drivers for SPARCsystem 600MP Systems

See Appendix B for information about writing device drivers for SPARCsystem 600MP systems.

---

## About OpenWindows

OpenWindows Version 2 is the default window system for SunOS 4.1.2, and is included on the SunOS 4.1.2 release CD; pre-installed systems with adequate disk space will be installed with OpenWindows Version 2.

OpenWindows Version 3 is co-packaged with SunOS 4.1.2; it is on a separate CD in the media box, along with the XGL 2.0 RTE software.

This chapter includes information about OpenWindows Version 2 and OpenWindows Version 3. See Chapter 10 in the *Open Issues* packet in the Release Mini-box for known problems with Versions 2 and 3.

### 3.1. Why Use OpenWindows Version 3?

Consider switching to OpenWindows Version 3 if the following new features and functions are important to you:

- 24-bit graphics support for sophisticated color applications
- Fast, direct access to hardware for graphics-intensive applications
- 8-bit internationalization capability
- Enhanced DeskSet 3.0
- ToolTalk, a new state of the art tool for inter-application communication
- Improved application integration through new network-transparent drag-and-drop mechanism
- 3D look and feel
- Mouseless support
- Audio capability in deskset environment
- Easy migration to SVR4

See the following documents to learn how to install and use OpenWindows Version 3:

- *OpenWindows Version 3 Installation and Start-Up Guide* (PN 800-6029-10)
- *CDmanager and CDM 2.0 User's Guide* (PN 800-6174-10)

### 3.2. General OpenWindows Features

OpenWindows includes the following features:

- X11/NeWS, a merger of two independent window system protocols
- XView Toolkit
- NeWS Toolkit (TNT)
- The OPEN LOOK user interface
- OPEN LOOK Intrinsic Toolkit (OLIT)
- OpenWindows DeskSet Environment

#### X Window System

X11 is version 11 of the X Window System developed at MIT. The X Window System is the industry standard for a network-based, operating-system independent window system.

#### NeWS

NeWS is the “Network extensible Window System” developed by Sun and originally offered as a separate window system.

#### XView Toolkit

The XView Toolkit is a development toolkit that goes with the X11 side of X11/NeWS. It is used to produce X11 applications based on the *OPEN LOOK* graphical user interface.

#### NeWS Toolkit

The NeWS Toolkit is a new toolkit for NeWS programmers.

#### OPEN LOOK Interface

The OPEN LOOK interface is a set of specifications that define consistent visual and operational features; it gives a uniform look and feel to all OpenWindows applications.

#### OPEN LOOK Intrinsic Toolkit

The OPEN LOOK Intrinsic Toolkit is a user interface toolkit based on MIT’s *Xt Intrinsic*.

#### OpenWindows DeskSet Environment

The OpenWindows DeskSet Environment provides mouse-oriented and icon-oriented tools in the OPEN LOOK for standard functions and services such as:

- File management (opening, closing, copying, moving files, etc.)
- E-mail
- System and CPU performance monitoring
- Calendar display and scheduling
- Printing files and mail messages



### 3.3. About OpenWindows Version 2

Note the following about OpenWindows Version 2. See Chapter 10 for a listing and descriptions of known problems with OpenWindows Version 2.

#### Do Not Install Sun IPX Supplement

SunOS 4.1.2 OpenWindows Version 2 includes the Sun IPX supplement; do not install this patch after installing SunOS 4.1.2.

#### Keyboard Map File for Japanese International Keyboard

The keyboard map file for the Japanese international keyboard is `Japan4.ps`. See Chapter 3 of the *OpenWindows Installation and Start-Up Guide* for more information.

#### Using SunView for Maximum Security Levels

If you require maximum levels of security, use the SunView window system. SunView, which is a kernel-based window system, presents a more secure environment than a networked window system such as OpenWindows. (Refer to the *OpenWindows Version 2 Release Notes*, page 13, for a detailed description of OpenWindows security.)

The features and use of OpenWindows are fully described in the manuals in the OpenWindows End User's Kits provided with the SunOS 4.1.2 Release Minibox (SX-9E). If you plan to use OpenWindows, you should make sure that you have this documentation.

### 3.4. Which Default Window System?

Depending on how your system is set up, one of the following scenarios will be played out when you log in:

- New user logging in for the first time; OpenWindows comes up as the default window system
- New user logging in for the first time; SunView comes up as the default window system
- New user logging in for the first time; system offers choice of SunView or OpenWindows as default system
- New user logging in for the first time; no default window system. User receives a shell prompt after logging in
- Old user logging in for the first time under this new operating system; old home directory was backed up and restarted. User gets same window system he/she previously had

Each situation is covered separately in the following sections.

#### New User; OpenWindows Comes Up as Default

If you are a new user on a SunOS 4.1.2 system and the "openwin" executable can be found, OpenWindows Version 2 will be your default window system. Note that this setting occurs in the `.cshrc` (or `.profile`) file in your home directory. Note also that the `desktop` program, described below, can easily change your default window system.

When you log in, you will see the message:

```
starting OpenWindows (Control-C to interrupt)
```

If you do not interrupt with **Control-C**, OpenWindows will come up after a short delay.

You should now check to make sure that the OpenWindows Calendar Manager has been initialized:

- Display the Workspace Programs menu by moving the mouse cursor to a screen background area and pressing the right mouse button
- Drag the cursor to the right edge of the Programs option to display the Programs submenu
- Select the Calendar Manager

If the Calendar Manager has not been initialized, at the bottom of your screen, below the calendar, you will see the error message:

```
rpc.cmsd is not responding...have you run install_cmgr?
```

If you do not see the message, **do not** run `install_cmgr`. If it has been run previously, running it again causes problems. If the message is displayed, you need to take the following steps:

- Become superuser and run `install_cmgr`:

```
%su
Password: [enter root password]
#/usr/openwin/bin/xview/install_cmgr
```

- Exit OpenWindows
- Log back in to your system and let OpenWindows come up again

### New User; SunView Comes Up as Default

If you are a new user on a SunOS 4.1.2 system and the `openwin` executable cannot be found, SunView will be your default window system. Note that this setting occurs in the `.cshrc` (or `.profile`) file in your home directory. Note also that the `desktop` program, described below, can easily change your default window system.

### New User; No Default Window System

If you are a new user but your account was not set up with the `add_user` program or by copying the default `.cshrc` (or `.profile`) and `.login` files into your home directory, you will not have a default window system; you will have a shell prompt when you log in.

### Old User; Previous Window System

If you are an old user logging in for the first time under this new operating system, and you have re-installed the home directory you had been using before installation of the new operating system, you will have the same window system you previously had.

### 3.5. Changing the Window System to be Invoked upon Login

With SunOS 4.1.2 you can change the window system invoked upon login by using the `desktop(1)` command. Note that this applies if you are using the default `.cshrc` file (from `/usr/lib/Cshrc`) and if you have modified the section of code regarding window-system invocation.

```
% desktop openwin
```

will change it to OpenWindows.

```
% desktop sunview
```

will change it to SunView.

### 3.6. Online Desktop Tutorial

An online tutorial, *Introducing Your Sun Desktop*, is provided in this release. The introduction is part of the default desktop and introduces you to some basic OpenWindows principles. It includes two short cookbook walkthroughs explaining how to copy, cut, and paste text, and how to use File Manager. Besides providing you with some basic skill training, the introduction allows you to start some of the deskset utilities, test your SPARCstation's audio capabilities, and select between German, French, and English versions of the introduction.

You can access the online tutorial at any time when running OpenWindows by selecting it from the Workspace Programs menu. If you'd like to get spot help in French and German for the File Manager and Mail Tool, see page 36 of *Introducing Your Sun Desktop*.

### 3.7. OpenWindows Online Tutorial

This online tutorial provides instruction on all the basic skills necessary for becoming a capable end-user of Sun Microsystems' OpenWindows software. In addition to the easy-to-follow explanations, the tutorial offers demonstrations or simple animated examples of how the software works.

The demonstrations are designed to display the software in detail and first-time users may find them especially helpful. Because the demonstrations' animated scripts create a separate environment, you have to turn off system security in order to run them.

You can, however, run the tutorial without accessing the demonstrations. This allows you to call up the tutorial for reference at anytime you are running OpenWindows and need a quick reminder.

There are two ways to view the online tutorial:

- With animated demonstrations

- Without the animated demonstrations



The animated demonstrations run on Sun's standard monochrome and color monitors (1152 X 900 pixels). They do not run correctly on the high resolution monitor (1600 X 1280 pixels). If you attempt to view an animated demonstration while running OpenWindows on a high-resolution monochrome monitor, an alert to this effect appears on the screen.

You should also be aware that the demonstrations included with your online tutorial were designed to work using OpenWindows' default settings. It is possible to change these settings (as the online tutorial itself explains), but if you do, the animated demonstrations may not perform correctly.



**Do not move the mouse or press any mouse buttons or keyboard keys while a demonstration plays. This will interrupt the demonstration and can cause your system to crash.**

### *Accessing the Online Tutorial with Animated Demonstrations*

To access the online tutorial with animated demonstrations:

1. Exit OpenWindows.

To exit OpenWindows, press and hold the right mouse button anywhere in the background of your screen (wherever there is no window or icon). A Workspace menu appears. Still holding the right mouse button down, drag down, highlight `Exit`, and release the mouse button.

2. At the `%` prompt on a full screen without windows, enter:

```
%/usr/openwin/demo/tutorial/start_tutorial
```

The online tutorial appears on your screen with a Console window displayed nearby. Do not quit the Console window; it is there to accept system messages for you.

Proceed to the tutorial section titled "Introduction-Read This First" for instructions on using the online tutorial. To do this, click the left mouse button twice in rapid succession on the screen button marked "GoTo" beside the words "Introduction-Read This First."

### *Returning to OpenWindows*

To return to OpenWindows when you finish the online tutorial:

1. Exit OpenWindows (as described in the previous section).
2. At the prompt enter:

```
%/usr/openwin/bin/openwin
```

### *Accessing the Online Tutorial without Animated Demonstrations*

To access the online tutorial without animated demonstrations, from within OpenWindows enter:

```
%/usr/openwin/demo/tutorial/tutorial.sh
```

An alert appears on screen:

```
Security is on.
You must start server with either:
  openwin/demo/tutorial/start_tutorial
  openwin/bin/openwin -defeateventsecurity
```

This is to remind you that *security* is on and animated demonstrations are not available. As long as you continue in the current mode (without demonstrations), a similar alert appears whenever you attempt to call up an animated demonstration.

We recommend that you operate the online tutorial without animated demonstrations whenever you want a quick refresher on OpenWindows features, tools, utilities, or functions.

### *Moving Around in the Tutorial*

When the tutorial comes up on your screen, the first thing you see is the Table of Contents. Next to each topic is a “GoTo” button. To learn about any topic or feature listed in the Table of Contents, position your mouse cursor over the GoTo button next to the topic you want and double-click using SELECT (the left mouse button). This takes you to the first page of the selected section of the tutorial.

### *Closing the Tutorial*

When you are done with the tutorial, or if you want to take a break, you can close the online tutorial window to an icon on your screen.

To close the online tutorial window, put your cursor in the border of the window. Press and hold MENU (right mouse button) and a pop-up window appears. Pull your cursor down to highlight the Close option and release the mouse button. The online tutorial is then suspended and a small icon appears on your monitor screen.

There are two ways to reopen the online tutorial from its icon. You can double-click SELECT (left mouse button) on the icon and the online tutorial re-opens for use. You can also press and hold the right mouse button (MENU) over the icon; when the menu appears, highlight the Open option and release the mouse button.

*Quitting the Online Tutorial*

Instead of closing the online tutorial, you can quit and exit the application entirely. To quit the online tutorial, position your cursor either in the tutorial window border or over the tutorial icon. Call up the window menu by pressing and holding MENU (right mouse button) and then highlight the Quit option. The tutorial window (or the icon) then disappears from the screen.

---

## New Hardware Features

SunOS 4.1.2 includes kernel or device support for the following new hardware products and features:

- 1.3GB SCSI disk drive
- GT graphics accelerator (Sun-4c)
- GS graphics accelerator (Sun-4c)
- GXplus graphics accelerator (Sun-4c)
- TC-compatible frame buffer (Sun-4c)
- VX graphics accelerator (Sun-4)
- Support for SPARCsystem 600MP series systems, including:
  - Up to five SCSI buses
  - Open boot PROM support
  - Sbus and Mbus expansion buses
  - Improved network connections
  - Various enclosure configurations
  - CPU board bus and port options
  - CPU board memory options
  - Monitor options
  - Printer interface options
  - single-slot GX graphics accelerator
  - MG1 frame buffer

### 4.1. Peripheral Support

SunOS 4.1.2 adds support for the new 1.3GB, 5 1/4-inch SCSI disk drive, a high-performance drive that incorporates Zone Bit Recording (ZBR), a method of recording that divides the surface of the disk into different zones, each with a different number of bytes per track. This keeps the bit density constant across the surface of the disk, thus achieving higher performance and capacity.

Note that the patch release titled *1.3 GB Disk Drive Enhancement*, for supporting this drive under SunOS 4.1 and SunOS 4.1.1, is not required; the part numbers for this patch were:

704-2948-11 (CD-ROM)  
700-2948-11 (tape)

## 4.2. Graphics Support

SunOS 4.1.2 also incorporates all functionality and fixes from the 4.1.1-GFX Rev.1 and 4.1.1-GFX Rev.2 releases. This includes Sun-4c support for the GS, GT, and GXplus graphics accelerator boards and the TC-compatible frame buffer, Sun-4 support for the VX graphics accelerator board, and Sun-4m support for the single-slot GX graphics accelerator. All of these graphics devices except for the VX graphics accelerator were previously supported via GFX or other feature releases; the VX graphics accelerator was an unbundled device.

### GXplus Graphics Accelerator

The GXplus is a memory enhanced version of the familiar double-wide Sbus GX card (also known as a `cg6`). It has four Mbytes of video RAM compared to one Mbyte on the standard GX. This enables the GXplus to support double buffering for smoother dynamic graphics, and fast retained windows for speedy damage repair. It also supports 1280 x 1024 resolution. GXplus is 100% backward-compatible with GX application software.

### GS Graphics Accelerator

The GS is an integrated frame buffer and graphics accelerator that features 24-bit true color, accelerates the rendering of 3D solids, and also accelerates window system performance; it is also known as a `cg12`. Additional features include:

- 24-bit double buffering (transparently dithered by hardware to 12-bit)
- 16-bit Z-buffer
- Gouraud shading using up to 8 colored light sources simultaneously
- Hardware picking and BitBlt
- Support for 76 Hz. monitor
- Separate cursor planes (in OpenWindows Version 3)
- Multiple rendering modes, allowing 1-bit, 8-bit, and 24-bit applications to be run and displayed simultaneously
- Support for SunView software application programmer interfaces (APIs), including Pixwin (but not Pixrects), and SunPHIGS 1.2, 1.3, 1.4, and later
- Support for OpenWindows Version 3 software application programmer interfaces (APIs) and toolkits, including SunPHIGS 2.0, XGL 2.0, GKS 4.1, XView, OLIT, TNT, and Xlib

The SunView window system is fully supported by the GS. OpenWindows Version 2 runs on the GS in monochrome overlay only. OpenWindows Version 3 on the GS supports visuals in 4-bit, 8-bit, and 24-bit plane groups.



**GT Graphics Accelerator**

The SPARCstation GT is a high-performance 24-bit true color graphics accelerator that provides high-quality advanced 2D and 3D accelerated graphics features. Features include:

- 8-bit and 24-bit double buffering
- 24-bit Z-buffer
- 8-bit double-buffered overlays
- 10 window ID planes
- four double-buffered, fast clear planes
- Separate cursor planes (in OpenWindows Version 3)
- Line, polygon, and stroke text anti-aliasing
- Support for OpenWindows Version 3 software application programmer interfaces (APIs) and toolkits, including SunPHIGS 2.0, XGL 2.0, GKS 4.1, XView, OLIT, TNT, and Xlib

The SunView window system is fully supported by the GT. OpenWindows Version 2 runs on the GT in monochrome overlay only. OpenWindows Version 3 on the GT supports visuals in 8-bit and 24-bit plane groups.

**VX Graphics Accelerator**

The VX is one of Sun's second generation visualization accelerators for Sun SPARCsystems. The VX, a successor to the TAAC-1, includes a high performance, floating point processor, a 16 MByte frame buffer, and a GX-accelerated system frame buffer.

**TC-Compatible Frame Buffer**

The TC is a basic "dumb" 24-bit true color frame buffer with an additional 2-bit overlay/enable plane group; it is also known as a cg8.

Under SunView, the TC supports 1-bit, 8-bit, and 24-bit plane groups. The user is cautioned that the 8-bit support is achieved by software simulation of an 8-bit mode within the 24-bit plane group; 8-bit performance is less than that found with a typical 8-bit frame buffer.

Under OpenWindows Version 3, the TC supports 1-bit and 24-bit visuals, with no 8-bit support. The 1-bit plane group is the default visual; if you wish, you can specify the 24-bit TrueColor to be the default visual at the time you start OpenWindows. See the *OpenWindows Programmer's Guide* for more information.

**4.3. Support for SPARCsystem 600MP Series**

SunOS 4.1.2 supports SPARCsystem 600MP series systems. SPARCsystem 600MP series systems, which are available in two and four processor configurations, offer multiprocessing to improve compute performance, plus network and I/O enhancements that improve server performance. Multiprocessing provides scalable CPU performance that can cover a wide range of customer needs and significantly extend the top system performance beyond existing Sun systems. All 600MP series systems use a tightly-coupled, shared memory multiprocessing mode and high-bandwidth MBus for the primary connection between the CPUs and memory. The processors are located on module cards that

plug into MBus connectors on the CPU board, offering easy upgrading to additional processors or future SPARC technology.

### **Support for Multiple SCSI Buses on SPARCsystem 600MP**

The SunOS 4.1.2 GENERIC sun4m kernel configuration file declares support for five SCSI buses and the devices that can be connected to them. The GENERIC\_SMALL kernel configuration file declares two SCSI buses and the devices that go with them.

### **Open Boot PROM**

The open boot PROM supports booting of a SPARCsystem 600MP series system. In most standard configurations, when you install SBE/S cards on the CPU board the devices are recognized automatically. If you have a non-standard configuration, you need to reconfigure the kernel. See Chapter 5, "Open Boot PROM 2.0 and Hierarchical Device Naming" for detailed information about the open boot PROM and how to boot a SPARCsystem.

### **Sbus and Mbus Expansion Buses**

Two different expansion buses provide the benefit of Sun's latest technical innovations while linking the system to established Sun products. The 600MP series systems are the first large server systems to use the SBus, which is the primary I/O bus for system services and expansion. SBus offers much higher I/O performance and uses low-cost CMOS technology for a variety of add-on options such as disk controllers and network interfaces. But, because 600MP series systems were designed to support VME boards, you can continue to use most of your VME bus options in Sun's servers.

### **Improved Network Connections**

For networking, 600MP series systems offer an on-board buffered Ethernet connection (the SBus SCSI/Buffered Ethernet Card — also called the SBE/S card) that can handle bursts of packets from high-speed client workstations better than a non-buffered connection, improving responsiveness and reducing network and server loading. Also supported is the new SBus FDDI and token ring options, which offer you tremendous flexibility in your choice of LAN technology.

Note that up to five Ethernets can now be supported on a SPARCsystem 600MP series system — one on the CPU board and one each on the four SBE/S cards.

### **Enclosure Configurations**

The 600MP series systems are packaged in one of three enclosures: the 5-Slot Office Pedestal (630MP), the 12-Slot Office Pedestal (670MP), or the 56-inch Data Center Cabinet (690MP). See the appropriate hardware manuals for details.

### **CPU Board Bus and Port Options**

The Sun 600MP CPU board operates at 40 MHz. The CPU board can have one MBus module (two processors) and four Sbus expansion slots or two Mbus modules (four processors) and three Sbus expansion slots.

The Sun 600MP CPU board also has the following ports:

- two RS232/RS423 synchronous serial ports
- Ethernet port
- SCSI port

- keyboard/mouse port
- scan port for remote diagnostic service processor
- audio port (SPARCstation 1-style audio)

#### CPU Board Memory Options

The memory on the Sun 600MP CPU board can be expanded in 16 or 64 MB SIMM increments to a maximum of 128 MB. The CPU board has 32 SIMM slots in two banks of 16. The following SIMM sets are available for insertion in these slots:

- 16 MB ECC Memory Expansion SIMMs (1 MB SIMMs)
- 64 MB ECC Memory Expansion SIMMs (4 MB SIMMs)

#### Monitor Options

The 600MP series systems are normally configured as networked servers. Customers can add a display monitor to use as console or to create a workstation configuration.

#### Graphics Support

The only supported graphics device for the SPARCsystem 600MP series are the single-slot GX graphics accelerator and the MG1 frame buffer.

#### Printer Interface Options

The 600MP series systems offer both serial and parallel interfaces. The serial interface is on the CPU board. The SBus parallel interface for the SPARCprinter is the parallel laser printer card.

#### 4.4. Related SPARCsystem 600MP Documents

The following documents contain additional information about the SPARCsystem 600MP series systems.

<b>Document</b>	<b>Part Number</b>
<i>SPARCsystem 630MP Installation Manual</i>	800-5937
<i>SPARCsystem 670MP Installation Manual</i>	800-5904
<i>SPARCsystem 690MP Installation Manual</i>	800-5935
<i>600MP CPU and Memory Board Installation &amp; Service Manual</i>	800-5318



---

## Open Boot PROM 2.0 and Hierarchical Device Naming

This chapter provides information about open boot PROM 2.0 and hierarchical device naming on all machines with open boot PROM 2.0 (SPARCstation 2, SPARCstation IPX, SPARCstation ELC, all SPARCsystem 600MP systems, and some SPARCstation IPCs).

Most users, once release software has been installed from tape or CD-ROM, only need to boot from their default boot source and never have to be concerned with either boot device aliases or hierarchical device names. For such users, the only times it is necessary to boot are when their machine has been turned off, in which case turning it on again causes it to reboot automatically, or when their system has crashed or frozen. In this case, turning the machine off and then on again automatically reboots from the default source.

The user of a desktop SPARCsystem who accidentally arrives at the PROM Monitor prompt by pressing **[L1/Stop]**–**[a]** or otherwise interrupting the system, can simply follow the screen prompt that is displayed in this situation and enter either **go**, at the **ok** prompt, or **c**, at the **>** prompt, in order to continue.

The only case in which it is necessary to enter a device name is when booting is to be from a non-default device. For many users who reach this point, the non-default device is a standard device for which there is a predefined alias (described below).

Prior to SunOS 4.1.1 and the SPARCstation 2, a disk controller purchased from manufacturers other than Sun could not be used for booting unless the manufacturer specifically designed the controller to work under a Sun boot PROM, or received code from Sun that could be incorporated in the controller. The open boot PROM and new conventions for hierarchical device naming now make it possible for a Sun Desktop SPARCsystem or a SPARCsystem 600MP system to boot from disk controllers that have not been specifically coordinated with Sun specifications.

### 5.1. Built-in Aliases for Standard Devices

All open boot PROM 2.0 systems have a set of predefined aliases for standard device names. These are adequate in the great majority of situations. The only time it should be necessary to go beyond these aliases is when you are using a newly configured type of device to boot from or when you are booting from a non-standard source.

## Switching from Restricted Monitor to Forth Monitor

There are two operational monitors available with open boot PROM 2.0 systems: the restricted monitor, which uses the `>` prompt, and the Forth monitor, which uses the `ok` prompt. You can do simple booting procedures from either prompt, but you need to be in the Forth monitor to perform more complex functions.

See the *Open Boot 2.0 Command Reference* manual (Part Number 800-6076-10) for detailed instructions for using the open boot PROM.

To switch from the `>` prompt to the Forth prompt, type `n` (for new-mode):

```
> n
ok
```

To switch from the `ok` prompt to the restricted prompt, type `old-mode`:

```
ok old-mode
>
```

## Booting Open Boot PROM 2.0 Systems from the CPU (except 690MP)

The table below shows a sample of the bootpaths that are required for booting a system with Open Boot PROM 2.0 and higher. To boot with one of the predefined aliases, type `boot` and the appropriate alias from the `ok` prompt or type `b` and the appropriate alias from the `>` prompt. Note that the SPARCsystem 690MP, which has IPI disks, has a different set of aliases for booting from disk.

Table 5-1 *Booting an Open Boot PROM 2.0 System from the CPU*

PROM Monitor Aliases for Device Path Names		
Open Boot Alias	Full Device Path	Boot From:
disk	/iommu/sbus/dma@f,81000/esp@f,80000/sd@3,0	Default disk (1st onboard)
disk0	/iommu/sbus/dma@f,81000/esp@f,80000/sd@3,0	First onboard disk
disk1	/iommu/sbus/dma@f,81000/esp@f,80000/sd@1,0	Second onboard disk
disk2	/iommu/sbus/dma@f,81000/esp@f,80000/sd@2,0	First external disk
disk3	/iommu/sbus/dma@f,81000/esp@f,80000/sd@0,0	Second external disk
tape	/iommu/sbus/dma@f,81000/esp@f,80000/st@4,0	Default tape drive (first)
tape0	/iommu/sbus/dma@f,81000/esp@f,80000/st@4,0	First tape drive
tape1	/iommu/sbus/dma@f,81000/esp@f,80000/st@5,0	Second tape drive
cdroma	/iommu/sbus/dma@f,81000/esp@f,80000/sr@6,0:a	CD-ROM partition a
cdrom	/iommu/sbus/dma@f,81000/esp@f,80000/sr@6,0:d	CD-ROM partition d(MUNIX)
net	/iommu/sbus/lebuffer@f,40000/le@f,60000	Default Ethernet

## Booting a SPARCsystem 690MP from the CPU

The table below shows a sample of the bootpaths that are required for booting a SPARCsystem 690MP with Open Boot PROM 2.0 and higher.

Table 5-2 *Booting a SPARCsystem 690MP from the CPU with Open Boot PROM 2.0*

PROM Monitor Aliases for Device Path Names		
Open Boot Alias	Full Device Path	Boot From:
ipi	/iommu/vme/idc@4d,1080000/id@0,0	Default disk (first)
ipi0	/iommu/vme/idc@4d,1080000/id@0,0	First disk drive
ipi1	/iommu/vme/idc@4d,1080000/id@1,0	Second disk drive
ipi2	/iommu/vme/idc@4d,1080000/id@2,0	Third disk drive
ipi3	/iommu/vme/idc@4d,1080000/id@3,0	Fourth disk drive
ipi4	/iommu/vme/idc@4d,1080000/id@4,0	Fifth disk drive
ipi5	/iommu/vme/idc@4d,1080000/id@5,0	Sixth disk drive
ipi6	/iommu/vme/idc@4d,1080000/id@6,0	Seventh disk drive
ipi7	/iommu/vme/idc@4d,1080000/id@7,0	Eighth disk drive
tape	/iommu/sbus/dma@f,81000/esp@f,80000/st@4,0	Default tape drive (first0)
tape0	/iommu/sbus/dma@f,81000/esp@f,80000/st@4,0	First tape drive
tape1	/iommu/sbus/dma@f,81000/esp@f,80000/st@5,0	Second tape drive
cdroma	/iommu/sbus/dma@f,81000/esp@f,80000/sr@6,0:a	CD-ROM partition a
cdrom	/iommu/sbus/dma@f,81000/esp@f,80000/sr@6,0:d	CD-ROM partition d(MUNIX)
net	/iommu/sbus/lebuffer@f,40000/le@f,60000	Default Ethernet

### Changing Default Boot Device

To change the default boot device, you need to execute the `set-defaults` command to make sure every NVRAM parameter is in its default state, and set the default boot device on the open boot PROM.

1. At the `ok` prompt, execute `set-defaults`:

```
ok set-defaults
Setting NVRAM parameters to default values.
ok
```

This command can be used to restore the NVRAM to a known state. Normally, executing this command once before or during the OS installation will be enough, but it could be used any time to get back to the ‘factory’ defaults of the NVRAM parameters.

2. Set the default boot device; for instance, to designate the first IPI disk as the boot device, do the following:

```
ok setenv boot-device ipi0
ok
```

This will ensure that the system will boot up without any user interaction after a power-on, for instance. If the OS is installed on the first disk (ID 0) of the first IPI controller, the boot device name is `ipi` or `ipi0`, as shown in the above example.

Optionally, the NVRAM parameter `diag-device`, which is used if the system is in DIAG mode, could be set to be the same as `boot-device` with the following command:

```
ok setenev diag-device ipi0
```

With this setting, auto boot will work even if someone hits the DIAG switch by mistake.

## 5.2. Hierarchical Device Naming

Devices are attached to a system on buses, with a device at one end of a bus controlling devices at the other end. This establishes a hierarchy of devices that can be represented as a tree of nodes, with each device a node and the system itself the top node. In hierarchical device naming, the name of a device is given as the path from the top of the device tree to the device itself.

Hierarchical device naming makes it possible to specify new device parameters not previously contained in a CPU's NVRAM for accessing disk devices. For example, to specify the system bus, bus adapter, and controller to be used by a new disk, one might want to specify a hierarchy such as:

```
/sbus/vme/[new controller type]/[new disk type]
```

Rules for specifying hierarchical device names are given in "Device Naming Conventions," below. All open boot device names must initially be entered as complete pathname hierarchies.

Hierarchical device names can be long and complicated; entering a full hierarchical device every time one wanted to execute a command to boot would be a tedious and error prone process. To simplify commands, the open boot PROM provides a set of predefined one-word aliases for standard devices and allows you to define an alias of your own for any hierarchical device name. For the great majority of users, no specific knowledge of hierarchical device naming will ever be necessary in order to operate their system.

## 5.3. Device Naming Conventions

A hierarchical device name is specified by giving a path from the top node of the device tree to the device itself. Each node on the device path is marked by a slash ("/"). An example of a device-name path is:

```
/sbus@1, f8000000/esp@0, 800000/sd@3, 0:b
```

The first "/" represents the highest node, the workstation itself. Connected to it is an SBus at the address `f8000000`. The next node is the SBus itself. At the first slot on the SBus ("@0"), with an address offset of `800000`, is an `esp` SCSI host adapter. A SCSI disk (`sd`) is connected to the `esp` SCSI host adapter as logical unit 1 (0) at target 3 (this corresponds to UNIX device name `sd0`). Finally, the



element after the “:” is a device-specific parameter. In this case, it identifies partition `b` of disk `sd0`, which might be used in booting the miniroot during installation.

The devices in a device tree are identified by node names consisting of:

*driver-name@unit-addresses:device-parameters*

`sbus@1, f8000000` is a node name in the device-name path above. `1` and `f8000000` are unit addresses for the SBus. No SBus device parameters are specified.

A *driver-name* can be made up of letters, digits, and the special characters “\_,” “+,” “-,” which are treated as letters. Upper and lower case are distinct; length can be from 1 to 31 characters. `sbus` and `esp` are driver names.

*@unit-addresses* is optional. If specified, there can be one or more unit addresses. A *unit-address* can be made up of letters and digits. It can be arbitrarily long, although sequences of more than eight characters are discouraged.

*:device parameters* is optional. If specified, there can be one or more device parameters. A *device parameter* can be made up of letters and digits. It can be arbitrarily long, although sequences of more than eight characters are discouraged.

#### 5.4. Using `show-devs` Command to Display Device Names

From the PROM Monitor prompt, you can use the `show-devs` command to display the devices known to a system:

```
ok show-devs driver-name
```

– *driver-name* is optional; when used, `show-devs` only displays device paths descending from the node identified by the entry.

```
ok show-devs
/options
/fd@1, f7200000
/ttyb
/ttya
/keyboard
.
.
.
/sbus@1, f8000000/esp@0, 800000/sd
/sbus@1, f8000000/esp@0, 800000/st
/packages/deblocker
/packages/sun-label
```

Note: Sample display; contents may vary, depending on user input and the system used.

```
ok show-devs /sbus
/sbus@1, f8000000/le@0, c00000
/sbus@1, f8000000/dma@0, 400000
/sbus@1, f8000000/esp@0, 800000
/sbus@1, f8000000/esp@0, 800000/st
/sbus@1, f8000000/esp@0, 800000/sd
```

Note: Sample display; contents may vary, depending on user input and the system used.

## 5.5. Using `devalias` Command to Show and Define Device Aliases

The `devalias` command allows you to list all device name aliases currently defined and to define additional aliases. The new aliases are maintained in system memory; they are not kept in NVRAM and need to be re-entered whenever your system reboots or is shut down.

Used alone, `devalias` lists aliases:

```
ok devalias
disk          /sbus@1, f8000000/esp@0, f800000/sd
tape         /sbus@1, f8000000/esp@0, f800000/st
net          /sbus@1, f8000000/le@0, f800000
```

Note: Sample display; contents may vary, depending on user input and the system used.

To define a new device path name alias, use `devalias` with an alias specification:

```
ok devalias alias device-path
```

## 5.6. Open Boot PROM Environment Parameters Stored in NVRAM

A number of environment parameters that control essential open boot PROM monitor functions are stored in NVRAM. These parameters can be examined, modified, and reset to factory defaults with the `printenv`, `setenv`, `set-default` and `set-defaults` commands.

### `printenv` Command

Used in the form

```
printenv environment-parameter
```

`printenv` displays the current value of the specified environment parameter and its original factory default value.

Used alone, without specifying an environment parameter, `printenv` displays all current values for NVRAM environment variables and their original factory default values:

```

ok printenv
Parameter Name      Value      Default Value
name                options   options
oem-logo
oem-logo?          false     false
oem-banner
oem-banner?        false     false
sbus-probe-list    0123     0123
output-device      screen    screen
input-device       keyboard  keyboard
keyboard-click?   false     false
keymap
diag-file
diag-device        net       net
boot-file
boot-device        disk     disk
auto-boot?         false    true
watchdog-reboot?  false    false
fcode-debug?      false    false
screen-#columns   80       80
screen-#rows      34       34
sunmon-compat?    false    true
diag-switch?      false    true

```

**setenv Command**

Use `setenv` followed by the name of an environment parameter and a value to enter a new parameter value. For example, to change the current value of the parameter `boot-device` so that a client system default boots from its server, you would enter:

```
ok setenv boot-device net
```

**set-default and set-defaults Commands**

The command `set-defaults` resets most parameters to their factory default values. The command `set-default` followed by the name of an environment parameter resets the specified parameter to its factory default value.



---

## Bugs Fixed in SunOS 4.1.2

Over 200 bugs have been fixed for SunOS 4.1.2. This chapter lists the more significant fixes, including CTE patches included in SunOS 4.1.2 (see Section 6.13). The bugs are listed in categories, and include the bug ID and a synopsis of the bug. The following categories are included:

- Kernel
- Network
- SCSI
- Graphics
- Utility
- Library
- Libsuntool
- Sunview
- Compiler/Loader
- C2
- Documentation
- Install
- CTE Escalations
- CTE OpenWindows Version 2 Patches

### 6.1. Kernel

1008324 TIOCCONS is a security hole  
1016767 /boot works for 1st disk controller only  
1029802 4/280 with xy451 panics: regs accessed while busy (patchid 100364-01)  
1032053 getreg should use fuword() when simulating instructions  
1032534 Kernel panics with panic: psig on exiting sunview  
1033861 Request an intelligent swap block freeing algorithm  
1036167 modified superblock may not be updated at unmount  
1036196 MLOCK/MUNLOCK broken  
1036449 modified superblock may not be updated at unmount

1037039 ufs\_readdir hangs on a Olen entry  
1037052 shmtd(2) will not detach a read only segment  
1037715 ISO orderly release is not supported, but TCPTLI says so  
1038651 Using tmpfs ie ram /tmp can cause data initlztn w Fortran on 4.1  
1038686 System panics with "panic: hat\_ptesync - invalid pme" message  
1039275 writing to an PROT\_READ area is giving SIGBUS not SIGSEGV  
1039287 tcp/tli transport cannot send more than 4052 bytes of data  
1039410 putting default swap partition in fstab causes system crash, dump  
1039840 unmount doesn't close block device  
1039846 unmount leaves metadata cached  
1039850 inode left locked on symlink creation  
1040722 Pseudo-ttys become attached to other users or hang  
1042350 tmpfs problem with ftruncate test in SCTS from SUNVS  
1042808 nopagereclaim erroneously set to 1 for Sun-4/4xx  
1043270 tmpfs disallows mmap beyond end of file  
1043801 panic: assertion failed, pmg->pmg\_keeptcnt == 1  
1044192 hard link to a directory succeeds on tmpfs for non-root users  
1044666 panic: ttcompat: unexpected ioctl acknowledgment  
1044668 umount/sync race can hang system  
1044888 Streams run out of dblks and hung the system  
1045360 spurious VME intrrpts cause kernel to panic: data f  
1045582 NeWSprint 1.0 hangs on 4/75 and 4/40  
1046449 multiple-register sbus cards don't work on a 4/75  
1047295 mounting certain iso 9660 cd's succeed but directory becomes data  
1047355 polling not working correctly  
1047586 mkfifo does not work on tmpfs  
1048128 xon flow control problems on serial i/o  
1049010 DLS60 kernel config file is incorrect.  
1049027 tmpfs hangs and deadlocks  
1049125 System panic's when reject a network connection via TLI  
1050340 mmap(MAP\_PRIVATE), madvise(MADV\_SEQUENTIAL) => panic: swap\_xlate  
1050540 mount and sync can deadlock and hang the box  
1050543 umount can cause a inactive panic  
1050546 putpage can hang the box  
1050548 nfs\_putpage can hang the box  
1050558 Sparcstation 2 running SunOS 4.1.1 crashes under some applications  
1051875 mkfs can create an unusable file system  
1051876 system hangs when pageout sleeps in ufs\_putpage  
1051983 buffer cache sz decreases w lg kernels causing high (patch 100330-02)  
1051992 pageout and clustering are not cooperating  
1052649 write system call is returning EINTR  
1052669 panic: data fault in strwrite - streams corruption  
1053582 mount/sync window causes panic: sleep  
1054054 +s and +t together do not work as expected  
1054999 syncip overhead in ufs\_inactive causes poor NFS performance  
1056058 panic when writing to a >=2g-bsize file  
1056245 Process hangs in page\_cv\_wait, ufs\_getapage after locking page  
1056992 4.1.1 leaks kernelmap  
1056992 4.1.1 leaks kernelmap (patch 100330-02)

1057211 VME master bus accesses can time out during heavy I/O  
 1057478 access to non-existnt mmap /dev/sbus cause panic: async memory err  
 1057481 System hangs when running TLI  
 1057929 sys hang: prestosrve and NC400 exhausting kernelmap (patch 100330-02)  
 1057963 cannot mount an exported tmpfs directory  
 1060281 poll() sys call gives bad returns for fd events  
 1061174 poll does not return -1 when it receives a signal  
 1063310 floppy driver reports drive as "unexpectedly busy"  
 1065858 Cant run OW xinit on 4/670, 4/630 running 412A1.2  
 1066086 4.1.2 kernel crashes under async I/O load, running MP  
 1066422 performance enhancements for the sun4m IPI driver  
 1066743 Track crosscall stats separately from intrrpt stats  
 1067681 lockfs signal behavior may break applications  
 1068051 profiling kernel support for sun4m  
 1068283 vmstat reports incorrect numbers  
 1068363 system crashes with 112mb memory and 128 maxusers  
 1068462 kernel workaround needed for bugid 1067719  
 1068548 PROM mailbox not properly mapped  
 1068907 remove CDROM driver debug info  
 1068910 Dump of data using /dev/sbus\* device files broken  
 1069072 Int div on sparc can be used to modify data  
 1070099 support for removable disks got busted in 4.1.1

## 6.2. Network

1006905 add lock file to render rpc.yppasswdd s1ngle threaded  
 1034328 client can crash if two procs unlink (patchid 100173-06)  
 1038060 ie0: WARNING: if\_snd full error still noted  
 1039326 vmunix: ie0: lost interrupt: resetting  
 1041303 ifconf() off by one  
 1045211 Problem with booting diskless clients  
 1045531 Lock Mgr. suspends "file locking" processes indefinitely  
 1045536 nfs expts to non-sun sys can result (patchid 100173-06)  
 1064433 Export of subtree doesn't work (patchid 100173-06)  
 1066287 nfs hang when looking at lg file changed on server (patchid 100173-06)  
 1066663 SIGIO signal for async io fails to pend across NFS (patchid 100357-01)  
 1029628 setuid bit copied to NFS file system  
 1030884 NFS full file sys gets ENOSPC even when overwriting existing file  
 1034328 client can crash if two proc's unlink the same file at once  
 1034750 automount hangs while trying to open a mount point  
 1037476 Sending bad proc to NFS server can cause mbuf leak  
 1038302 NFS export option "anon=-1" does not work  
 1038308 4.1 automount sets up a different sym link 5 mins after 1st access  
 1039406 3960-3980th repeated sckt open & close fails w addr already in use  
 1039839 nres\_gethostbyaddr logs erroneous messages to NIS server's console  
 1039977 bug in the NFS DEBUG code, file nfs\_export.c  
 1041409 cannot execute setuid root programs in NFS mounted filesystem  
 1041559 system is panicing in mclput  
 1042435 client side noninterruptible hang

1043343 increase arp table size  
1044565 pc-nfs client failed locking entire UNIX file  
1045536 NFS exports to non-sun systems can result in file truncation  
1045700 lockf fails when two processes try to read, write simultaneously  
1045993 NFS file caching still occurring on locked files  
1045995 unlink(2) on a file will prevent NFS file locks on the same file  
1045996 File locks are not removed on the server when a client is rebooted  
1046001 fcntl is interrupted by signals and returns the wrong error code  
1046945 'ping' caused machine to panic  
1047557 Old pages not being purged if file gets truncated on server  
1048875 Missing /etc/mstab file can fork the automounter  
1052330 locking on NFS file doesn't cause cache purged after the 1st time  
1052879 panic iesynccmd with SunNet OSI 7.0  
1053552 ip\_input drops pkts w IP opts as bad checksum w mclput pnc patch  
1053679 file region locking of NFS files was broken in 4.x  
1054669 telnetd allows password snooping  
1057673 panic: iinactive may result from klm code  
1057685 4.1.1 NFS servers freezing  
1057878 panic due to window in rtfree()

### 6.3. SCSI

1036481 bsr after reading EOF doesn't record space back, wrong pstn stat  
1042822 st: warning tape may be wearing.... on Exabytes  
1045071 sd: does not scan past 1 GB when using format utility  
1045586 Handling of special FORMAT command is broken w.r.t blkno decode  
1046305 some XXgetcap cases reversed  
1046580 bug in esp host adapter can cause panic during error recovery  
1048141 esp does not always recognize a marginal SCSI bus  
1049417 select phase not timed out causing indefinite hangs  
1049674 general bug report for some esp problems  
1049830 timeouts in polled mode cause a panic  
1050067 esp driver resource alloc fails when esp is child if dma  
1051244 SCSI Tape drive is misusing signals  
1052613 max dma limits set by host adapters are too small  
1052659 FSR on the old SCSI tape driver fails in 4.1.1  
1052888 st driver can cause a panic: zero divide trap  
1055184 3/2 sundev/sd.c uses only one buffer per unit  
1058682 Reassign block (format "repair") malfunctions beyond 1GB  
1059139 Accessing a SCSI tape drive causes system crash  
1062430 st drv'r writes extra file mrks if opened then closed after writing  
1065300 front load tape doesn't reliably read tapes  
1067300 st: soft errors sometimes cause loss of data  
1070535 esp: sync xfer rate is not renegotiated  
1070540 esp: incompl't cleanup after proxy msg causes panic  
1070876 delay after scsi bus reset required  
[various] improvements in st ioctl, EOF, EOT, EOM handling & status rprt



## 6.4. Graphics

- 1045577 rasfilter\_rgbtobgr always fails
- 1046046 The colormap is not correct when 128 colors are used on 4.1.1
- 1046327 P4 CG6 hardware cursor invisible under Sunwindows on OS4.1.1
- 1063738 Diffuse positional lights on color-per-vertex tri
- 1063743 Ambient positional lights for color-per-vertex
- 1064295 Degenerate hollow triangles cause internal edge
- 1066677 GT crashes w SunPHIGS Quick Update of batched polyl
- 1066772 wide patterned polylines without vertex colours are
- 1066851 gtconfig -M flag doesn't preserve VIDEO\_ENABLE
- 1066983 quick check pgm to visually test accel port to dev
- 1067972 hi res version of cg12 (gsxr) text is trash in pr\_b
- 1068136 gtconfig needs sync generator setup pgms for NTSC
- 1068153 Leaf nodes which set no attributes cause QUM mode
- 1068802 pixwin double buffer bug for GS
- 1069396 gtconfig: sync generator pgm change needed for
- 1069691 phigschild crashes GT system when huge application
- 1070466 Setting hatch\_transparency to false affects perf
- 1071971 Get jitter when using 1280 @ 67hz video output forma

## 6.5. Utility

- 1016437 lpd does not check file names for legality
- 1022363 format: remove dummy defect list commit for embedded SCSI disks
- 1025250 Zero-length defect list causes format to fail extractn of mnfc...
- 1030087 sendmail yp aliasing does not work with non-sun yp masters
- 1031577 modload does not recognize the "vector" keyword
- 1035625 not enough information on embedded scsi defect lists
- 1036159 A user can run programs with root's group privileges...
- 1039221 bar core dumps w -x or -t option
- 1045185 Disks are displayed in controller order but chosen in disk order
- 1045636 4.1 /bin/mail problem delivering mail
- 1047340 /bin/mail can be used to invoke a root shell
- 1049010 DLS60 config file does not work
- 1050675 Original, extract format command fails if defect count is zero
- 1051638 format mode selects don't support SCSI-2 page format
- 1052805 Low-level format of MD21 disks usually fail with Warning...
- 1053733 pax gets into infinite loop at EOT
- 1055402 Bourne shell scripts using inline redirection and
- 1058003 When reading from /dev/printer lpd does not check how much data it has read
- 1058236 checksum error even with -c -i options
- 1059212 Improvement needed in mode select error handling for SCSI disks
- 1059225 Do not hardwire any SCSI disk mode select parameters
- 1063772 possible to overwrite a file on system using lpr
- 1067216 Format fails on 207mb maxtor drive
- 1068402 mkfs should initialize UFS clean flag
- 1068640 formatting CDC drives generates errors

## 6.6. Library

- 1038500 localtime or tzsetwall corrupts malloc space (libc)
- 1041074 The disk version of the "C" locale is incomplete
- 1044722 undefined symbol in libxpg.a
- 1044909 /usr/lib/expr preserve race condition was used to break into a system
- 1045471 4.1(+) shared C libraries reference undefined symbols
- 1050040 fcvt() segment faults under 4.1.1
- 1051619 system() uses the system call fork() instead of vfor
- 1051881 colldef rejects apparently valid input
- 1052398 strxfm is not 8 bit clean
- 1052463 colldef can't handle a substitute statement with hex numbers
- 1053346 There shouldnt be an imposed length limit for strings in strcoll()

## 6.7. Libsuntool

- 1038363 window\_destroy does not release memory; 8k memory leak per textsw

## 6.8. Sunview

- 1028856 2/2 master defaults database is not being read
- 1038313 1/1 Can't open and use more than 64 windows concurrently
- 1039576 1/1 selection\_svc and rpc can be used to gain access to system files
- 1039760 3/2 notifier fails to handle fcntl for FNDELAY correctly
- 1040606 2/3 selection\_svc security problem

## 6.9. Compiler/Loader

NOTE: new compiler/loader in SunOS 4.1.2 includes substantial fixes that may affect some application programs. See "Link-editor Bug Fixes" in Chapter 2 for more detail.

- 1019004 -assert definitions can fail to report undefined symbols
- 1032208 ldd doesn't write file names if standard output is not a tty
- 1032209 ldd runs out of file descriptors
- 1032739 ld core dumps with many libs in mult. directories
- 1033086 Calling dlopen() on a file with BSS kills your application
- 1034788 -r and -pic do not mix bad secondary magic number on 4.1
- 1034833 ld: can't mixed -r with -Bstatic or -A flag in 4.1
- 1034844 missing symbol from ld -r 4.1 object; exists in 4.0.3 ld output
- 1037879 Cannot create executable w shared obj which points to another
- 1042261 ld only recognizes first directory in LD\_LIBRARY\_PATH
- 1043082 Bogus first arg to 'ldconfig' kills dynamic linking
- 1043300 dlclose prevents access to functions even after further dlopen
- 1044524 multiply defined symbols and seg. fault caused by 4.1's ld
- 1045194 dlsym returns bad addr for uninit global var in dlopened library
- 1045272 ld -u & -r dont seem to work properly
- 1046379 cant call a fnctn in a shared lib from a fnctn in another ...
- 1046462 ld fails with seg fault proc nested incl
- 1050594 Uninitialized struct slot causes intermittent failures
- 1052428 ld.so usage of -L options confusing, leads to security problems
- 1064820 ld can produce bogus diags with cascaded
- 1069404 bad handle errors with dlsym(dlopen(NULL,1),"xxx")

**6.10. C2**

- 1040334 yppasswd wont allow user to change passwd from client..daemon dies
- 1040465 The passwd -e command does not work when c2 security is in place
- 1044204 rpc.passwdauthd can be used to guess passwords on c2 systems
- 1047131 getauditflagschar() routine as distrib in 4.1 libc does not work

**6.11. Documentation**

- 1051880 colldef input format not documented
- 1051882 semantics of colldef are unclear

**6.12. Install**

- 1041627 add\_services can fail if partition changes are made after SunInstall.
- 1047095 install from CD stops and complains if floppy is in
- 1047696 Can't boot 4.1.1 munix on a sun4c w/40mb of memory
- 1063858 disk form takes 5 minutes to exit in a 40 disk machine
- 1064779 dataless install corrupts file system
- 1069645 should add support for keytables in munix

**6.13. CTE Escalations Fixed in SunOS 4.1.2**

Listed below are the Corporate Technical Escalations (CTE) patches which have been incorporated in SunOS 4.1.2. The list is in patch ID order and includes bug IDs and a synopsis of each fixed bug.

- 100071-01 1038313 Can't open and use more than 64 windows concurrently
- 100072-01 1031577 modload does not recognize the "vector" keyword
- 1038305 modload does not recognize the "vector" keyword
- 100075-07 1044565 pc-nfs client failed locking entire UNIX file
- 1045700 lockf fails when 2 procs try to read, write simultaneously
- 1046001 fcntl is interrupted by signals, returns the wrong err code
- 1045996 File locks not removed on server when a client is rebooted
- 1045995 unlink(2) on a file prevents NFS file locks on same file
- 1057673 panic: iinactive may result from klm code
- 1056300 failed tlock test #1: should ret EINVAL instead of success
- 1061087 File /tmp./foo left on the server after a client reboot
- 1061173 File record can be read-, write-locked by diff procs simul
- 100085-03 1039576 selection\_svc, rpc can be used to gain access to sys files
- 1040606 selection\_svc security problem
- 100109-01 1032053 getreg should use fuword() when simulating instructions
- 100125-04 1054669 telnetd allows password snooping
- 100133-01 1036159 A user can run programs with root's group privileges...
- 1030087 sendmail yp aliasing does not work with non-sun yp masters
- 100134-03 1042822 st: warning tape may be wearing.... on Exabytes
- 100141-02 1039839 nres\_gethostbyaddr logs erroneous msgs to NIS server's cns
- 100149-03 1041559 system is panicing in mclput
- 1053552 ip\_input drops pkcts w IP opts as bad checksum w mclput pnc patch
- 100159-01 1039406 3960-3980th repeated sckt open & close fails w addr already in use
- 100168-01 1037747 crontab -e doesn't update file edited in less than 1 setup\_client
- 100170-03 1034833 ld: can't mixed -r with -Bstatic or -A flag in 4.1
- 1034788 -r and -pic do not mix bad secondary magic number on 4.1

- 1044524 multiply defined symbols and seg. fault caused by 4.1's ld
- 1042261 ld only recognizes first directory in LD\_LIBRARY\_PATH
- 1045272 ld -u & -r dont seem to work properly
- 1037879 Cant create executable w shared obj which points to another
- 1019004 -assert definitions can fail to report undefined symbols
- 100173-03 1039977 bug in the NFS DEBUG code, file nfs\_export.c
- 1032959 NFSPROC\_MKDIR returns incrr attributes
- 1029628 setuid bit copied to NFS file system
- 1037476 Sending bad proc to NFS server can cause mbuf leak
- 1038302 NFS export option "anon=-1" does not work
- 1034328 client can crash if two proc's unlink same file at once
- 1045536 NFS exports to non-Sun systems can result in file truncatn
- 1030884 NFS full file sys gets ENOSPC even wn overwrite exist file
- 1045993 NFS file caching still occurring on locked files
- 1047557 Old pages not being purged if file truncated on server
- 1052330 locking on NFS file doesn't cause cache purged after 1st time
- 1053679 file region locking of NFS files was broken in 4.x
- 1041409 cannot execute setuid root pgms in NFS mounted filesystem
- 100173-06 1034328 client can crash if two procs unlink
- 1045536 nfs expts to non-sun sys can result
- 1064433 Export of subtree doesn't work
- 1066287 nfs hang when looking at lg file changed on server
- 100174-01 1042350 tmpfs problem with ftruncate test in SCTS from SUNVS
- 1038651 Using tmpfs ie ram /tmp can cause data initlztn w Fortran on 4.1
- 1044192 hard link to dir succeeds on tmpfs for non-root users
- 1043270 tmpfs disallows mmap beyond EOF
- 1057963 cannot mount exported tmpfs directory
- 1049027 tmpfs hangs and deadlocks
- 1054054 +s and +t together do not work as expected
- 1047586 mkfifo does not work on tmpfs
- 100179-01 1047355 polling not working correctly
- 100181-01 1039760 notifier fails to handle fcntl for FNDELAY correctly
- 100186-01 1046945 'ping' caused machine to panic
- 100188-01 1008324 TIOCCONS is a security hole
- 100192-01 1046046 The colormap is not correct wn 128 colors used on 4.1.1
- 100198-01 1037052 shmtd(2) will not detach a read only segment
- 100199-01 1037715 ISO orderly release is not supported, but TCPTLI says so
- 1039287 tcp/tli transport cannot send more than 4052 bytes of data
- 1049125 System panic's when reject a network connection via TLI
- 1057481 System hangs when running TLI
- 100201-01 1040465 passwd -e cmd doesn't work when c2 security is in place
- 1044204 rpc.passwdauthd can be used to guess pswrds on c2 systems
- 1040334 yppasswd wont allow user to change passwd from client...
- 1047131 getauditflagschar() routine in 4.1 libc does not work
- 100203-01 1050040 fcvt() segment faults under 4.1.1
- 100206-01 1038363 window\_destroy does not release memory; 8k memory leak per textsw
- 100207-01 1016767 /boot works for 1st disk controller only
- 100208-01 1045471 4.1 (+) shared C libraries reference undefined symbols
- 100210-01 1028856 master defaults database is not being read

100211-02 1038050 if an "eom" ioctl gets interrptd, tape synchronization lost  
 100216-01 1046449 multiple-register sbus cards don't work on a 4/75  
 100217-01 1047295 mounting certain iso 9660 cd's succeed but directory becomes data  
 100224-02 1047340 /bin/mail can be used to invoke a root shell  
 1045636 4.1 /bin/mail problem delivering mail  
 100225-02 1048128 xon flow control problems on serial I/O  
 100228-02 1032534 Kernel panics with panic: psig on exiting sunview  
 100232-01 1050558 SPARCstation 2 running SunOS 4.1.1 crashes under some applications  
 100243-01 1046580 bug in esp host adapter can cause panic during error recovery  
 1048141 esp does not always recognize a marginal SCSI bus  
 1046305 some XXgetcap cases reversed  
 100244-01 1052669 panic: data fault in strwrite - streams corruption  
 100244-02 1033287 Frequent exhaustion of chanmap resource observed  
 100249-01 1038308 4.1 automount sets up a different sym link 5 mins after 1st access  
 100250-01 1052659 FSR on the old SCSI tape driver fails in 4.1.1  
 100251-01 1044909 /usr/lib/expreserve race condition used to break into a system  
 100254-01 1044666 panic: ttcompat: unexpected ioctl acknowledgment  
 100255-01 1044888 Streams run out of dblks and hang the system  
 100256-01 1053733 pax gets into infinite loop at EOT  
 100257-02 1033086 Calling dlopen() on a file with BSS kills your application  
 1043300 dlclose prevents access to functions even after further dlopen  
 1052428 ld.so usage of -L options confusing, leads to security problems  
 1046379 cant call a fnctn in a shared lib from a fnctn in another ...  
 1045194 dlsym returns bad addr for uninit global var in dlopened library  
 1032208 ldd doesn't write file names if standard output is not a tty  
 1032209 ldd runs out of file descriptors  
 1043082 Bogus first arg to 'ldconfig' kills dynamic linking  
 100259-01 1054999 syncip overhead in ufs\_inactive causes poor NFS performance  
 100262-01 1052888 st driver can cause a panic: zero divide trap  
 100265-01 1039275 writing to an PROT\_READ area is giving SIGBUS not SIGSEGV  
 100266-01 1038500 localtime or tzsetwall corrupts malloc space (libc)  
 100268-02 1051880 colldef input format not documented  
 1051881 colldef rejects apparently valid inpt  
 1051882 semantics of colldef are unclear  
 1051883 spelling error in colldef message  
 1052463 colldef can't handle a substitute statement with hex numbers  
 1053346 There shouldnt be an imposed length limit for strings in strcoll()  
 1052398 strxfrm is not 8 bit clean  
 100273-01 1038686 System panics with "panic: hat\_ptesync - invalid pme" message  
 1043801 panic: assertion failed, pmg->pmg\_keept == 1  
 100275-01 1057211 VME master bus accesses can time out during heavy I/O  
 100281-01 1057685 4.1.1 nfs servers freezing  
 100294-01 1056245 Process hangs in page\_cv\_wait, ufs\_getapage after locking page  
 1051992 pageout and clustering are not cooperating  
 1051876 system hangs when pageout sleeps in ufs\_putpage  
 1050546 putpage can hang the box  
 1045582 NeWSprint 1.0 hangs on 4/75 and 4/40  
 100305-06 1016437 lpd does not check file names for legality  
 1058003 Reading from /dev/printer, lpd does not check how much it has read

	1063772	possible to overwrite a file on system using lpr
100313-01	1044668	umount/sync race can hang system
100330-02	1051983	buffer cache sz decreases w lg kernels causing high
	1056992	4.1.1 leaks kernelmap
	1057929	sys hang: prestosrve and NC400 exhausting kernelmap
100346-01	1044722	undefined symbol in libxpg.a
100349-02	1066773	K_PICK_ECHO_STRUCT_EN_RANGE_PLUS_DESCENDANTS may stop echoing in a descendant
100356-01	1067615	kernel hangs while processing aiocancel system call
100357-01	1066663	SIGIO signal for async io fails to pend across NFS (patchid)
100364-01	1029802	4/280 with xy451 panics: regs accessed while busy
100376-01	1069072	Int div on sparc can be used to modify data
100376-02	1069702	Integer division on SPARC can be used to gain root access

## 6.14. CTE OpenWindows Version 2 Patches

Patch-ID# 100131-01 Synopsis: Custom CADENCE xnews server with two bug fixes Bugs Fixed: 1043582 popup windows cause damage if the are partially off screen. 1043824 Intermittent failure to deactivate automatic pointer grab.

Patch-ID# 100156-01

Synopsis: xauth sets wrong hexkey data in the add displayname option

Bugs Fixed:

1045483: xauth sets wrong hexkey data in the add displayname option

Patch-ID# 100158-01

Synopsis: xview notifier cannot handle number of files > 31

Bugs Fixed:

1040844: xview notifier cannot handle files > 31

Patch-ID# 100166-01

Synopsis: PANEL\_SLIDER SEGVs when slider bar is moved - monochrome only.

Bugs Fixed:

1042937: PANEL\_SLIDER SEGVs when slider bar is moved - monochrome only.

Patch-ID# 100176-08

Synopsis: OpenWindows 2.0: Patch release 8 for X11-NeWS server

Bugs Fixed:

1040959 - odd line widths identical to even widths (5==4) shapes bug.

1043824 - Intermittent failure to deactivate automatic pointer grab.

1043582 - popup windows cause damage if the are partially off screen

1041986 - XGetImage will crash server if planemask is set to AllPlanes...

1043914 - CharStrings does not contain entries for unencoded characters.

1044043 - pointer grabs generate incorrect enter/leave events.

1044086 - incorrect rendition of some stroke characters NeWS bug.

1044246 - killing a client while the server is grabbed causes SEGV

1044315 - The server can get into an infinite loop and consume cpu time

1044475 - XDrawArc doesn't draw large circles correctly

1045150 - R to L copies less than one longword.

- 1046997 - server hangs if button event generated when socket is full
- 1045563 - XDrawLine with negative points crash the server
- 1045650 - Bad pixmaps are created and NO ERROR is returned....
- 1045652 - Creating 100 to 200 pixmaps causes the server to crash
- 1045857 - FrameMaker 2.1X crashes server
- 1045887 - cgsix, sparc station, server crashes on XDrawSegment
- 1047358 - KeymapNotify events being generated in the wrong order.
- 1047494 - NeWS line > 1 cap style CapRound, caps hanging bug.
- 1048120 - NeWS line > 1 not drawn correctly if cap style CapRound.
- 1050109 - Line Width > 1 not drawn correctly.
- 1051287 - DGA leaves file descriptor open for color map.
- 1051689 - Short rlineto lose accurate current point for wide lines.
- 1051848 - Plane masking bug.
- 1052157 - xcs\_clear sh\_mem\_fill clear region server crash.
- 1052239 - sh\_Shape\_Raster() bad calculation memory corruption core dumping.
- 1052479 - Out of bounds framebuffer access.
- 1052697 - Out of bounds framebuffer access.
- 1052712 - NeWSPrint imagecanvas boundaries rounding bug.
- 1052749 - Shapes bad calculation memory corruption core dumping.
- 1052822 - Out of bounds framebuffer access.
- 1053164 - Fill operator causes dropouts of Type 3 Fonts.
- 1053342 - Visibility Notify not received when window partially obscured.
- 1053511 - Windows over 32000 pixels BadAlloc shapes bug.
- 1053954 - Visibility Notify enabled impacts server performance.
- 1054009 - Set screen saver bug.
- 1054362 - Postscript accentuated characters disappear text shifts.
- 1054811 - NeWS Lotus graphs not rendering right and bottom lines.
- 1054918 - XNeWS would not render 256 colors.
- 1055187 - XFillRectangle on a cg3 stipple incorrect results.
- 1055265 - Plaid OW demo crashes LSCrev2 legosc
- 1055656 - XSetFillStyle FillTitled stipple fill incorrect.
- 1055672 - Pageview does not display Postscript document correctly.
- 1056088 - Protect the server from out of framebuffer accesses.
- 1056131 - Closepath doesn't produce a line join if last element zero length.
- 1056324 - NewsPrint Sparc Printer Bad Line Widths.
- 1056454 - Frame on OWV2 can crash legosc quadro systems.
- 1057110 - LSC Rev 2 chip has bug in frame buffer arbitration.
- 1057234 - XGL and server graphics can crash legosc.
- 1057502 - Doubledash lines not rendered when rop set.
- 1057750 - xcs\_clear xcsProcs shapes fill area needs range clipping.
- 1058592 - XSetClipRectangles inconsistent clipping action bug.
- 1058818 - xc\_create\_transmem\_raster not clipping child raster to fit parent.
- 1059182 - Postscript file will not render properly.
- 1059186 - Postscript line drawing and rendering bug.
- 1059230 - SparcPrinter won't print excel postscript files.
- 1059933 - Can't printout NeWSprint characters are upside down.
- 1060093 - Stippling and clipping not displayed correctly.
- 1060964 - colormap corruption server core dump when running props.
- 1061098 - X11 wide lines coincident first & last points not joined.

1061305 - XDrawArcs test in MIT Xlib test crashes xnews server.  
1061436 - xnews return wrong Fontmatrix on redefined font PostScript bug.  
1061881 - do not return per character metrics for fixed size fonts.  
1062574 - Window grabber clip problem cv\_unmap.  
1062616 - pageview will send xnews process into an infinite loop  
1062834 - events open declaration syntax error.  
1063240 - docviewer intro screen corrupted after installing xnews patch.  
1063261 - cv\_reshapecanvas to send expose properly.  
1063464 - ListFontsWithInfo slow performance.  
1063821 - window mapping bug.  
1069548 - poor performance of XQueryFont  
1069590 - xnews server crashes after allocating 216 colors

Also fixed (no bugid's):

- Tile and Stipple related bug fixes.
- XGetImage ZPixmap transfered whole raster error.
- Shape\_Obj excessive local temporary shapes being created.
- DragAndDrop cursor does not appear.
- Snpatool does not work correctly.
- AnswerBook does not display pages correctly.
- Performance enhancements for font queries
- psterm title bar does not repaint after resizing
- IslandDraw core dumps at startup time
- VideoPix's vfctool dies with X error
- Mwg does not repaint root window correctly

Patch-ID# 100183-01

Synopsis: Openwin 2.0:rescale font to 60pt size and convert to vfont doesn't work

Bugs Fixed:

1045767: when you rescale folio font to 60pt size and convert to vfont, it dent work

Patch-ID# 100184-02

Synopsis: sv\_xv\_sel\_svc and rpc can be used to gain access to system files

Bugs Fixed:

1040747: sv\_xv\_sel\_svc and rpc can be used to gain access to system files.

Patch-ID# 100205-01

Synopsis: Highlighting problems within filemgr running on Xterminals.

Bugs Fixed:

1045939: filemgr incorrectly inverts icons on a NCD mono X terminal

Patch-ID# 100213-01

Synopsis: Application core dunks when user selects text item and drags it on the scrolling list.

Bugs Fixed:

1047260: drag text field into scroll list causes app. core dump



Patch-ID# 100231-01

Synopsis: PANEL\_LABEL\_WIDTH does not set button size correctly.

Bugs Fixed:

1037262: PANEL\_LABEL\_WIDTH doesn't work

Patch-ID# 100245-01

Synopsis: olwm banding bug, label centering bug

Bugs Fixed:

1038716: Olwm window labels are not always centered properly.

1047748: Olwm can get into the "move window" rubber banding mode by mistake.

1055511: Olwm save workspace only works for WM\_SAVE\_YOURSELF clients.

1053101: Olwm XV\_SHOW fails when DISABLED and ENABLED quickly.

Patch-ID# 100246-01

Synopsis: panel buttons fail when multiple server connections are made.

Bugs Fixed:

1041574: panel buttons fail when multiple server connections are made.

Patch-ID# 100247-01

Synopsis: Xview libraries multiple bug fixes.

Bugs Fixed:

1051808: Control panel with a scrolling list sometimes displays wrongly sized.

1048055: Canvases and control areas < 2 pixel apart draw in wrong size.

1028072: xv\_set for scrolling lists has side effects.

1046128: Non-exclusive menus calling wrong notify proc when item selected.

1045626: Can't specify different font for each panel item.

1049478: Panel buttons only inherit panel fonts conditionally.

Patch-ID# 100352-01

Synopsis: bldfamily, convertfont & makeafb compress fontnames causing conflicts

Bugs Fixed:

1043430: convertfont & bldfamily default filename length too small

1052174: bldfamily should handle font name conflicts better

Patch-ID# 100354-02

Synopsis: cm displays NoName as user if user has a "-" in it's name

Bugs Fixed:

1046847: cm comes up with NoName displayed as user if user has a "-" in it'sname.

Patch-ID# 100355-01

Synopsis: OpenWindows 2.0: patch for panel text item bug

Bugs Fixed:

1065873: PANEL\_VALUE\_DISPLAY\_LENGTH in text item does not work for values less than 4

Patch-ID# 100366-01

Synopsis: filemgr does not handle automounted directories correctly

Bugs Fixed:

1043301: subdirectory mounts don't work with hierarchical mounts

1040123: indirect maps creates symbolic links inconsistently with subdirecto

Patch-ID# 100367-01

Synopsis: Cursor advances too far when using compose key in OLIT 2.5 appl.

Bugs Fixed:

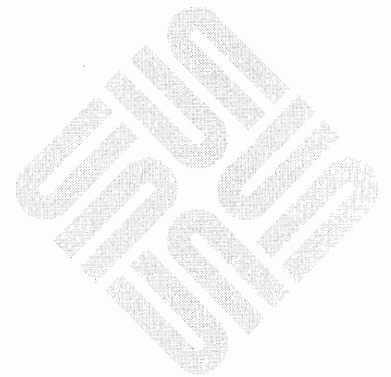
1067129: TextEdit widget does not insert 8-bit characters correctly

---

## Part 3 — Installation and System Configuration

Part 3 includes the following chapters:

- Chapter 7, “Installation Notes and Warnings”
- Chapter 8, “System Configuration”





---

## Installation Notes and Warnings

Be sure to read this chapter before proceeding with installation of SunOS 4.1.2.

This chapter provides information about the installation of SunOS 4.1.2, including warnings about OpenWindows restrictions. It also contains information about installation of unbundled products, including warnings about the incompatibility of some versions of these products with release software or other Sun products.

### 7.1. Upgrade Utility Available

SunOS 4.1.2 includes `sunupgrade`, a utility for upgrading from SunOS 4.1.1 or SunOS 4.1.1 Rev B to SunOS 4.1.2. See *Installing SunOS 4.1.2 System Software* for a description of the upgrade procedure.

### 7.2. Do Not Install Earlier Patches to SunOS 4.1.2

Many patches that were needed in SunOS 4.1.1 and SunOS 4.1.1 Rev B have been included in SunOS 4.1.2. Check with your local Sun Support center if you are unsure if a particular patch has been included in SunOS 4.1.2 or to see if a SunOS 4.1.2 version of the patch is available.



Do *not* install any patches from earlier releases after installing SunOS 4.1.2; this may cause your system to crash.

### 7.3. Installing Unbundled Products on a Server Supporting Multiple Releases

Clients running a different release than their server do not have access to unbundled products installed in directories belonging to the server's release. In some cases, this means that `cdmanager` or `extract_unbundled`, the installation tools for unbundled products, has to be run separately for each release. In other cases, a workaround of several steps may be necessary in order to make a product available to a client.

Unbundled products are loaded on a system using `cdmanager` or the `extract_unbundled` installation script. For many products, the script allows the user to specify the receiving directory. When this is the case, `cdmanager` or `extract_unbundled` can simply be run more than once to install a given product in different release directories.

For some products, however, the receiving directory is hardcoded into `cdmanager` or `extract_unbundled`. In such cases, `cdmanager` or `extract_unbundled` will load a product in a directory belonging to the server's release, where it is not available to clients running under a different release.

### Workaround to Install Language Products for a Client's Release

The following workaround shows how to install language products so that they are accessible to clients running a different release from their server. The same approach can be applied to other products whose paths are hardcoded in `cdmanager` or `extract_unbundled`.

The workaround uses the example of a sun4c client running SunOS 4.1.2 on a Sun-4 server running under SunOS 4.1.1. All language products belonging to the server's application architecture are loaded into the server's `/usr/lang` directory. It is assumed that `extract_unbundled` has already been run to install language products on the server. A partial file listing of the server's `/export/exec` directory shows:

```
drwxr-sr-x  3 root          512 Jun  8 13:38 kvm
drwxr-sr-x  9 root          512 Mar  6 17:26 proto.root.sunos.4.1.1
drwxr-sr-x  9 root          512 Jun  1 11:38 proto.root.sunos.4.1.2
drwxr-sr-x  3 root          512 Jul 23 15:02 sun3
lrwxrwxrwx  1 root              4 Jun  7 20:34 sun4 -> /usr
lrwxrwxrwx  1 root          4 Jun  7 20:34 sun4.sunos.4.1.1 -> /usr
drwxr-sr-x 29 root        1024 Jun 25 09:14 sun4.sunos.4.1.2
```

To install a language so that it is accessible to a client running a different release than its server:

1. Halt all processes on systems belonging to the server's release and the client's release that might try to access binaries in the server's `/usr/lang` directory. In this example, halt all sun4 and sun4c clients.
2. Change directories so that you are in the server's receiving directory. In this case:

```
# cd /export/exec/sun4.sunos.4.1.1
```

3. Change the name of the language directory:

```
# mv lang lang.old
```

4. If you do not already have a directory for receiving the product on the client system, set it up now. For example:

```
# mkdir /export/exec/sun4.sunos.4.1.2/lang
```

5. Create a link to the client directory that is to receive the product:

```
# ln -s /export/exec/sun4.sunos4.1.2/lang /export/exec/sun4/lang
```

6. Use `extract_unbundled` to install the product.
7. Remove the link from `/export/exec/sun4/lang` to the client's receiving directory:

```
# rm lang
```

8. Restore the original language directory whose name you changed:

```
# mv lang.old lang
```

## 7.4. OpenWindows Version 2 Restrictions

OpenWindows Version 2 and OpenWindows Version 3 are described in Chapter 3. Note the following about OpenWindows Version 2 before you begin installation of the release. Also, see Chapter 10 in the *Open Issues* package for descriptions of OpenWindows bugs.

### Do Not Install Sun IPX Supplement

This patch has been incorporated in the OpenWindows Version 2 included with SunOS 4.1.2; the patch is unnecessary. Do not install it.

### OpenWindows Does Not Fit on 104MB System Disks

OpenWindows requires over 30MB of disk space. This is more than will fit on a 104MB disk loaded with even a minimal version of SunOS4.1.2. See “Desktop SPARCsystem with Two 104MB Disks: Using Second Disk for /home and Additional Swap Space” and “Desktop SPARCsystem with Two 104MB Disks: Using Second Disk for OpenWindows” in Chapter 8 for descriptions of the methods for using a second 104MB disk for either /home or for OpenWindows.

### 8MB Main Memory, 20MB Swap Partition, Required

In order to run OpenWindows, your system must have at least eight megabytes of main memory. Twelve megabytes or more are recommended.

Before installing OpenWindows software on your system, you need to make sure that your swap partition is at least 20MB; 24MB or more is recommended.

### OpenWindows Requires SunView Libraries In Order to Run

In order to run OpenWindows, you must have the SunView\_Users software category installed on your system. SunView\_Users contains libraries needed by OpenWindows. Without SunView, you get the message:

```
ld.so: libsuntool.so.0.## not found
```

### 7.5. Graphics:

A special patch tape (SunOS 4.1 GFX Rev 1) was provided for running GFX graphics under SunOS 4.1; two patch tapes (SunOS 4.1.1 GFX Rev 1 and SunOS 4.1.1 GFX Rev 2) were provided for running GFX graphics under SunOS 4.1.1. SunOS 4.1.2 now incorporates the contents of these tapes.



If you install SunOS 4.1.2 and then try to install any of the SunOS 4.1 or SunOS 4.1.1 GFX patch tapes, you will crash your system.

### 7.6. Hardware Patches

#### Do Not Install SunOS 4.1 Patch for Running SPARCprinter on 4.1.2 System

The *Read This First* (RTF) with Sun's SPARCprinter instructs the user to install a patch for running the printer under SunOS 4.1. The patch is now incorporated in SunOS 4.1.2 and the RTF's instruction should be disregarded.



If you install SunOS 4.1.2 and also try to install the patch from SunOS 4.1 you will destroy your kernel and crash your system.

#### Do Not Install Sun 1.3GB Disk Enhancement

A special patch was provided for Sun-4 systems running SunOS 4.1.1 and Sun-4c systems running SunOS 4.1.1 Rev B, to enable use of the 1.3GB disk on these systems. The changes have been incorporated in SunOS 4.1.2; the patch is unnecessary. Do not install it.



**7.7. Prestoserve 2.0 and 2.3**

Prestoserve 2.0 has been superceded by Prestoserve 2.3, which only supports Sun-4c and Sun-4m machines. If you have a Sun-4 machine, you can continue to use Prestoserve 2.0 with SunOS 4.1.2.

**Error Message During Prestoserve 2.3 Installation**

During installation of Prestoserve 2.3 on a SunOS 4.1.2 system, the following message sequence will be displayed:

```
2.3_Prestoserve : This software is not compatible with the
current operating system

        This is the list of compatible operating systems:
                4.1.1

Do you want to continue? [y|n]
```

Respond “y” and Prestoserve 2.3 will be successfully installed.

**Prestoserve 2.0 Sundiag Problem (1067352)**

The Sundiag menu for Prestoserve 2.0 shows both a user test and device test for Prestoserve. Run only the device test.

**7.8. NSE 1.2 not Compatible with SunOS 4.1.2**

Do not use NSE (Network Software Environment) Release 1.2 on a system running SunOS 4.1.2. If you run NSE 1.2 under SunOS 4.1.2, it will corrupt some system files and may cause a system failure. This problem is corrected in NSE 1.2.1 and subsequent releases.

**7.9. Sun DBE 1.0 and 1.1 Not Supported under SunOS 4.1.2**

Do not use SunDBE 1.0 or SunDBE 1.1 with SunOS 4.1.2; they are not supported and they will not work.

**7.10. Sun Cross Compilers 3.0 Not Supported under SunOS 4.1.2**

Do not use Sun Cross Compilers 3.0 with SunOS 4.1.2; it is not supported.

**7.11. SunTrac Tutorial Date Errors**

The SunTrac tutorial program contains reference to a `Play`, that has its starting date on 01/01/90. Since the real date is now greater than that date, a feature of the product is activated which requires the user to modify a considerable number of dates in the Tutorial. This can hinder testing considerably and create documentation errors. The `Clear Date Error` menu item will clear the date errors each time they are encountered.

## 7.12. SPE 1.1 Problems

The following two problems may be encountered running SPE 1.1 under SunOS 4.1.2; the first problem has been fixed in SPE 1.2.

### SPE Build Failure

If SPE 1.1 is built (using the `$SPE_ROOT_DIR/build-spe` script) under SunOS 4.1.2, it will fail after trying to write the SPE image to a nonexistent directory. That directory will be either

```
$SPE_ROOT_DIRECTORY/sun3-unknown
or
$SPE_ROOT_DIRECTORY/sun4-unknown
```

depending on the application architecture of your system.

There are two ways to avoid this problem. The first solution is to link the nonexistent directory to

```
$SPE_ROOT_DIRECTORY/sun3-4.0
or
$SPE_ROOT_DIRECTORY/sun4-4.0
```

(whichever is appropriate to your system).

*Alternatively*, you can edit the `$SPE_ROOT_DIRECTORY/os` shell script, changing

```
/SunOS Release 4.0/
to
/SunOS Release 4./
```

### Stack Overflows

The second SPE problem is caused by stack overflows, which may cause the Lisp process to die. Specifically, overflowing the stack and then, without first unwinding the stack, overflowing it again (recursive stack overflows) will kill your system. Avoiding these stack overflows, especially recursive stack overflows, will prevent this problem.

## 7.13. SunShield 1.0 on SunOS 4.1.2

To install SunShield 1.0 after installing SunOS 4.1.2, you must extract the patch from the SunOS 4.1.2 CD-ROM, as described below.

1. **Insert the SunOS 4.1.2 CD-ROM into the drive.**
2. **Unmount the CD-ROM if it is mounted:**

```
# cd /
umount /dev/sr0
```

3. **Run `extract_patch`:**

```
# /usr/etc/extract_patch -parm_1_0
```

4. **Change to SunShield (“arm”) directory and untar the patch:**

```
# cd /usr/tmp/arm_1_0
# install_arm_1_0
```

**5. Respond y to question:**

```
You need a ARM/ASET CD to continue.
Do you still want to continue [y|n]? y
```

**6. Respond y when asked if CD is inserted:**

```
Please insert the ARM/ASET CD.
Is the CD inserted? [y]:y
```

**7. Designate cdm or cdmanager as the tool to use:**

```
Which tool do you want to use?
1. cdm
2. cdmanager
Your selection:1
```

You can use `cdm` whether or not you are running OpenWindows; you must be in OpenWindows to run `cdmanager`. Follow the self-explanatory prompts to complete the installation of the patch.

## 7.14. SunNet/SunLink Networking Products

When installing some 6.0 and 6.1 SunNet/SunLink products, you must run the `/usr/etc/sunlink.install` script *after* you run `extract_unbundled`, but *before* you run any product-specific installation scripts. If you do not run `sunlink.install` first, the product-specific installation scripts fail.

The new installation steps are as follows:

1. Run `/usr/etc/extract_unbundled` to extract the product off the tape.
2. Run `/usr/etc/sunlink.install`, selecting the product being installed.
3. Run any product-specific installation or configuration scripts according to the product documentation.

The following list indicates the affected SunLink products. Note that some of the products in the list have been superseded by later versions, which may not cause any installation problems.

```

6.1 BSC3270
6.0 BSCRJE
7.0 Channel Adapter
6.0 DDN
6.0 DNI*
6.0 HSI
6.0 INR*
6.0 MCP*
6.1 SNA3270
6.0 Peer-to-Peer

```

\*There may be additional steps for these products. See the next sections for details.

### Installing the Internetwork Router

If you are installing the SunLink Internetwork Router on a system that is running SunOS 4.1.2, you must perform the following step:

After you run the `extract_unbundled` and `sunlink.install` scripts, but before you run the `install.inr` script, enter the following command as root:

```
# rm /usr/sunlink/inr/sys/arch/OBJ/in_pcb.o
```

where *arch* is a directory name that designates your machine architecture, for example, `sun4` for a Sun-4. You can then proceed to run `install.inr` as documented in the *SunLink Internetwork Router System Administration Guide*.

### Exporting SunLink Software for Multiple SunOS Versions

When you install SunLink software, the installation script installs the software only in the hierarchy for the SunOS version that the machine receiving the installation is running. That is, if you install on a machine running SunOS 4.1.2, the SunLink software goes into the hierarchy for 4.1.2. If that machine is a server that has diskless clients that are running different SunOS versions from the server's and that require access to SunLink software, you must do the following:

- Create a `sunlink` mount point at the end of the version-specific hierarchy that the client mounts on `/usr`.
- Edit the client's `fstab` file so that the client mounts the hierarchy where the SunLink product was installed on the newly-made mount point.

For example, assume you have a Sun-4 server running SunOS version 4.1.2 that serves diskless Sun-4s running SunOS 4.1.2 and SunOS 4.1.1. Your 4.1.1 clients mount one hierarchy as /usr:

```
/export/exec/sun4.sunos.4.1.1
```

while your 4.1.2 clients mount a different hierarchy as /usr:

```
/export/exec/sun4.sunos.4.1.2
```

After you install SunLink software on the server, it is available as:

```
/usr/sunlink/<product_name>
/export/exec/sun4/sunlink/<product_name>
/export/exec/sun4.sunos.4.1.1/sunlink/<product_name>
```

To allow your 4.1.2 clients to access the SunLink software, use the following commands to create the mount point sunlink:

```
server# cd /export/exec/sun4.sunos.4.1.2
server# mkdir sunlink
```

Note that the hierarchy /export/exec/sun4.sunos.4.1.2 is already mounted as /usr on the 4.1.2 clients. In the fstab for these clients, add an entry so that /export/exec/sun4.sunos.4.1.1/sunlink (the hierarchy that contains SunLink software) is mounted on the newly created mount point /usr/sunlink after the /usr entry, for example:

```
<server>:/export/exec/sun4.sunos.4.1.2      /usr      nfs ro 0 0
<server>:/export/exec/sun4.sunos.4.1.1/sunlink /usr/sunlink nfs ro 0 0
```

### setsid Problems When Running SunLink DNI 6.0

If you install SunOS 4.1.2 on a machine running SunLink DNI 6.0, you must provide a “wrapper” around the DNI virtual terminal daemon (dnilogind) so that it conforms to the POSIX- standard requirements for acquiring a controlling terminal. This problem does not exist with DNI 7.0.

To allow a Sun node to remain accessible via SunLink DNI when a set host command is issued on a VAX/VMS† system, as root, enter the following commands after DNI installation is complete:

† VAX and VMS are trademarks of Digital Equipment Corporation.

```
# cd /usr/sunlink/dni
# mv dnilogind .dnilogind
# cat > dnilogind
#!/bin/sh
/usr/etc/setsid -b /usr/sunlink/dni/.dnilogind "$@"
^D
# chmod a+x dnilogind
```

It is not necessary to reboot your machine.

For background on the reasons for this requirement, see the man page on `setsid` (8V).

### setsid Problems When Running SunLink X.25 6.0

If you install SunOS 4.1.2 on a machine running SunLink X.25 6.0, you must provide a “wrapper” around the X.29/X.3 server (`x29`) so that it conforms to the POSIX- standard requirements for acquiring a controlling terminal. This is fixed in X.25 7.0 and subsequent releases.

As root, enter the following commands after X.25 installation is complete:

```
# cd /usr/sunlink/x25
# mv x29 .x29
# cat > x29
#!/bin/sh
/usr/etc/setsid -b /usr/sunlink/x25/.x29 "$@"
^D
# chmod a+x x29
```

It is not necessary to reboot your machine.

For background on the reasons for this requirement, see the man page on `setsid` (8). (This man page is not present in previous SunOS versions.)

### Incorrect Instruction in SunLink BSC3270 System Administration Guide (1044933)

On page 35 of the *SunLink BSC3270 System Administration Guide* the instructions for booting from the `newvmunix` kernel are incorrect and will generate incorrect output from a number of subsequent commands. The correct instructions are:

1. Copy the new kernel, `newvmunix`, to the root (/) directory.
2. Change the name of the original `vmunix` kernel to `vmunix.old` and change `newvmunix` to `vmunix`:

```
# mv vmunix vmunix.old
# mv newvmunix vmunix
```

3. Reboot your system

## 7.15. Language Products

The following topics are described in this section:

- Sun C++ 2.1 Patch Installation
- FORTRAN 1.4 Patch Installation
- Pascal 2.1 Patch Installation
- FORTRAN, C, Pascal, Modula-2: Missing Debugging Libraries
- FORTRAN 1.2 Problems

### Sun C++ 2.1 Patch Installation

SunOS 4.1.2 introduces new library functions and system calls. For Sun C++ 2.1 to run under SunOS 4.1.2 you must install a patch. This section describes how to install the patch.

The patch includes the following:

- A README file,
- The patch installation script
- A directory with header files for SunOS 4.1.2.

### Where is the Sun C++ Patch Installed?

Install the patch in the directory in which Sun C++ 2.1 was previously installed. The default is `/usr/lang` for standalone and homogeneous servers. You may also specify a non-default directory as the installation directory. Just make sure it exists and that you have already installed Sun C++ 2.1 in it; otherwise, the installation script will not allow you to install the patch.

The patch script will alter the original `SC1.0/include/CC` header file directory, and some of the original `SC1.0/include/CC` header files *will be lost*.



*Install or upgrade to SunOS 4.1.2 and Sun C++ 2.1 before you install the Sun C++ 2.1 patch.*

If you do not understand the terms used here, see the installation instructions in the *Sun C++ 2.1 Read This First*.

### Preparing for Patch Installation

If you are installing on a standalone workstation and using the default installation directory, you may use default installation directions. If you are going to install the software onto a server, follow the non-default installation directions. If you need additional information, see the `man` page for `extract_patch(8)`.

### Patch Installation to Default Directory

After installing SunOS 4.1.2 and C++ 2.1 from the SunOS 4.1.2 CD-ROM, do the following to install the C++ patch in `/usr/lang`, the default installation directory for C++ 2.1.

1. **Insert the SunOS 4.1.2 CD-ROM into the CD-ROM drive and mount the CD-ROM.**

2. `su` to root and extract the patch directory by typing the following on a command line:

```
# /usr/etc/extract_patch -DEFAULT -pcplusplus_2_1
```

The script will extract the patch files from the CDROM; if the default destination, `/usr/lang`, exists and contains the directory `SC1.0/include/CC`, the script will display the README file and ask if you want to continue:

```
Ready to install C++ 2.1 patch for 4.1.2 in /usr/lang
Do you want to continue: [y|n]?
```

3. Enter `y`:

```
Do you want to continue: [y|n]? y
Doing the patch.....
Done!
```

Sun C++ 2.1 should now work under SunOS 4.1.2; if it does not, contact your local Sun Answer Center.

#### Patch Installation to a Non-Default Directory

After installing SunOS 4.1.2 and C++ 2.1 from the SunOS 4.1.2 CD-ROM, do the following to install the C++ patch to a directory other than `/usr/lang`, the default directory.

1. `su` to root and extract the patch directory by typing the following on a command line:

```
# /usr/etc/extract_patch -pcplusplus_2_1
```

The script will extract the patch files from the CDROM, display the README file, and ask you whether you want to execute the patch installation program. Enter `y`.

2. Reply `y` when asked if you want to see a description of the patch script:

```
Do you want to see a description of this patch script [y|n]? y
Patch of Sun C++ FCS release 2.1 for SunOS 4.1.2
Patching should take approximately 1 minute.
```

3. If you are installing the patch on a machine that is running on SunOS 4.0, 4.1, or 4.1.1 you will see the following message:



```
WARNING: This patch script is for Sun OS 4.1.2
You are running it on Sun OS 4.0 or 4.1 or 4.1.1
Do you want to continue anyway [y|n]?
```

Enter **y** if you are installing the patch into a server running on SunOS 4.0, 4.1, or 4.1.1 that will support machines running on SunOS 4.1.2.

4. Respond to the script's questions about what type of system you have (standalone or server), what type of server (always choose homogeneous), and what type of client will the product run on (sun4 only).

After you answer the questions, the script will display the default Sun C++ 2.1 directory for your configuration:

```
Currently the default C++ 2.1 directory is /usr/lang
Do you want to change the default directory [y|n]?
```

5. Enter **n** if the default directory is correct. Enter **y** if you want to specify a different directory. You will then be asked to type in the name of your installation directory.
6. See Step 2 under the "Default Patch Installation," the script will continue in the same manner.

Sun C++ 2.1 should now work under SunOS 4.1.2; if it does not, contact your local Sun Answer Center.

## **FORTRAN 1.4 Patch Installation**

Install the FORTRAN patch in the directory where FORTRAN is installed; /usr/lang/SC1.0 or /usr/lang/SC1.0.1 is the default location. You may also specify a non-default directory as the installation directory. Just make sure it exists and that you have already installed SC1.0 or SC1.0.1 FORTRAN in it; otherwise, the installation script will not allow you to install the patch.

The patch installation requires 3.8MB in /usr/tmp and 3.8MB in /tmp. Do the following to install the patch:

1. **Become root and extract the patch directory by typing the following on a command line:**

```
# /usr/etc/extract_patch -DEFAULT -pfortran_sc1_0
```

This script will extract the patches from the CD-ROM and place them into /tmp/fortran\_sc1\_0.

2. **Change to the directory just created:**

```
# cd /tmp/fortran_sc1_0
```

3. **Copy the files in /tmp/fortran\_sc1\_0 to <langdir>/SC1.0, where <langdir> is the location you installed the FORTRAN compiler (default is**

`/usr/lang`).

4. **Replace the libraries in `SC1.0` or `SC1.0.1` with those in `/tmp/fortran_sc1_0`**
5. **Execute `ranlib -t` on all of the `.a` and `.sa` files copied.**

### Pascal 2.1 Patch Installation

Install the Pascal patch in the directory where Pascal is installed; `/usr/lang/SC1.0` or `/usr/lang/SC1.0.1` is the default location. You may also specify a non-default directory as the installation directory. Just make sure it exists and that you have already installed SC1.0 or SC1.0.1 Pascal in it; otherwise, the installation script will not allow you to install the patch.

Do the following to install the patch:

1. **Become root and extract the patch directory by typing the following on a command line:**

```
# /usr/etc/extract_patch -DEFAULT -ppascal_sc1_0
```

This script will extract the patches from the CD-ROM and place them into `/tmp/pascal_sc1_0`.

2. **Change to the directory just created:**

```
# cd /tmp/pascal_sc1_0
```

3. **Copy the files `lipbc.so.2.2.1` and `lipbc.sa.2.2.1` to `<langdir>/SC1.0`, where `<langdir>` is the location you installed the Pascal compiler (default `/usr/lang`).**
4. **Run `ranlib`:**

```
# ranlib -t <langdir>/SC1.0/lipbc.sa.2.2.1
```

### FORTRAN, C, Pascal, Modula-2: Missing Debugging Libraries

If you are trying to debug or profile with C, FORTRAN, Pascal, or Modula-2, you may get an error message indicating that a particular function is missing. This may be missing profiling or debug libraries.

For example, if the profiling libraries are not loaded, you will get a message as shown below.

```
% f77 -p test.f
test.f:
  MAIN bork:
ld: -lc_p: No such file or directory
%
```

Ask your system administrator to help you install the missing libraries.

**FORTRAN 1.2 Problems**

The installation of FORTRAN 1.2 may fail on SunOS release 4.1.2. This problem is fixed in FORTRAN 1.4.

To fix this problem, modify the `/usr/tmp/1.2_fortran` file and restart the installation as indicated below.

1. Stay logged in as superuser.
2. Change directory to `/usr/tmp`

```
% cd /usr/tmp
```

3. In any editor, revise the `/usr/tmp/1.2_fortran` file as follows:

```
Change:  SOS_COMPAT="4.0"
to:      SOS_COMPAT="4.1.2 4.1.1 4.0"
```

and save the file.

4. Issue the following command:

```
% /usr/tmp/unbundled/1.2_fortran -rrmt_host -ddev
```

Where *rmt\_host* is the name of the remote host if the tape is mounted remotely, and *dev* is the device specification (`st0`, `mt1`, etc.).

5. Restart the installation.

```
% install_unbundled -f
```

**7.16. Type-4 Keyboard Compatibility with Unbundled Products**

Some unbundled software products make full use of native-language keyboards, while others will add national language support in future releases. The following lists show the status of major software products at the current time.

Initial testing indicates that the following products provide full national language support when used with native language keyboards. Note that in most compilers, extended characters may not be used in variable names.

FORTRAN 1.2	Pascal 2.0	Modula-2 2.1
C 1.0	C++ 2.0	X Compilers 3.0
DeskSet 1.0	SunGKS 3.0	SunPHIGS 1.1
IR 6.0	HSI 6.0	DDN 6.0
MCP 6.0	FDDI 1.0†	NSE 1.2†
SunNet Manager 1.0	SunNet License 1.0	Transcript 2.1.1

†Not supported in SunOS release 4.1.2.

The following products will correctly recognize any of the national keyboards, but will not handle 8-bit characters. Behavior when an accented character is typed is unpredictable; some will produce graphics characters and some will ignore the keystroke.

SunWrite†/Paint/Draw 1.1	SunTrac 1.3	BSC3270 6.1
BSCRJE 6.0	CG3270 6.0	SNA Peer 6.0
Local 3270 6.1	Channel 7.0†	SNA3270 6.1
X.25 6.0	OSI 6.0	MHS 6.0
SunCobol 1.0	SCLisp 3.0	NetISAM 1.0.DE

### NeWS 1.1 on Type-4 Keyboards

When NeWS 1.1 was released, Type-4 keyboards did not exist. In order to use NeWS 1.1 with a Type-4 keyboard, the following patch is required. The patch causes NeWS to treat a Type-4 keyboard as a Type-3.

```

$NEWSHOME/lib/NeWS/UI.ps.

The patch is as follows (context "diff" file):
cd $NEWSHOME/lib/NeWS
*** UI.ps-      Wed Jan 18 11:49:15 1989
--- UI.ps       Tue Mar  7 12:27:00 1989
*****
*** 174,179 ****
--- 174,180 ----
    /KB_VT100    1      def
    /KB_SUN2     2      def
    /KB_SUN3     3      def
+   /KB_SUN4     4      def
    /KB_ASCII   15     def

    /TR_NONE     0      def
*****
*** 699,704 ****
--- 700,708 ----
    (NeWS/sun1_keys.ps) run
  }
  KB_SUN3 {
    (NeWS/sun3_keys.ps) run
  }
+  KB_SUN4 {
+    (NeWS/sun3_keys.ps) run
+  }
    /Default {

```

---

## System Configuration

This chapter explains and outlines some things you can do to improve system performance on Desktop and server systems; the chapter is comprised of the following sections:

- About Kernel Configuration
- Performance Improvement Hints
- Device Configuration on Open Boot PROM Systems
- Customizing the Kernel for a Large Server
- Disk Layout for Systems with 104MB Disks
- Booting from IPI Disks on SPARCsystem 600MP Systems

### 8.1. About Kernel Configuration

Why would you want to reconfigure and rebuild your kernel? To save main memory (and improve performance) by specifying precisely what is needed for the applications used, and to add any required drivers and modules.

When you install SunOS 4.1.X, you use the large `GENERIC` kernel by default. The `GENERIC` kernel supports all Sun hardware and software features, including many that are probably not necessary for your system. Customizing it or using an alternative kernel can significantly reduce memory requirements and improve performance. It is highly recommended that you reconfigure the `GENERIC` kernel as soon as possible after installation. Kernel configuration options are briefly described below. For detailed information, see *System and Network Administration*.

There are three basic alternatives to the `GENERIC` kernel configuration file:

- Install one of the Sun-supplied preconfigured `GENERIC_SMALL` kernel configuration files as is. This is the simplest alternative, but is likely to save less memory than the others.

You can install a `GENERIC_SMALL` kernel with the `install_small_kernel` script, which can be run from the miniroot or from a shell as superuser. Standalone systems can use the script to install their own kernels; servers can use it to install small kernels for their clients.

- Build a custom kernel using one of the Sun-supplied kernel configuration files as a template.

- Build a completely customized kernel by editing the `GENERIC` configuration file yourself. This is the most complex of the three alternatives, but may improve performance the most.

### Sun-Supplied Kernel Configuration Files

Kernel configuration files are templates that make it easier to customize your system's kernel. The kernel configuration files for a standalone system or the kernel architecture of a server are located in:

```
/usr/kvm/sys/k-arch/conf
```

– *k-arch* is replaced by a kernel architecture: sun4, sun4c, sun4m

The kernel configuration files for the clients of a server are in:

```
/export/exec/kvm/k-arch/sys/k-arch/conf
```

– *k-arch* is replaced by kernel architecture (sun3, sun3x, sun4, sun4c, sun4m)

The tables that follow list the kernel configuration files supplied for each of the SPARC kernel architectures. These are in addition to the `GENERIC` configuration files, of course.

Table 8-1 *Sun-Supplied Kernel Configuration Files for sun4c Architectures*

Configuration File Name	Supported Architecture
GENERIC_SMALL	Desktop SPARCsystem with up to 8 SCSI Disks, 4 SCSI Tapes, 2 CD-ROM
DL60	Diskless Desktop SPARCsystem (Does not support SCSI Devices)
NFS60	Desktop SPARCsystem with up to 8 SCSI Disks, 4 SCSI Tapes, 2 CD-ROM Configured to Boot from NFS Filesystems
DLS60	Desktop SPARCsystem with one SCSI Disk, Configured to Boot from Network, but Use Local Swap
SDST60	Desktop SPARCsystem with up to 8 SCSI Disks, 2 SCSI Tapes, 2 CD-ROM

Table 8-2 *Sun-Supplied Kernel Configuration Files for sun4 Architectures*

Configuration File Name	Supported Architecture
GENERIC_SMALL	Sun-4/110 or 4/330 with up to 4 SCSI Disks and 2 SCSI Tapes
DL	Diskless Sun-4/260, 4/280, 4/110, or 4/330
DL110	Diskless Sun-4/110
DL330	Diskless Sun-4/330
SDST110	Sun-4/110 with up to 4 SCSI Disks and 2 SCSI Tapes
SDST330	Desktop SPARCsystem with up to 4 SCSI Disks and 2 SCSI Tapes
XDXT260	Sun-4/260 with up to 2 SMD-4 Controllers, 2 Xylogics 1/2" Tape Drives, 2 SCSI Disks, and 1 SCSI Tape
XYXT260	Sun-4/260 with up to 2 Xylogics 540/451 Controllers, 2 Xylogics 1/2" Tape Drives, 2 SCSI Disks, and 1 SCSI tape
IDST390	Sun-4/390 with 2 IPI Controllers with 8 drives each and 4-SCSI tape devices
IDST490	Sun-4/490 with 4 IPI Controllers with 8 drives each and 8 SCSI tape devices
SDST470	Sun-4/470 with up to 4 SCSI or IPI Controllers, with maximum of 8 drives across controllers, 8 SCSI tape devices

Table 8-3 *Sun-Supplied Kernel Configuration Files for sun4m Architecture*

Configuration File Name	Supported Architecture
GENERIC_SMALL	Sun-4m with up to four users
DL	Diskless SPARCsystem 600MP
NFS	Sun-4m with SCSI disks and tape; boot from server
SDST	Standalone Sun-4m with SCSI disks and tape
IDST	Sun-4m with IPI disks and SCSI tape

## 8.2. Device Configuration on Open Boot PROM Systems

The process of rebuilding the kernel of an open boot PROM system (sun4c, sun4m) is much simpler than for other Sun systems running SunOS 4.1.2. The PROM monitor for open boot PROM, in conjunction with the kernel, eliminates the need for most of the device-specification lines in kernel configuration files.

Standard SunOS kernel configuration files contain many lines describing bus connections, controller addresses and slave units on controllers. An open boot PROM kernel uses simple declarations for describing devices. For example, in the sun4c GENERIC configuration file, the following lines are all that is necessary to specify the inclusion of device drivers and kernel support for them; the PROM and the kernel auto-configuration code do the rest.

```

device-driver sbus      # 'driver' for SBus interface
device-driver bwtwo    # monochrome frame buffer
device-driver cgthree  # 8 bit color frame buffer
device-driver cgsix    # 8 bit accelerated color frame buffer
device-driver dma      # 'driver' for dma engine on SBus interface
device-driver esp      # Emulex SCSI interface
device-driver fd       # Floppy disk
device-driver audio    # sound chip
device-driver le       # Lance ethernet
device-driver zs       # UARTs

```

The `flags` word that was formerly used to specify ignoring CARRIER DETECT for the `ZS` (UART) driver has been replaced by data in the EEPROM, which the kernel auto-configuration code asks the PROM to fetch. The fields (`ttya-ignore-cd` and `ttysb-ignore-cd`) may be set either using either the `eeprom(8S)` command, or commands from the PROM monitor (see `monitor(8S)`).

### Declaring SCSI Buses, Disks, and Tapes

The kernel configuration file also declares the SCSI buses, disks, and tapes that may be connected to the system. This must be user-entered, rather than determined by auto-configuration, because Sun supports some non-CCS (Command Command Set) devices which do not respond to the SCSI INQUIRY command (which would normally determine what kind of a device it is).

Two steps are involved. The existence of one or two SCSI buses connected to the system must be declared, and disks and tapes must be identified and associated with their bus.

The following screen shows the entries for declaring a first (`scsibus0`) and second (`scsibus1`) SCSI bus:

```

scsibus0 at esp
scsibus1 at esp

```

The disks and tapes associated with a SCSI bus are declared as shown in the following example. The example gives GENERIC kernel configuration file default declarations for the first SCSI bus.

```

disk sd0 at scsibus0 target 3 lun 0 # first hard SCSI disk
disk sd1 at scsibus0 target 1 lun 0 # second hard SCSI disk
disk sd2 at scsibus0 target 2 lun 0 # third hard SCSI disk
disk sd3 at scsibus0 target 0 lun 0 # fourth hard SCSI disk
tape st0 at scsibus0 target 4 lun 0 # first SCSI tape
tape st1 at scsibus0 target 5 lun 0 # second SCSI tape
disk sr0 at scsibus0 target 6 lun 0 # CD-ROM

```



To be more specific, the first line above says that there may be a disk, `sd0`, on `scsibus` number 0, at SCSI target address 3, logical unit 0. These declarations merely state that this device *may* be at this location; look for it when booting, and (if not found) look again if a program attempts to open it while the system is running.

The default declarations in the GENERIC kernel configuration file for disks and tapes on a second SCSI bus are:

```
disk sd4 at scsibus1 target 3 lun 0 # fifth hard SCSI disk
disk sd5 at scsibus1 target 1 lun 0 # sixth hard SCSI disk
disk sd6 at scsibus1 target 2 lun 0 # seventh hard SCSI disk
disk sd7 at scsibus1 target 0 lun 0 # eighth hard SCSI disk
tape st2 at scsibus1 target 4 lun 0 # third SCSI tape
tape st3 at scsibus1 target 5 lun 0 # fourth SCSI tape
disk sr1 at scsibus1 target 6 lun 0 # second CD-ROM device
```

The default declarations for a third and fourth SCSI bus can be found in the GENERIC kernel configuration file for your Sun-4c or Sun-4m system:

```
/usr/sys/sun4c/conf/GENERIC
/usr/sys/sun4m/conf/GENERIC
```

### Booting from IPI Disks on SPARCsystem 600MP Systems

If you intend to boot or reboot a 4.1.2 based Sun-4m system from an IPI disk device, you must make sure that certain "pseudo" device drivers are included in the kernel configuration file, in addition to the standard `ipi`, `is`, `idc`, `id`, and `ipibus` configuration information that is required to include device driver support for the IPI disk devices.

The additional `pn`, `ipisc`, and `idpseudo` devices, which are included in the standard Sun-4m GENERIC configuration file, interact with the open boot PROM at boot and reboot time to obtain and provide self-identifying information. The following entries must be in the kernel configuration file:

```
device-driver pn          # Self-ident. VME/IPI pseudo driver
device-driver ipisc       # Self-ident. IPI-3 Slave Cntrl pseudo driver
device-driver idpseudo   # IPI Disk pseudo device driver
```

If you are not using the GENERIC kernel, or if you have removed the above lines from the GENERIC kernel, be sure to include the lines in the kernel configuration file before making a new kernel.

### 8.3. Customizing the Kernel for a Large Server

On large servers it may be necessary to adjust two basic parameters in the kernel configuration file:

- `maxusers`
- number of ports (for systems with ALM-2's)

#### Maximum `maxusers` Values for Sun-4, Sun-4c, Sun-4m Servers

For best performance, you want to set the `maxusers` value in the kernel configuration file to a high value. Use the formula described below to determine the `maxusers` value, but be aware of the following limitations; if you exceed these values, your kernel may not boot.

Table 8-4 *Maximum `maxusers` Values for Sun-4, Sun-4c, Sun-4m*

Architecture	64MB RAM or less	128MB RAM	640MB RAM	2.5GB RAM
<b>sun4</b>	225	225	NA	NA
<b>sun4c</b>	225	225	NA	NA
<b>sun4m</b>	185	180	155	45

Note that the maximum `maxusers` value for a SPARCsystem 600MP system is between 45 and 185, depending on the system configuration — the more memory, the lower the `maxusers` value. These `maxusers` values correspond to 1024 and 3253 processes, respectively.

#### Formula for Determining `maxusers`

The `GENERIC` kernel configuration file in the directory `/usr/kvm/sys/k-arch/conf` sets a default value for `maxusers` that is too small for large servers, and is more suitable for personal-use workstations. Check the file for the line

```
maxusers          maxusers_value
```

and, if necessary, increase `maxusers_value` to reflect the actual load on the server. The number of streams allocated is based on this number, so for proper allocation, any `getty`'s running on serial ports should be considered active sessions, and if any lines are running both dial-in and dial-out service via the upper 128 minor numbers, the dial-out should be considered an additional user.

The calculation of `maxusers` is generally:

number of framebuffer sessions (i.e., windows and other tools, or one for a nonwindow login on the console),

plus

number of network sessions (telnet, ftp, rsh, and rlogin sessions to or from this host)

plus

number of serial ports with `getty`'s running on them,

plus

maximum number of concurrent dialout (`tip` and `uucp`) sessions.

Then add a few, and round the figure upward, because you will probably underestimate, and the cost of the extra kernel size is not much when you have large memories (32 MB and up) and disks (600 MB and up). In general, systems with eight ALM-2 boards will also tend to have larger physical memories and larger, faster disks, so setting `maxusers` on the high side is usually preferable to setting it too low. See limits on `maxusers` size in the table above, however.

### Adjust Number of ALM-2 Ports

In the kernel configuration file, the line

```
pseudo-device  mcpa64
```

needs to be adjusted to include all ALM-2 serial ports that are likely to be used (if you have no ALM-2's, you can delete the line or comment it out). For any kernel that is expected to support any of the upper four ALM-2 boards (`mcp4` through `mcp7`), this line should be changed to

```
pseudo-device  mcpa128
```

to provide `async` protocol support for the whole set. Note that if you use a higher numbered ALM, then the `mcpa` number must be great enough to handle that ALM and all lower numbered boards, installed or not.

### Create `/dev` Entries

When adding ALM-2's, you must create the `/dev` entries, as follows:

```
# cd /dev
# MAKEDEV mcp0 mcp1 mcp2 ...
```

where an `mcpn` entry is specified for each ALM-2, up to a max of `mcp7`. The `/dev` entries created by this command are the names which must be entered into `/etc/ttytab`.

**Make Entries in**`/etc/ttytab`

You will have to manually make entries in the `/etc/ttytab` file. See the *System & Network Administration* manual, Section 11.3, "Adding a Terminal to your System."

The procedure for adding `tty` ports:

1. Determine the total number of logins to be supported.
2. Apply the algorithm described above for computing `maxusers`.
3. Rebuild the kernel, if required.
4. Make entries in `/dev` for the new `tty` ports.

Do this with `/dev/MAKEDEV`. Each argument to `MAKEDEV` represents one peripheral board, e.g., "`MAKEDEV mcp0`" means make all `tty` port entries (16) for the first ALM-2, "`MAKEDEV mcp7`" means make all `tty` port entries (16) for the 8th ALM-2 board.

5. Make entries in `/etc/ttytab` for each `tty` port. The format is shown in *System & Network Administration*, section 11.3.

Sample `/etc/ttytab` entries:

```
ttyh0  "/usr/etc/getty std.9600"      unknown      on secure
ttyh1  "/usr/etc/getty std.9600"      unknown      on secure
ttyh2  "/usr/etc/getty std.9600"      unknown      on secure
ttyh3  "/usr/etc/getty std.9600"      unknown      on secure
ttyh4  "/usr/etc/getty std.9600"      unknown      on secure
ttyh5  "/usr/etc/getty std.9600"      unknown      on secure
ttyh6  "/usr/etc/getty std.9600"      unknown      on secure
ttyh7  "/usr/etc/getty std.9600"      unknown      on secure
ttyh8  "/usr/etc/getty std.9600"      unknown      on secure
ttyh9  "/usr/etc/getty std.9600"      unknown      on secure
ttyha  "/usr/etc/getty std.9600"      unknown      on secure
ttyhb  "/usr/etc/getty std.9600"      unknown      on secure
ttyhc  "/usr/etc/getty std.9600"      unknown      on secure
ttyhd  "/usr/etc/getty std.9600"      unknown      on secure
ttyhe  "/usr/etc/getty std.9600"      unknown      on secure
ttyhf  "/usr/etc/getty std.9600"      unknown      on secure
ttyi0  "/usr/etc/getty std.9600"      unknown      on secure
```

**Possible Error Messages**

If too many `/etc/ttytab` terminals are enabled, or too many remote logins occur, compared to the `maxusers` setting in the `config` file, the message

```
stropen: out of streams
```

may be displayed on the console and in the `/usr/adm/messages` log. This is an indication that you should increase the `maxusers` value, and re-build the kernel.

After configuring, building, and booting the new kernel, start up the system and run it for a while. Any messages relating to exhaustion of streams resources are a

red flag that you may need to refigure the correct value for `maxusers`.

## Monitoring Performance

Monitor long term performance with `netstat -m`, and reduce resource allocations if it is shown that peak allocations never get near the allocated maxima. You will want to do this re-allocation of resources if the performance of the machine is suffering due to lots of memory being hogged by the kernel — especially if the cost of reconfiguring the kernel is low (low impact on users).

Bear in mind that allocation of data buffer resources may start to fail when the high water mark reaches 80% of the configured maximum, as the system tries to reserve some resources for high priority messages. No matter how careful you are to watch the resources, you always need some extra room, so don't tune your system too tightly; tune it so that the observed maximum numbers are between 50% and 75% of the absolute limit values in your kernel configuration file.

## 8.4. Performance Improvement Hints

Consider the following ideas for improving performance.

### Use the `tmpfs` Filesystem

The `tmpfs` filesystem allows a system's virtual memory resources to be used as a filesystem. Files and directories can be created and deleted with normal UNIX semantics. The `tmpfs` filesystem does not require additional disk space and allows data to be accessed quickly. `tmpfs`-mounted directories appear identical to standard UNIX filesystems to users and most UNIX utilities.

For details on using the `tmpfs` filesystem, see the *System and Network Administration* manual and the `tmpfs(4S)` man page.



The `tmpfs` filesystem is commonly used to improve the performance of the `/tmp` directory. However, note that unmounting `/tmpfs` or rebooting your system removes all files under `/tmpfs`, including those in subdirectories.

### Use Static Routing When Possible

Workstations that have only one Ethernet interface and do not act as NFS servers can do static routing, rather than dynamic routing with `in.routed`.

To institute static routing, “comment out” (place a “#” in front of) lines in your `/etc/rc.local` file so that they appear as shown:

```
#if [ -f /usr/etc/in.routed ]; then
#   in.routed; echo -n ' routed'
#fi
```

Routing table entries will still be added or modified by the kernel as a result of ICMP redirect messages.

Diskless clients have a route provided automatically by the server. On other workstations, a command of this form can be added to `rc.local` just after the `in.routed` information:

```
/usr/etc/route add default router 1
```

This action frees up both the pages used by `in.routed` and most of the memory allocated for routing table entries.

### Do Not Enable Process Accounting

Accounting is not enabled unless it is configured into the kernel (options `SYSACCT`) or there is a `/var/adm/acct` directory when your system boots. See the man page for `rc(8)` for more information.

### Eliminate Unnecessary Server Processes

Workstations typically only require the following server processes:

- portmap
- ypbind
- biod (four processes)
- syslogd
- update
- inetd
- lpd
- sendmail
- keyser (if you use secure NFS)

Eliminating any additional server processes will increase performance.

### Do Not Enable File-System Quotas

Initially, the default is for file-system quotas **not** to be enabled. If quotas have been enabled, replace `/usr/ucb/quota` with `/usr/bin/true`. This will prevent possible delays at login time due to calls to `rpc.rquotad` on each NFS server from which you have a file system mounted.

## 8.5. Disk Layout for Systems with 104MB Disks

Two procedures are described in this section:

- Desktop SPARCsystem with Two 104MB Disks: Using Second Disk for `/home` and Additional Swap Space
- Desktop SPARCsystem with Two 104MB Disks: Using Second Disk for OpenWindows

### Desktop SPARCsystem with Two 104MB Disks: Using Second Disk for `/home` and Additional Swap Space

This section describes how to set up a `/home` filesystem on the `g` partition of your second disk and how to use your second disk for added swap space. Note that you cannot use `sd1g` for both `/home`, as described here, and for OpenWindows, as described in section 8.5.4 below; you must choose between one use or the other.

A Desktop SPARCsystem with preinstalled SunOS 4.1.2 on a 104 megabyte disk does not have a separate `/home` filesystem for local text and data files. “home” files are maintained in subdirectories of the `/usr` filesystem, which they share with unbundled and third-party application software. What may appear online as a separate `/home` filesystem is really a “symbolic link” to files in `/usr/export/home`.

If you use space on a second disk as a `/home` filesystem, more space is available for application software on the first disk and for local user files on the second disk.

You may need to use your second disk to provide extra swap space; consider this if you start getting the following error message:

```
Not Enough Memory
```

when you start up a program. In many cases, additional swap space improves system performance, even without an initial shortage.

### Using the `g` Partition on Your Second Disk as `/home`

To use the `g` partition of your second disk as `/home`:

1. As a precaution, if there are files in `/usr/export/home` on your first disk, use `dump(8)` to make backup copies.
2. Become superuser and create a new filesystem in partition `g` on your second disk:

```
%su
#newfs /dev/rsd1g
```

3. Create an entry in your `/etc/fstab` file so that the new filesystem is automatically mounted as `/home` when you boot your system:

```
#/dev/sd1g /home 4.2 rw 1 3
```

4. If you have user accounts and files in `/usr/export/home`, transfer them to the `g` partition of your second disk:

```
#mount /dev/sd1g /mnt
#cd /usr/export/home; tar cfh - . | (cd /mnt; tar xpf -)
```

5. List the files copied to the new filesystem to make sure the transfer took place correctly:

```
#ls -lR /mnt
```

6. If the files were copied to `/mnt` as desired, you can now remove them from `/usr/export/home`, freeing the space they occupied for other software:

```
#rm -rf /usr/export/home/*
```

- Remove /home as a link to /usr/home and create a new /home directory as a mount point for the /home filesystem on the second disk:

```
#rm /home
#mkdir /home
```

- Mount the /home filesystem on your second disk on the /home directory of your first disk:

```
#mount /home
```

The procedure given above is an easy way to set up a home filesystem on your second disk. It is possible to move other filesystems to your second disk, to change the sizes of the given partitions, and to use partitions other than `g` for /home. For further information on this, see *System and Network Administration* and *Installing SunOS System Software*.

### Using Your Second Disk for Added Swap Space

To use the `b` partition of your second disk for additional swap space:

- Become superuser and create an entry in your `/etc/fstab` file so that whenever you boot, the `b` partition of your second disk is automatically mounted for use as swap space in addition to the swap space on your first disk. The entry for your `fstab` file is:

```
/dev/sd1b swap swap rw 0 0
```

- To turn swapping from your second disk on, `su` to `root` and type:

```
#swapon -a
```

On your second disk, you can use partitions other than `b` for swap space. To do so, replace `sd1b` in the preceding steps with `sd1` and the name of the partition you want to use for swap.

### Desktop SPARCsystem with Two 104MB Disks: Using Second Disk for OpenWindows

If you use the first 104MB disk of a Desktop SPARCsystem as the system disk for SunOS 4.1.2 release software, there will not be enough room for OpenWindows. This section describes how to use the `g` partition of your second disk (`sd1g`) for OpenWindows.

If you put OpenWindows in `sd1g`, you will have about 21MB of space in `/usr` on your first disk for use by `/home`.



Experienced users who want to have more options in using their second disk should refer to the *System and Network Administration* manual, which provides detailed information on disk partitioning, mounting partitions, and using symbolic links.

Note that you cannot use `sd1g` for both `/home`, as described above, and for OpenWindows, as described below; you must choose between one use or the other.

The procedures you need to follow in using `sd1g` for OpenWindows depend on whether or not you use SunInstall to install SunOS 4.1.2 release software.

### If You Use SunInstall

If you use SunInstall, you can set up a `/usr/openwin` filesystem in `sd1g` from the start; see *Installing SunOS System Software* for instructions on setting up file systems on a second disk. After using SunInstall you can load OpenWindows in `sd1g` by following the instructions given in the section “Installing the OpenWindows Software in `sd1g`,” below.\*

If you have the preinstalled version of SunOS 4.1.2 on your first disk or you use re-preinstall or Quick Install, you will need to create a `/usr/openwin` filesystem in `sd1g` and carry out additional steps before you can load OpenWindows into `sd1g`.

### Systems with SunOS 4.1.X through Preinstallation, Quick Install, or Re-preinstall

To use the `g` partition of your second disk for OpenWindows:

1. Become superuser (you will need to be superuser for the remaining steps) and create a new filesystem in `sd1g`. This will be your OpenWindows filesystem.

```
%su
Password: [enter root password]
#newfs /dev/rsd1g
/dev/rsd1g: 160230 sectors in 763 cylinders of 6 tracks, 35 sectors
          82.0MB in 48 cyl groups (16 c/g, 1.72MB/g, 768 i/g)
super-block backups (for fsck -b #) at:
 32, 3440, 6848, 10256, 13664, 17072, 20480, 23888, 26912, [etc.]
. . .
```

Note: Sample display; contents may vary, depending on user input and the system used.

2. Use `fsck` to check the new filesystem:

\* Do not try to load OpenWindows into `sd1g` with SunInstall. SunInstall will balk, with the message that there is insufficient space.

```
#fsck /dev/rsd1g
** /dev/rsd1g
** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
2 files, 9 used, 74713 free (17 frags, 9337 blocks, 0.0% fragmentation)
```

- Transfer the contents of /usr/openwin on your first disk to your new filesystem in sd1g. These contents, subdirectories and symbolic links for use by OpenWindows, were preinstalled or set up when you used Quick Install or re-preinstall.

```
#mount /dev/sd1g /mnt
#cd /usr/openwin
#tar cfh - . | ( cd /mnt; tar xpf - )
```

- Make sure the transfer was carried out correctly; compare the contents of /usr/openwin and sd1g:

```
#ls -F /usr/openwin
bin/      etc@     lib/     man@     sys/
demo/     include@ local/   share/
#ls -F /mnt
bin/      etc@     lib/     man@     sys/
demo/     include@ local/   share/
```

- If the contents matched, remove the contents of /usr/openwin from /usr on your first disk:

```
#rm -rf /usr/openwin/*
```

- Edit your /etc/fstab file so that your OpenWindows filesystem on sd1g is automatically mounted as /usr/openwin whenever you boot your system. Edit /etc/fstab and add the line:

```
/dev/sd1g /usr/openwin 4.2 rw 1 2
```

- Mount sd1g on /usr/openwin:

```
#mount /usr/openwin
```

You are now ready to load the OpenWindows software. The remaining steps are the same as those for systems that defined the `/usr/openwin` partition using `SunInstall`.

### Installing OpenWindows Software in `sd1g`

To install OpenWindows in the filesystem you created in `sd1g`, change directories to `/usr` and use the new `extract_files(8)` command to load the OpenWindows software categories from your SunOS 4.1.1 release media.\* You must be superuser to use `extract_files`.

```
#cd /usr
#/usr/etc/install/extract_files sr0 OpenWindows_Users
#/usr/etc/install/extract_files sr0 OpenWindows_Demo
#/usr/etc/install/extract_files sr0 OpenWindows_Fonts
#/usr/etc/install/extract_files sr0 OpenWindows_Programmers
```

Note that you do not need to load `OpenWindows_Programmers` unless you plan to develop window-based applications that will run in an OpenWindows environment.

---

\* In addition to the man page for `extract_files`, see Chapter 2 for information on using `extract_files`.



---

## Part 4 — Open Issues

Part 4, which is found in a package on top of the release binder in the Release Minibox, is comprised of two important chapters:

- Late-Breaking News
- Known Problems

Be sure to insert the *Open Issues* package (Part Number 800-6643-10) into the binder and to read the two chapters before proceeding with the installation of SunOS 4.1.2.







---

## SunOS 4.1.2 Open Issues

**This package contains important information for the *SunOS 4.1.2 Release Manual*. Insert the attached material in the Release binder, behind the *Open Issues* tab.**





---

## Part 4 — Open Issues

Part 4, which is found in a package on top of the release binder in the Release Minibox, is comprised of two important chapters:

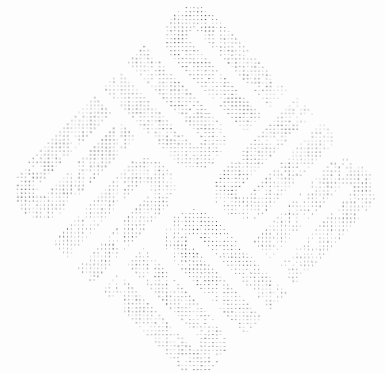
- Late-Breaking News

This chapter includes warnings about patch installations, lists of additional bug fixes and patches, documentation changes, and ergonomics compliance standards for German installations.

- Known Problems

This chapter includes lists and descriptions of bugs in SunOS and OpenWindows Versions 2 and 3.

Be sure to insert this *Open Issues* package into the binder behind the tab for Part 4 and to read the two chapters before proceeding with the installation of SunOS 4.1.2.





---

## Late-Breaking News

This chapter provides the latest news about the FCS version of SunOS 4.1.2, dated December 1991. This chapter includes the following sections:

- Warnings about Patch Installations
- Additional Bugs Fixed and Patches Added
- Additional CTE Escalations Fixed in SunOS 4.1.2
- Release Manual Documentation Changes and Additions
- Sunupgrade Workarounds and Documentation Changes
- Ergonomics Compliance (Germany)
- Ergonomievorschriften (German-language version)

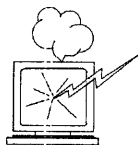
If you are going to use `sunupgrade` to upgrade to SunOS 4.1.2, be sure to read Section 9.5 before starting the upgrade.

See Chapter 10 for listings and descriptions of known problems with the release.

### 9.1. Warnings about Patch Installations

Do not install a patch unless you are sure the patch is compatible with the currently-installed release.

#### Do Not Install Sun IPX Supplement



This patch has been incorporated in the OpenWindows Version 2 included with SunOS 4.1.2; the patch is unnecessary and may cause system problems if it is installed on top of SunOS 4.1.2. Do not install it.

#### Do Not Install Sun 1.3GB Disk Enhancement

A special patch was provided for Sun-4 systems running SunOS 4.1.1 and Sun-4c systems running SunOS 4.1.1 Rev B, to enable use of the 1.3GB disk on these systems. The changes have been incorporated in SunOS 4.1.2; the patch is unnecessary. Do not install it.

## 9.2. Additional Bugs Fixed and Patches Added

The following bugs were fixed and CTE patches added since the *SunOS 4.1.2 Release Manual* went to press.

### Kernel Bugs Fixed

1047696: Panic when trying to install software on Sun-4c with 40MB or more of memory; `hat_pmgreserve` error.

1068363: SPARC 2 system crashes with `maxusers` greater than 83.

### Network Bugs Fixed

1065361: Wrong `gid` for existing file created again.

1045211/1042491: Some network services invoked by `inetd` may fail to run.

### Linker-loader Bugs Fixed

1070559: `ld.so` exhausts swap space with some shared libraries.

### Graphics Bugs Fixed

1071121: GT crashes when picking PHIGS solids with front face culling enabled.

### Library

1052558: System call should do a `vfork`, not a `fork`.

### New Patches Available (Bugs 1074337, 1073234)

Patches are available for the following bugs through your local Sun Answer Center if your system is under warranty or if you have a support contract:

- Bug: 1074337: When formatting disks on Sun-4s, if the defect list happens to be large enough to require two disk blocks, you may get a format failure:

```
formatting... done
assertion failed: file 'ctlr_scsi.c', line 2331
```

See the description of the bug in Section 10.2 for more detail.

- Bug: 1073234: `mmap` system call fails on Sun-4m for data fetch of non-existent device memory address. See the description of the bug in Section 10.2 for more detail.

## 9.3. Additional CTE Escalations Fixed in SunOS 4.1.2

In addition to the Corporate Technical Escalation (CTE) patches listed in Section 6.13 of the *Release Manual*, the following CTEs were also fixed in SunOS 4.1.2. The patch ID is followed by a list of the bugs fixed by the patch, with brief descriptions.

Patch 100244-02 [roll from 100244-01]

1033287 Frequent exhaustion of `chanmap` resource observed

Patch 100356-01 [new]

1067615 Kernel hangs while processing `aiocancel` system call.

- Patch 100364-01 [new]  
1029802 4/280 with xy451 panics: registers accessed while busy.
- Patch 100346-01 [new]  
1044722 Undefined symbol in libxpg.a.
- Patch 100357-01 [new]  
1066663 SIGIO signal for async IO fails to pend across NFS.
- Patch 100173-06 [roll from 100173-04]  
1066287 NFS hang when looking at large file being changed on server.  
1064433 Export of subtree doesn't work  
1045536 NFS exports to non-Sun system can result in file truncation.
- Patch 100305-06  
1063772 Possible to overwrite any file on system using lpr.  
1058003 Reading from /dev/printer, lpd doesn't check how much data it has read  
1016437 lpd does not check file names for legality.  
1040453 lpd can be used by any user to delete any file on the system.

#### 9.4. Release Manual Documentation Changes and Additions

##### OpenWindows Version 2 CTE Patches: Bugs Not Fixed in SunOS 4.1.2

The CTE OpenWindows Version 2 patches listed in Section 6.14 of the *Release Manual* are available through your local Sun Answer Center if your system is under warranty or if you have a support contract; the bugs corresponding to these patches were *not* fixed in SunOS 4.1.2.

##### Corrections to List of CTE Patches Fixed in SunOS 4.1.2

Patch ID 100173 is incorporated at the -06 level — not the -03 level; it is listed at both levels.

Patch ID 100174 is incorporated at the -03 level — not the -01 level.

Patch ID 100199 is incorporated at the -02 level — not the -01 level.

##### Location of New Commands mps and mpstat

The new SPARCsystem 600MP commands, `mps` and `mpstat`, described in Chapter 2 of the *Release Manual*, are only available for MP systems; they are found in `/usr/kvm`.

## 9.5. Sunupgrade Workarounds and Documentation Changes

### Change Time Zone Before Running `check_perm`

Unless you are in the US/Pacific time zone, you should set the time zone to US/Pacific before running `check_perm`; if you don't, SunOS 4.1.1 files that you have not changed since installation will be flagged as changed and included in `/usr/tmp/volatile_file`. If you then use `/usr/tmp/volatile_file` as your volatile file list, all those files will be saved with `.411` tags.

A simple workaround for this problem is to execute one of the following commands, depending on whether you are running the C shell or Bourne shell. Since you run the commands in a sub-shell, your time zone will not be permanently reset.

csh example:

```
# (setenv TZ US/Pacific; check_perm [check_perm arguments])
```

sh example:

```
# (export TZ; TZ=US/Pacific; check_perm [check_perm arguments])
```

### File `/etc/uucp/Systems` Is Replaced

Note that the file `/etc/uucp/Systems` is replaced during sunupgrade. If you wish to preserve your current version of the file, you will have to copy or move it before running sunupgrade, or you could add `/etc/uucp/Systems` to your own volatile file list and use it instead of the default list, `volatile_list`.

Alternatively, if you run `check_perm`, the file will be included in `/usr/tmp/volatile_file`, which can then be used when running sunupgrade. A backup copy of `/etc/uucp/Systems` will then be saved, with a `.411` suffix.

### If You Have XNews Server Patch

If you have installed the XNews Server patch (100176-08), you may see the following message during sunupgrade:

```
/usr/openwin/bin/xnews won't be upgraded because not
a regular openwindows server
```

This patch, which is not included in SunOS 4.1.2, will not be overwritten by sunupgrade. The rest of the OpenWindows package will be upgraded.

**Installation Manual:  
Upgrade; Rebuild Kernel  
Before Reboot**

On page 2-9 (step 15) and 2-13 (step 16) of *Installing SunOS 4.1.2 System Software* the following note should be added to the last sentence of the last paragraph of the step:

“If you customize your kernel configuration file, you will have to rebuild and install the new kernel before rebooting.”

**Installation Manual:  
Upgrade; Multi-user Error  
Message**

It is assumed that you ran `shutdown` before starting `sunupgrade`, as described in the instructions; `sunupgrade` checks to see if `/etc/utmp` is empty before starting the upgrade. If it is not empty, an error message indicating that you are running in multi-user mode will be displayed and `sunupgrade` will exit. You will have to zero out `/etc/utmp` or boot multi-user and run `shutdown` before running `sunupgrade`.

**Installation Manual:  
Upgrade; Remote Upgrade  
Needs `nfsd`**

In step 3 of the instructions for remote upgrades, you are told that you may have to start `rpc.mountd` manually. You may also have to start `nfsd`. See the `mountd(8c)` and `bioid(8)` man pages for details.

## 9.6. Ergonomics Compliance

Note that this section is included in both English- and German-language versions; the German version follows the English version.

### German Ergonomics Standard

To comply with the German ZH1/618 ergonomics standard, you must use the Graphical User Interface available under SunView. The OPEN LOOK Graphical User Interface does not comply with the standard.

### Using SunView with 16- and 17-Inch Monitors

To comply with the ZH1/618 standard, the display character height must be at least 2.6 mm. SunView complies with the standard on 19-inch monitors, but you must change the SunView default font size on 16- and 17-inch monitors. To change the default font size for SunView, complete the following procedure.

1. Start SunView.
2. Start the Defaults Editor program from the SunView menu or from a command tool or shell tool by typing:

```
% defaultsedit &
```

3. In the SunView defaults category, change the Font default to:

```
/usr/lib/fonts/fixedwidthfonts/screen.r.16
```

4. Save your edits and quit the Defaults Editor.
5. Exit and restart SunView.

The new font size will now be used to display text.

### Shelltool Support for International Keyboards

To use the shelltool (in SunView or OpenWindows) with non-ASCII characters generated by non-U.S. keyboards or characters generated with the Compose key of any keyboard, you must edit the shell initialization file to enable an 8-bit data path. To use the shelltool (in SunView or OpenWindows) with non-ASCII characters generated by non-U.S. keyboards or characters generated with the Compose key of any keyboard, you must edit the shell initialization file to enable an 8-bit data path.

To enable the non-ASCII characters in a shelltool:

1. Type `env` in a command window to determine the shell you are running.

If `SHELL=/bin/csh` is displayed, you are running the C shell, so you should edit your `.cshrc` file.

If `SHELL=/bin/sh` is displayed, you are running the Bourne shell, so you should edit your `.profile` file.

2. Edit the shell initialization file.



Edit your `.cshrc` file if you are running the C shell, or edit `.profile` file if you are running the Bourne shell. These files are in your home directory. To change to your home directory, type `cd` and press Return. You can use any text editor to edit these files.

In the `.cshrc` file, add these two lines at the beginning of the file;

```
setenv LC_CTYPE iso_8859_1
stty pass8
```

In the `.profile` file, add these two lines at the beginning of the file:

```
LC_CTYPE=iso_8859_1; export LC_CTYPE
stty pass8
```

Changes to the `.cshrc` and `.profile` files will take effect in subsequent shelltools; or you can "source" these files in existing shelltools by typing `source` and the name of the file.

This example enables the non-ASCII characters in a C shell:

```
% env
  (list of environment variables...)
SHELL=/bin/csh (this is a C shell)
% cd
% vi .cshrc
  (add lines from Step 2)
% source .cshrc
%
```

## 9.7. Ergonomievorschriften

### Deutsche Ergonomie-Norm

Um die deutsche Ergonomie-Norm ZH1/618 zu erfüllen, muß die unter SunView bereitgestellte grafische Benutzeroberfläche verwendet werden. Die grafische Benutzeroberfläche OPEN LOOK entspricht dieser Norm nicht.

### Verwendung von SunView bei 16- und 17-Zoll-Monitoren

Um der Norm ZH1/618 zu entsprechen, muß die Zeichenhöhe mindestens 2,6 mm betragen. SunView erfüllt diese Bedingung bei 19-Zoll-Monitoren, jedoch muß bei 16- und 17-Zoll-Monitoren die SunView-Standardschriftgröße erst geändert werden. Mit der folgenden Vorgehensweise kann die SunView-Standardschriftgröße angepaßt werden:

1. Starten Sie SunView.
2. Wählen Sie aus dem SunView-Menü das Programm "Defaults Editor" oder geben Sie bei einem Command Tool oder Shelltool folgendes ein:

```
% defaultsedit &
```

3. Ändern Sie in der SunView-Standardkategorie die vorgegebene Schriftart ("Font") wie folgt:

```
/usr/lib/fonts/fixedwidthfonts/screen.r.16
```

4. Speichern Sie die Änderung und beenden Sie den "Defaults Editor".
5. Verlassen Sie SunView und starten Sie es neu.

Für die Textanzeige wird jetzt die neue Schriftgröße verwendet.

### Shelltool für internationale Tastaturen

Um das Shelltool (in SunView oder OpenWindows) in Verbindung mit Nicht-ASCII-Zeichen, die mit Tastaturen aus anderen Ländern als den USA erzeugt wurden, oder Zeichen, die mit der Compose-Taste einer beliebigen Tastatur generiert wurden, verwenden zu können, muß die Shell-Initialisierungsdatei so editiert werden, daß ein 8-Bit-Datenpfad freigegeben wird.

Um die Nicht-ASCII-Zeichen in einem Shelltool freizugeben:

1. Stellen Sie durch Eingabe von `env` in einem Befehlsfenster fest, welche Shell aktiv ist.

Wenn `SHELL=/bin/csh` angezeigt wird, handelt es sich um die C-Shell, und Sie sollten Ihre `.cshrc`-Datei editieren.

Wenn `SHELL=/bin/sh` angezeigt wird, ist die Bourne-Shell aktiv, und Sie sollten Ihre `.profile`-Datei editieren.

2. Editieren der Shell-Initialisierungsdatei.

Editieren Sie Ihre `.cshrc`-Datei, wenn die C-Shell aktiv ist bzw. Ihre `.profile`-Datei, wenn die Bourne-Shell aktiv ist. Diese Dateien befinden

Sich in Ihrem Stammverzeichnis. Um zu Ihrem Stammverzeichnis zu gelangen, schreiben Sie `cd` und drücken Sie die RETURN-Taste. Sie können diese Dateien mit jedem beliebigen Texteditor editieren.

Fügen Sie in der `.cshrc`-Datei die folgenden zwei Zeilen am Anfang der Datei ein:

```
setenv LC_CTYPE iso_8859_1
stty pass8
```

Fügen Sie in der `.profile`-Datei die folgenden zwei Zeilen am Anfang der Datei ein:

```
LC_CTYPE=iso_8859_1; export LC_CTYPE
stty pass8
```

Änderungen der `.cshrc`-Datei bzw. `.profile`-Datei treten bei nachfolgend aktivierten Shelltools in Kraft. Sie können die Änderung der Dateien jedoch auch bei bereits aktiven Shelltools wirksam machen, indem Sie `source` und den Namen der Datei eingeben.

Das folgende Beispiel veranschaulicht das Freigeben der Nicht-ASCII-Zeichen in einer C-Shell:

```
% env
  (Liste der Umgebungsvariablen...)
SHELL=/bin/csh (dies ist eine C-Shell)
% cd
% vi .cshrc
  (Zeilen von Schritt 2 einfügen)
% source .cshrc
%
```



---

## Known Problems

This chapter lists known problems with SunOS 4.1.2. Problems and bugs are grouped in the following sections:

- Section 10.1 System Administration
- Section 10.2 Kernel
- Section 10.3 Graphics
- Section 10.4 Network
- Section 10.5 Utilities
- Section 10.6 Compiler
- Section 10.7 Hardware
- Section 10.8 PROM
- Section 10.9 Sundiag
- Section 10.10 Sun 386i
- Section 10.11 Documentation
- Section 10.12 CD-ROM
- Section 10.13 Library
- Section 10.14 Miscellaneous
- Section 10.15 OpenWindows Version 2
- Section 10.16 OpenWindows Version 3 Product Notes

When available, bug ID numbers are given in parentheses after headings. They can be used for referencing problems if you need to contact a Sun Answer Center or sales representative.

### 10.1. System Administration

#### Unable to Install Multiple OS with SunInstall via Multiple CD-ROMs (1042906)

SunOS software can only be installed from one CD-ROM during SunInstall. If you want to install 4.1.2 and 4.1.1 or 4.1 on a machine (to provide Sun-3 support, for instance), allocate disk space and use `add_services` to load the other release from CD-ROM, or extract the other release from tape, if available.

### Full Install Option Under Quick Install Does Not Create /home Partition on 207MB Disks (1044999)

### add\_services(8) Requires 350KB in /usr (1032894)

If you choose the “Full Install” option under Quick Install to load release software on a 207MB disk, /home is created as a symbolic link to /usr/export/home in order to make optimal use of disk space.

If you use add\_services to add a client to a server of the same application architecture, but of a different kernel architecture, the /usr partition must have at least 350KB of available space. This applies to clients and servers running under the same release; it is not a problem under multiple releases. The requirement results from the fact that the root software category is shared between different architectures of the same release. The existing code assumes that root should go into /usr/share, instead of /export/exec. If no space is available in /usr/share, the system sends an error message such as:

```
Not enough space in sd0a
```

### Automounter May Cover /home on System Mounting /home from Local Disk

If your system is using the automounter and you mount the home partition of your local disk on /home, the automounter may cover your /home directory and prevent you from accessing it.

The automounter is enabled on most systems by default and uses automounter maps (often these are NIS maps) to find file systems and determine mount points. The automounter preempts the directories that it uses as mount points so that only the file systems it mounts on them are accessible. In most cases, the user’s home file system has been entered on an automounter map, and the automounter mounts it on the user’s /home directory. A problem can arise if you maintain your home file system on a local disk, your home file system is not included on an automounter map, and you mount it on your /home directory. In this case, the automounter does not know about your home file system, cannot mount it on /home, and prevents you from accessing it.

In order to have access to your home file system, you have to give it a mount point that has not been preempted by the automounter. To do this, you can become superuser, create a new directory, and mount your home partition on it, as in the following example:

```
%su
Password: [enter root password]
#mkdir /usr/username
#mount /dev/sd0h /usr/username
```

You can also set up an entry in your /etc/fstab file so that your local home file system is automatically mounted whenever your system reboots or you use the mount(8) command to invoke your fstab file. (See fstab(5) for information on setting up an entry in /etc/fstab.)

If you have an entry in fstab that mounts your local home file system on /home, the automounter will not only prevent you from accessing your file

system, it will prevent you from unmounting it so that you can remount it elsewhere. In this situation, you need to edit your `fstab` file so that `/home` is no longer a mount point for your home file system. You must then reboot your system; the automounter prevents you from unmounting your local file system from `/home` in any other way.

**If `add_client -i` Fails, Use `rm_client` Before Retrying (1033185)**

If the command `add_client -i` fails, you must use `rm_client` to delete any clients entered on the failed attempt before using `add_client -i` to re-enter the same clients.

**`add_client` Does Not Set Up Multiple Hostnames for Multiple Ethernets (1017238)**

A server with multiple ethernet cards will have separate hostnames for each of them. The `add_client` utility only knows the hostname for the first ethernet. As a result, diskless clients created on secondary ethernet cards will only have the hostname for the first ethernet and will not be able to boot. To correct this, you must manually change the hostname of the first ethernet to the hostname of the client's ethernet in the following files on the server:

```
/etc/bootparams
/export/root/client/etc/fstab
```

In addition, SunInstall only enters the hostnames of the first two ethernet cards on a server in a client's `/export/root/client/etc/hosts` file. You must manually enter the hostnames of any additional ethernet cards.

If you are using NIS, you will also need to update the `bootparams` map on the NIS server.

**Long Delay before Prompt in Remote Install from CD on SPARC 390/4X0 System (1043209)**

If you install SunOS 4.1.X from a remote CD-ROM on a SPARC 390/4X0 system and boot the miniroot, there is a series of prompts up to:

```
root file system type (spec 4.2 nfs):
```

After you respond to this prompt there is a delay of up to 90 seconds before the next prompt while the system polls possible controllers. The delay is due to the increased number of controllers and drives supported under SunOS 4.1.1.

## 10.2. Kernel

### Asynchronous I/O Peaks Can Hang System (1073679)

The kernel keeps a cache of stacks for kernel LWPs in memory, and grows this cache on demand. Since the cache has no upper bounds and never shrinks, it grows to accommodate the peak async I/O usage, and keeps that memory forever. This can starve the kernel, causing all processes to sleep and the system to hang.

### Disks on esp2, esp3, esp4 not seen by miniroot (1060701)

Because disks on esp2, esp3, and esp4 are not seen by miniroot, you can only install from SCSI bus 0 or 1.

### format: Assertion Failure on 1.3GB Elite Drives on Sun-4 (1074337)

When formatting disks on Sun-4s, if the defect list happens to be large enough to require two disk blocks, you may get a format failure:

```
formatting... done
assertion failed: file 'ctrl_scsi.c', line 2331
```

The label is not written on the disk, so subsequent invocations of `format` will indicate the disk as `<drive type unknown>`. This is due to a problem with reads/writes of more than one sector at a time on disks using `group1` commands. A patch is available through your local Sun Answer Center if your system is under warranty or if you have a support contract.

### mmap System Call Fails on Sun-4m for Data Fetch of Nonexistent Device Memory (1073234)

Trap does not recognize the Mbus timeout and signal a SIGBUS when a data fetch of a nonexistent device memory address is requested; "Unexpected trap" error message is displayed. A patch is available through your local Sun Answer Center if your system is under warranty or if you have a support contract.

### Start RFS on Both Primary and Secondary Servers If Both Are in Domain (1028779)

If an RFS domain has both primary and secondary name servers, you must start the primary name server first and then, within one to two minutes, start a secondary server. The password (if there is one) for the primary server must be given before the password for the secondary server.

When the primary server is started it tries to contact a secondary name server, if one is listed in the file `/usr/nserve/rfmaster`. If the secondary server isn't started, or isn't started in time, the primary server "times out".

### Large maxusers Parameter May Cause System to Panic (1038406)

If the `maxusers` parameter in the kernel configuration file is set too high, the resulting kernel will panic during the system boot sequence with one of the following two messages:

```
panic: insufficient virtual space for segu: nproc too big?
Watchdog Reset!
```

If this occurs, reboot the system using the generic kernel and make a new custom kernel using a smaller `maxusers` value. The `maxusers` limit varies,



depending on the system and the way its kernel is configured. See “Maximum **maxusers** Values for Sun-4, Sun-4c, Sun-4m Servers” in Section 8.3 of the *Release Manual* for more detailed information. In general, the following **maxusers** values should be safe:

```
sun4: 286
sun4c: 225
sun4m: 45 to 185
```

### Using **kadb** Via tty Port on SPARCstation 2 Hangs Console tty (1043532)

If you use **kadb** to debug your system and your console device is one of the tty ports, it will hang in respect to UNIX the first time you enter **kadb**. The system remains perfectly usable and can be accessed via the other tty, frame buffers, or over the network. However, any UNIX processes attempting I/O to the console tty will hang.

## 10.3. Graphics

### Using the SunView Version of the OPEN LOOK Deskset on 24-Bit Systems

The SunView version of the OPEN LOOK Deskset displays strange colors and other undesired effects when used on a 24-bit frame buffer. This includes the TC, GXP and GS systems. We recommend you do not use the SunView version of OPEN LOOK Deskset with any of these systems.

### SunPHIGS 1.2 May Not Double Buffer Correctly on a GS Graphics Accelerator (1041918)

If you are using canvas region workstations in SunPHIGS 1.2 on a GS graphics accelerator, one of the workstations may “flash” while another is updated. This occurs when an application has been inactive for 30 seconds or more and then switches its focus from one canvas region workstation to another. The work-around is to set the SunView environment variable shown below:

```
%setenv PW_COPY_ON_DBL_RELEASE
```

This problem has been corrected in SunPHIGS 1.3.

## 10.4. Network

### Second Ethernet Starts Even When Not Configured (1066220)

If you have multiple ethernet interfaces and only configure the primary ethernet (le0) the second ethernet (le1) seems to be started automatically even though there is no `hostname.le1` file. The second Ethernet is not actually started, but it is left with the `IFF_RUNNING` flag set; no IP address is configured onto the interface. This has no effect on any other software in the system. You may see the following error message during bootup:

```
le0: No carrier - transceiver cable problem?
le0: RARP timed out
```

These are harmless messages and can be ignored. To disable these messages, edit `/etc/rc.boot` (save old one) and comment out the following line:

```
ifconfig -ad auto-revarp up
```

Subsequent reboots will not show these errors.

### Some Network Services Invoked by inetd May Fail to Run (1045211, 1042491)

The program `inetd(8c)` provides an internetwork daemon that invokes network services listed in the file `/etc/inetd.conf`. On rare occasions, the service invoked fails to run. The service most likely to fail is `tfptpd`, which is necessary for booting diskless clients. Two other services that may be affected are `in.cmsd` (OpenWindows Calendar Manager) and `in.comsat` (Mail Tool).

If a service invoked by `inetd` fails to run, terminate `inetd` and restart it. To terminate `inetd`:

1. Get the process ID for `inetd`:

```
% ps -uax | grep inetd
```

The process ID is the first number in the process table that results. In the example below, the process ID for `inetd` is 153.

```
% ps -uax | grep inetd
gavg 6041 2.7 5.5 40 192 p2 S 10:54 0:00 grep inetd
root 153 0.0 0.0 56 0 ? IW Oct 12 0:30 inetd
```

Note: Sample display; contents may vary, depending on user input and the system used.

2. Become superuser, terminate the process (153 in the example), and restart `inetd`:

**ypinit on Slave Server  
Generates Error Message  
(1029284)**

```
% su
Password: [root password]
# kill 153
# /usr/etc/inetd
```

Running `ypinit -s ypmaster` on an NIS slave server may generate multiple error messages on the screen, among them:

```
RPC Program Not Registered
```

The transfer of maps takes place correctly, in spite of the error messages that may partially fill the screen.

You can prevent the error messages by logging in to the master server as `root` and entering `ypxfrd` to start the master server's `ypxfrd` daemon before you run `ypinit` on the slave server.

You can also prevent the messages from appearing if you have the `ypxfrd` daemon start automatically when the master server boots. To do this, edit the file `/etc/rc.local` as shown:

```
if [ -f /usr/etc/ypserv -a -d /var/yp/`domainname` ]; then
    ypserv;                echo -n ' ypserv'
    ypxfrd;                echo -n ' ypxfrd'
fi
```

**RFS Mounts on Directories  
with Inodes Greater Than 64k  
Will Fail returning `getwd  
fails: cannot find`**

RFS will truncate inode numbers of 64K or greater. If you plan on advertising RFS resources, be sure that the file system to be mounted is not capable of creating inode numbers of 65536 or greater. The maximum possible inode number for a file system can be found by executing `df -i` and adding the `iused` and `ifree` values for the file system.

**Internet Domain Name  
Service (DNS) Requires  
Hostnames in  
`/etc/hosts.equiv` and  
`.rhosts`**

If you are running NIS in conjunction with DNS and the host is not in your NIS map or DNS domain, you must have entries for the hostname in the files `/etc/hosts.equiv` and `.rhosts`.

## 10.5. Utilities

### **cpio Will Not Copy Rock Ridge Symbolic Links (1069718)**

Because of a problem with `cpio` copying files with symbolic links from a Rock Ridge file system, you may have to use `tar(1)`.

### **Timeout Message for Preinstalled Systems Installed with QuickInstall**

Preinstalled Desktop SPARC systems and Sun-4 and Sun-4c machines which have been installed with the QuickInstall option to `suninstall` will display a timeout message when booting:

```
ifconfig: RARP: timeout
```

The system is attempting to do automatic network configuration, but the network is not set up correctly, or the system may not be attached to a network. In the latter case, another message will be displayed:

```
le0: No carrier - transceiver cable problem?
```

Both of these messages can be ignored on the non-networked installation.

The messages can be helpful when trying to debug Automatic Network Configuration. For the “RARP: timeout” message, the system information could not be obtained from the NIS database in the `hosts`, `ethers`, or `bootparams` maps. The “le0: no carrier” message indicates a problem with the transceiver cable; it is probably not connected or is connected incorrectly.

If Automatic Network Configuration is not your goal, then the “RARP: timeout” message is to be expected.

### **maxcontig Parameter Cannot be Set Above Seven with tune -a (1043735)**

If you use the `-a` option of `tunefs(8)` to set the `maxcontig` parameter for a partition, you must set `maxcontig` at 7 or less. A value greater than seven does not generate an immediate error message, but prevents you from mounting the disk partition you specified. When you try to mount the partition, you get an error similar to the following:

```
# mount /dev/sd0a /mnt
mount: /dev/sd0a on /mnt: I/O error
mount: giving up on: /mnt
```

### uucp Password File Problem at Boot Up

As explained below, an interaction between the password file entry for `uucp` and the `uucp` entry in `/etc/rc` at bootup may cause the following error message:

```
su: uucico: illegal option -- c
usage: uucico [-xNUM] [-r[0|1]] -sSYSTEM -uUSERID -dSPOOL -iINTERFACE
```

The default `passwd` entry for `uucp` is

```
uucp:*:4:8::/var/spool/uucppublic:
```

When adding `uucp` logins to the `passwd` file, it is common to make `uucico` the login shell. In this case, the `passwd` entry for `uucp` becomes:

```
uucp:*:4:8::/var/spool/uucppublic:/usr/lib/uucp/uucico
```

This becomes a problem in interaction with `/etc/rc`. The default `/etc/rc` file has the following line, which is executed upon boot up:

```
su uucp -c /usr/lib/uucp/uusched &
```

Normally, the `-c` option of `su` would be passed along with the rest of the command line to the shell. But in this special case the shell is `uucico`, not `/bin/sh`, which is the default if no other shell is specified. Since `uucico` doesn't have a `-c` option, the option fails and the `su: uucico: illegal option` error message results.

If you get the error message at boot up, you can remove the `uucico` field from the `uucp`'s `passwd` entry. However, although this takes care of the problem with `/etc/rc`, it creates a new problem in its place: the absence of the `uucico` field means that remote systems calling in to do `uucp` transfers will be prevented from carrying them out.

### Error Messages During Heavy IPI Disk Activity (1036367)

During heavy IPI disk activity, error messages similar to the one below may appear. They can be disregarded.

```
Apr  9 13:43:46 muishu vmunix: id003h: block 849694 (849694 abs):
write: Conditional Success. Data Retry Performed.
```

## 10.6. Compiler

### SPARC Compiler Optimization Level -O2 Can Produce Incorrect Code (1031879)

With SPARC compilers earlier than compiler release 0.0 (in 1990) optimization level -O2 (`cc -O` or `cc -O2`) can cause the SPARC assembler to generate incorrect code. In the following C program, the assembler code produced with level -O2 optimization does not test the `while` loop, which leads to an infinite loop.

```
int boothowto = 1;

int
main()
{
    int unit;

    if (boothowto & 1) {
retry:
        unit = -1;
        while (unit == -1) {
            if (unit != -1) {
                printf("unit = %d when it should be -1!0,
                    unit);
                exit(1);
            }
            unit = 0;
            print_unit(&unit);
        }
    } else {
        unit = 0;
        goto retry;
    }
}

print_unit(unitp)
int *unitp;
{
    printf("print_unit: unit = %d0, *unitp);
}
}
```

To prevent the problem, compile with `-O~M` specified to the assembler. If you are running the assembler directly, this is:

```
% as [normal options] -O1
```

If the assembling takes place as part of compiling a high-level language, it is:

```
% cc [normal options] -Ooption as -O1
```

## 10.7. Hardware

### Disk Label of Second 104MB Disk on Desktop SPARCsystem May Be Incorrect (1045344)

If you get the following message the first time you turn your system on, your second 104MB disk has an incorrect disk label.

```
sd1 at esp0 target 1 lun 0
sd1: corrupt label - wrong magic number
sd1: Vendor 'Quantum', product 'P105SS', 205075 512 byte blocks
```

To correct the label, carry out the following steps:

1. Become superuser and use `format(8S)` to reformat your second disk (`sd1`):

```
%su
Password: [enter root password]
#format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
 0. sd0 at esp0 slave 24
    sd0: <Quantum ProDrive 105S cyl 974 alt 2 hd 6 sec 35>
 1. sd1 at esp0 slave 8
    sd1: <Quantum ProDrive 105S cyl 974 alt 2 hd 6 sec 35>
Specify disk (enter its number):
```

2. Enter 1 to select your second disk:

```
Specify disk (enter its number): 1
selecting sd1: <Quantum ProDrive 105S>
[disk formatted, defect list found]
FORMAT MENU:
  disk          - select a disk
  type          - select (define) a disk type
  partition     - select (define) a partition table
  current       - describe the current disk
  format        - format and analyze the disk
  repair        - repair a defective sector
  show          - translate a disk address
  label         - write label to the disk
  analyze       - surface analysis
  defect        - defect list management
  backup        - search for backup labels
  quit

format>
```

3. Enter commands as shown in the following sequence of screen prompts and user responses:

```

format> defect
DEFECT MENU:
. . .
defect> commit
working list was not modified.
defect> quit
FORMAT MENU
. . .
format> format
Ready to format. Formatting cannot be interrupted
and takes 2 minutes (estimated). Continue? y
Beginning format. The current time is Fri Oct 26 13:26:43 1990
Formatting...done
Verifying media...
. . .
Total of 0 defective blocks repaired.
format> partition
PARTITION MENU:
. . .
partition> select
    0. Quantum ProDrive 105S
    1. original sdl
Specify table (enter its number) [1]: 0
partition> label
Ready to label disk, continue? yes
partition>quit
FORMAT MENU:
. . .
format>quit
#

```

## 10.8. PROM

### Long Reset Time on Sun-4m (1067283)

When you install miniroot on a SPARCsystem 600MP system, the time between copying the miniroot to disk and rebooting the miniroot is between 45 and 60 seconds; the screen is blank during this time, which may cause you to think something is wrong.

### Problem Redirecting I/O to Terminal on SPARCstation 2 with CG6 Framebuffer (1042243)

If you have a SPARCstation 2 with a CG6 framebuffer, entering `tt ya io` at the ok PROM Monitor prompt to redirect I/O to a terminal fails and generates the error message:

```
Memory address not aligned
```



Attempting to reboot generates the message:

```
panic: data fault
```

To redirect I/O to a terminal, you need to enter the following commands at the ok PROM Monitor prompt and then power cycle your machine.

```
ok setenv input-device ttya
ok setenv output-device ttya
```

When your system comes up, the output will be redirected to the terminal.

### **Sun-4/330 Cannot Boot Miniroot from Internal Disk with Some PROMs (1044450)**

The boot PROMs on some Sun-4/330s prevent them from booting the miniroot from an internal SCSI disk. The workaround for this is similar to the workaround given in the description of bug 1032123 below. The only difference is that in Step 3 of the workaround, all occurrences of *id000b* must be replaced with the boot address appropriate to your system.

### **Some Older Sun-4s and Sun-3s May Not Be Able to Boot Miniroot under SunOS 4.1.X**

Old boot PROM revisions on some Sun-3 and Sun-4 systems may prevent them from booting the miniroot under SunOS 4.1.X. When this occurs, messages similar to the following are displayed:

```
checksum xxxxxxxx != yyyyyyyyyy
trying to boot anyway

Illegal Instruction ....

Error/doing reset
```

The workaround for this is similar to the workaround given in the description of bug 1032123 below. The only difference is that in Step 3 of the workaround, all occurrences of *id000b* must be replaced with the boot address appropriate to your system.

### **Bug in Some Boot PROMs Prevents `munix` from Booting the Miniroot on IPI Drives (1032123)**

A bug in SPARCserver 390 boot PROMs earlier than 3.0.3 and in SPARCsystem 4X0 boot PROMs earlier than 3.0 prevents `munix` from booting the miniroot on IPI drives.

`munix` (Memory Unix) is a reduced version of UNIX that runs entirely in RAM and contains the `format` program for formatting and partitioning disks. `munix` is loaded off the release media primarily so that `format` can be used on disks that will contain system data.

The *miniroot* is a minimal version of UNIX that is loaded into the swap partition of the system disk in order to run `SunInstall`.

Prior to SunOS 4.1, if you booted `munix` to run `format`, you had to boot off the release media a second time in order to copy the miniroot to disk and then run `SunInstall`. Now, a script automatically copies the miniroot to disk when you quit the `format` program and allows you to boot the miniroot from disk:

```
format> q
Mini-root installation complete.
What would you like to do?
  1 - reboot using the just-installed miniroot
  2 - exit into single user shell
Enter a 1 or 2:
```

If you now enter 1 to boot the miniroot from an IPI disk, the PROM bug prevents booting and generates one of the following messages:

```
checksum xxxxxxx != yyyyyyyyyy
trying to boot anyway

Illegal Instruction ....

Error/doing reset
```

There is a workaround which starts at the point where the screen displays:

```
Mini-root installation complete.
What would you like to do?
  1 - reboot using the just-installed miniroot
  2 - exit into single user shell
Enter a 1 or 2:
```

Carry out the following steps to boot the miniroot and use `SunInstall`.

1. Halt your system: Press `LI/Stop` - `a`
2. Enter the command for booting CD-ROM.

```
>b sd(0,30,1) -asw
```

A series of screen prompts follows.

3. Respond to the screen prompts as follows:

```

root file system type (4.2 nfs ): 4.2
root device (.....): id000b
root on id000b fstype 4.2
Boot: vmunix -asw
root file system type (4.2 nfs ): 4.2
[45 second pause]
root device (.....): id000b
swap file system type (spec 4.2 nfs): spec
swap device (.....): id000b
Swapping on root device, ok? y
#

```

Note: Sample display; contents may vary, depending on user input and the system used.

### Spurious Warning Message from IPI Disk Controllers (1023347)

When booting from IPI disks the following message may be displayed:

```
vmunix: idc0: ctrlr message: 'Warning: bad EEPROM checksum'
```

The warning is harmless and may be ignored.

### Boot PROMs 3.0 and Higher Can Only Boot Off IPI Disk Units 0 and 1 (1037179)

A boot PROM bug in PROMs 3.0 and higher limits booting to IPI disk units 0 and 1. In combination with the following bug, this restricts systems to installing SunOS on either id000 or id001.

### Installing miniroot on id010 or Higher Causes Invalid Boot String (1069384)

After installing the miniroot on any disk ID higher than id007 (id010, for example), an invalid boot string is generated; if you attempt to reboot using the just-installed miniroot, the boot will be attempted from id(0,28,1) instead of the correct id(1,0,1).

### CG6 Frame Buffer Generates Errors with Some PROMs (1030399)

On SPARCserver 390s with PROMs earlier than 3.0.2 and on SPARCsystem 4X0s with PROMs earlier than 3.0, the CG6 frame buffer may generate screen errors or garbage screen when dmesg runs on the console. Problems include keyboard buffering (characters not being printed on the screen or recognized until a carriage return is entered), and mouse event states not being reset (for example, if an event state is not reset, once you scroll up on a scrollbar, you cannot scroll down, or do anything else with the mouse).

## 10.9. Sundiag

### Sundiag Reports Loopback Errors While Testing mcp (1068117, 1068776)

If you install the SunLink HSI/S 1.0 software after installing SunOS 4.1.2 on your system, the sundiag provided with the HSI/S distribution will overwrite the SunOS 4.1.2 sundiag, which includes bug fixes for bugs 1068117 and 1068776.

To preserve the bug fixes, save the SunOS 4.1.2  
/usr/diag/sundiag/sunlink test file prior to installing the HSI/S

software (for example, move the `sunlink` file to `sunlink.orig`), and restore it after installing the HSI/S software (move `sunlink.orig` back to `sunlink`).

### Sundiag 2.3 Fails Startup Probe Because of False `devinfo` (1071674)

Sundiag uses the `/usr/etc/devinfo` program in its probing routine when it first starts up. `devinfo` finds eight drives for any IPI controller, regardless of whether there are eight drives on the controller. The result is that `sundiag` displays probing failures such as the following if fewer than eight drives are on an IPI controller:

```
bench_3401 ./probe ERROR: /dev/rip0c open error: No such device
```

### There is No `fddi` Test in Sundiag 2.3 Alpha 5 (1070359, 1070602)

Sundiag 2.3 does not include an option to test Sun-4m VME `fddi` devices. A `.usertest` file containing the following line can be used in `/usr/diag/sundiag` to have `nettest` verify the device:

```
nettest, Fddi, net_fddiO D=10
```

## 10.10. Sun 386i

### 386i diskless client is not added successfully (1071266)

When installing 386i diskless client on SunOS 4.1.2, the 386i SunOS 4.0.1 server kit script, `sun386client` tries to modify `/var/yp/Makefile`, checking only for the existence of the `auto.master` map. Only two of the three necessary NIS maps are added; `auto.vol`, which is needed by 386i clients, is not added.

### Sun386i Server Kit Causes All Clients to Automount `/home` (1043173)

On a heterogeneous server, the Sun386i Server Kit creates automount maps in `/etc/fstab` for all clients, including non-386i clients. This means all clients run `automount` to mount `home/servername`. The problem is that all non-386i clients already had `/home` entries in `fstab`. Thus they have separate `automount /home` entries and non-automounted `/home` entries.

The workaround is for the server and all non-386i clients to edit `/etc/rc.local` and change the line

```
automount && echo -n ' automount'
```

to

```
automount -m && echo -n ' automount'
```

## 10.11. Documentation

### *Corrections to System and Network Administration*

#### **Misleading Instructions on Using a UNIX file for Swap Space (1039946)**

On page 247, Step 6 of the instructions for using a file to create extra swap space calls for entering the following command:

```
# mkfile -n 16m /export/swap/raks
```

The command is correct for setting up a file to be used by a client system over the network. The command fails if it is used to create a local file on the same machine that will be using it. Characteristic error messages are:

```
assertion failed panic: bn! UFS_hole
```

and

```
watchdog reset
```

To set up a local file for a standalone system or server to use for added swap, you must leave out the `-n` option:

```
# mkfile 16m /export/swap/raks
```

#### **Error Message and Explanation Incorrect**

In Table 21-6 (incorrectly labeled “Table 21-1”), page 687, an error message and its description are incorrectly given. The error message is:

```
No uucp server      A tcp/IP call is attempted,
                    but there is no server for uucp.
```

The description should read:

```
No uucp service number  No entry for uucp/tcp can be found in
                        /etc/services, but there is a hyphen
                        in a Systems file entry port field
```

**Unresolved Cross References**

- On pages 257 and 261, the reference “See the section on *@TitleOf(repair.sector)*” should read:  
See the section on *Repairing a Defective Sector*
- On page 269, the reference “See the section on *@TitleOf(defect.list)*” should read:  
See the section on *Creating a Defect List*
- On page 279, the reference “See the section on *@TitleOf(using.format)*” should read:  
See the section on *Using format for Basic Maintenance*

**Corrections to the SunOS Reference Manual****Remove skyversion(8) man Page**

The printed version of the *skyversion(8)* man page is specific to Sun-2 systems, which are no longer supported. It should be ignored. The online man page has been removed.

**Correction to Network Programming Guide****Incorrect Warning at Beginning of Chapter 10**

The warning at the beginning of Chapter 10 of the *Network Programming Guide* is in error. Socket-based IPC has **not** been superseded by the UNIX system Transport-Level Interface; any statements implying that it has been are incorrect. The first paragraph in Chapter 10 should be replaced with the following text:

This chapter provides detailed information, with various examples, on the UNIX system Transport-Level Interface. This interface is intended to provide the user with a more structured transport-level interface than that which is provided by the socket interface, and as such deliver transport-independence in a transparent manner.

TLI does not supersede sockets but is intended to offer the developer another mechanism by which to assure that applications under development can and will exist autonomously, regardless of the network or transport protocols.

**10.12. CD-ROM****Actions Causing SunCD to Hang System (1033100, 1032990)**

To prevent your system from hanging when using a CD-ROM, **do not**:

- Attempt to access a defective or “bad” CD-ROM (one that contains recoverable recording errors);
- Access a non-HSFS disc on the CD-ROM drive.

- Eject the CD-ROM while the CD-ROM file system is still mounted.
- Turn the CD drive off while the CD-ROM file system is still mounted.

### SunInstall Can Only Find CD-ROM at `sr0` (1044687)

SunInstall has an internal list of devices it can use for reading installation media. The only CD-ROM device included is `sr0`.

### CD-ROM `eject(1)` May Not Work on a Read Failure (1033102)

If for some reason the CD-ROM driver can't read a CD (as in the case of a defective disc) the `eject` command may also fail, returning the following error message:

```
# eject cd
eject: Open fail on cd -> /dev/rsr0: I/O error
#
```

You can eject the disc manually.

### CD-ROM Error Messages on Console (1032918)

Messages similar to the one below may appear on the console at various times, most often when you mount the CD-ROM or run demos from it. They can be disregarded.

```
sr0a: read recoverable, block 198000
sense key(0x1): soft error, error code(0x18): soft data error
```

## 10.13. Library

### `textdomain(3)` Requires Two Arguments (1045495)

The *SunOS Reference Manual* documents the `textdomain(3)` library function as only requiring a single argument. However, the code for `textdomain()` expects a second argument. Without the argument, a program calling `textdomain()` dies with a segmentation violation. The second argument has no operational effect. It was called for in the original design of the function, but when the design changed and it was no longer necessary, the code that tested for its presence was never dropped.

The workaround for the problem is to use a dummy second argument in programs that call `textdomain()`:

```
textdomain("domain_name", "");
```

## 10.14. Miscellaneous

### **mt status and mt asf Commands Under SunOS 4.0.3 Not Compatible with SunOS 4.1.X**

The use of the `MTGETSTATUS` ioctl data structure changed in SunOS 4.1. As a result, SunOS 4.0.3 (and earlier) binaries for `mt status` and `mt asf` cannot be used with SunOS 4.1.X.

### **You Cannot Install SunShield on SunOS 4.1.2 using cdmanager (1069654)**

See the workaround in Chapter 7 of the *Release Manual*.

## 10.15. OpenWindows Version 2

### **You Must Remove or Edit .xinitrc File in /home**

If you already have a `.xinitrc` file in your home directory, make sure you either remove it or edit it according to the instructions in Chapter 2 of the *OpenWindows Installation and Start-Up Guide* before you start OpenWindows for the first time.

### **You Must Move app-defaults Directory**

The `app-defaults` directory in `/usr/openwin/lib/app-defaults` should be moved to `/usr/openwin/lib/X11/app-defaults` before installing OpenWindows on your system. To make the change, become superuser and enter the following commands:

```
% su
Password: [enter root password]
# cd /usr/openwin/lib
# mkdir X11
# mv app-defaults X11
```

### **Running NeWS Applications on a Non-Networked Standalone System**

In order to run NeWS applications on a standalone workstation that is not connected to a network, you must start OpenWindows with the `-noauth` option so that security is not enabled. This is necessary to bring up the tutorial as part of the default desktop on a standalone workstation. If the `-noauth` option is not used, the following error message will be displayed:

```
XNeWS Network security violation
Rejected connection from: hostname
```



**OpenWindows Invoked from Command Line does Not Reset Foreground Color Correctly (1041554)**

If you invoke OpenWindows from a command line, it will not reset the foreground color when it exits. For example, if you are in color SunView and invoke OpenWindows from the shell (%) prompt, when you exit OpenWindows, you will not be returned to the initial SunView color foreground. To reset the foreground color, enter the command:

```
%clear_colormap
```

**Exiting OpenWindows May Cause Display of Error Messages (1044695)**

In some cases, when the OpenWindows window manager (olwm(1)) exits a program, an error message is generated. This may happen when `fasthalt(8)` or `fastboot(8)` are used. Examples of the error messages generated are:

```
XIO: fatal IO error 32 (Broken pipe) on X server "0.0"
```

and

```
WIN ioctl number c0286722: Inappropriate ioctl for device
```

You can safely ignore the messages if they are displayed when you intentionally exit OpenWindows.

The messages are generated when `olwm` exits an X11 program that is not fully compliant with the ICCCM mechanisms for interacting with a window manager.

**Control-C When OpenWindows Version 2 is Starting Freezes Window System (1039856)**

If you press your interrupt character (usually `Control-C`) to halt OpenWindows when it is starting up, the OpenWindows windows still come up on screen and the mouse cursor can be moved across them, but all mouse and keyboard buttons are frozen, so that no window activity is possible. This is because the *Window Manager* was interrupted, but other components of the window server were not. Since the window system cannot respond to your keyboard in this situation, you need to `rlogin` to your own system from another machine on the network and stop the window server. To stop the window server:

1. Get the server's process ID:

```
# ps uax | grep xnews
```

2. Halt the process:

```
# kill [process-ID]
```

If you cannot `rlogin` to your system, call your local Sun Answer Center for assistance. This problem is corrected in OpenWindows Version 3.

**Error Message for Incorrectly Set Keyboard DIP Switches**

In the unlikely event that your keyboard DIP switches are set incorrectly, you will see the following message when you start up OpenWindows:

```
ClassKeyboard couldn't initialize the keyboard.
Process: 0x1ebc9c (Unnamed process)  Error: undefined
Stack:  (NeWS/interest.ps) marker /BasicKeyDicts marker
Executing:  ascii000
At:  {*ascii000  ascii0S0  ascii00L  ascii0S0  asciiC00  asciiC00  asciiC00
asciiC00}
In:  Reading file ('NeWS/interest.ps',R)
Sic transit gloria PostScript
giving up.
xinit:  Connection refused (errno 61): unable to connect to server
```

If you are using a Type-4 keyboard and SunOS 4.1.X, all DIP switches on a U.S. keyboard must be set to "0" or "off". See Chapter 3 of the OpenWindows *Installation and Start-Up Guide* for the settings for international keyboards.

**Problem with F1 (Help) Key on Type-4 Keyboards**

On Type-4 keyboards, the F1 (Help) key does not work properly if either Caps Lock or Num Lock is on.

**Error Message for Incorrect Permissions on /tmp Directory**

If OpenWindows displays the following error message:

```
XNeWS: there is already a NeWS server running on :0
giving up.
/usr/openwin/bin/xinit:  Permission denied (errno 13): unable to
connect to X server
```

Check the permissions on the /tmp directory. They should be: drwxrwsrwt.

**Do Not Resize Text Sub-Window Smaller Than the Top of the Bottommost Split**

Resizing split text subwindow smaller than the top of the bottommost split causes the program that owns the text subwindow to crash. Any changes in any of the program's windows that have not been saved previously are lost. This bug affects File Manager, Mail Tool, and Text Editor, as well as all other application programs that use text sub-windows. If you are using the split feature of text sub-windows, you should not resize the text sub-window smaller than the top of the bottommost split.

### Mail Tool: Running out of Disk Space in /tmp

Mail Tool does not handle running out of disk space in /tmp gracefully. If you have too large a mail spool file, or too little space in /tmp, Mail Tool may give an error message, terminate, and possibly leave a lock file, which you will need to remove, in /usr/spool/mail. The name of the file is *username.lock*. To remove the lock file, enter:

```
% rm /usr/spool/mail/username.lock
```

If Mail Tool runs out of space in /tmp while running, it is possible that deleting messages from the *In Tray* may result in the deleted messages being duplicated and sequenced out of order.

It is recommended that you hold down the size of your spool file by limiting the number of messages in your In Tray. In addition, you can conserve disk space by removing unnecessary files from the file system containing /tmp.

### Do Not Disable Scrolling in One of the Windows of a Split-Screen Command Tool

Disabling scrolling in one of the windows of a split-screen Command Tool may cause it to crash.

### File Manager Can Crash If Too Many Editing Sessions Are Started

The File Manager can crash if too many editing sessions are started in a short period of time. When this happens, all unsaved changes are lost. To prevent such crashes, Sun strongly recommends that you do the following:

1. Select `Tool Properties` from the Properties menu button on the File Manager control panel.
2. Click on the `Other` option next to `Default Document Editor`.
3. Fill in the blank with:
 

```
textedit "$FILE"
```
4. Press the `Apply` button at the bottom of the window.

### File Manager's Wastebasket Icon May Be Invisible Or Appears As a Short Text String with No Picture

If the File Manager's Wastebasket icon appears as a short text string with no picture, double-click on the string. This will open up the Wastebasket window. When you close the Wastebasket window, the icon will assume its normal appearance.

If the Wastebasket icon is initially invisible, exit and restart the File Manager. In most cases, this will solve the problem. If the icon is still not visible, either clicking around the edge of the screen where your other icons are located or select `Screen Refresh` from the Utilities menu. `Screen Refresh` produces a short text string, as described above. Follow the previous instructions to get the normal Wastebasket icon.

### Application Programs Can Move Windows in Front of the Lock Screen on Monochrome Monitors

On monochrome monitors, application programs can move new or updated windows in front of the Lock Screen. As a result, the contents of a window may be displayed even when the screen is locked. This can happen, for example, if you lock the screen as soon as you start saving a large Mail Tool infile. When the Mail Tool save is completed, the message view window will still be displayed, in spite of the screen lock. Color monitors do not exhibit this problem.

### Adjusting for OpenWindows Monitors with Overscan Capabilities

OpenWindows Version 2 supports monitors with overscan capabilities. For monitors running in overscan mode, the server can be started up with an adjusted viewing size in order to compensate.

In future systems, the need to run in overscan mode may automatically be determined at startup time. However, there will be occasions when X11/NeWS is run on a second non-overscan monitor, or on monitors that are not properly aligned, or are of an unusual size. In these cases, overscan mode may have to be shut off or the dimensions of the visible area changed. To allow for the eventualities, the following command-line arguments will be supported by X11/NeWS once the overscan code is installed.

- `-dev [fbname]`

This option will tell the server what device to display on. This replaces the usage of the FRAMEBUFFER environment variable in OpenWindows 1.0.

- `-overscan [percent]`

This option instructs the server to shrink the visible area by the indicated percentage and perform the necessary offsetting. If a percentage of `-1` is given, the shrinkage will be set to the default. If a percentage of `0` is given, overscan mode will be disabled and the server will use the full size of the screen. This will allow users to override any defaults if a particular monitor doesn't behave to their liking.

In addition, if the `-overscan` option passes a non-zero value, it will enable the special overscan functionality (the flooding of the unused border regions with the root X color/pattern and the constraining of the cursor to the root canvas).

- `-rect [L T W H]`

This option instructs the server to use a viewing area described by the rectangle passed in. Note that this mode will not perform the special cursor/border functions which are part of overscan mode unless overscan mode is also enabled. This option simply sets an explicit viewing area.

Examples:

```
# Run xnews in default overscan mode, overriding any defaults.
xnews -overscan -1

# Run xnews in overscan mode but with a set shrinkage of 10%,
# overriding any defaults.
xnews -overscan 10

# Run xnews with overscan mode off, overriding any defaults.
xnews -overscan 0

# Run xnews with an explicit view area and overscan mode shut off,
# overriding any defaults.
xnews -overscan 0 -rect 200 200 600 600
```

**GX Hardware Cursor May Be  
Left on Screen after Exiting  
OpenWindows**

There are some cases in which the GX hardware cursor is left on the screen after exiting OpenWindows. The workaround is to compile and run the following program:

```

/*
 * gxcursor -- disable GX cursor
 * to compile: cc -O -o gxcursor gxcursor.c -lpixrect
 */
#include <stdio.h>
#include <sys/types.h>
#include <sys/ioctl.h>
#include <sun/fbio.h>
#include <pixrect/pixrect.h>
#include <pixrect/memvar.h>

struct cg6pr {
    struct mprp_data mprp;        /* memory pixrect simulator */
    int fd;                      /* device file descriptor */
    struct pr_size cg6_size;     /* screen size */
    caddr_t cg6_fbc;            /* FBC base */
    caddr_t cg6_tec;           /* TEC base */
};

#define THCOFFSET                (5 * 4096)
#define THC_CURSOR                0x8FC

main(argc, argv)
    int argc;
    char *argv[];
{
    char *dev = argc > 1 ? argv[1] : "/dev/fb";
    int fd;
    struct fbgattr fbattr;
    Pixrect *pr;

    if ((fd = open(dev, 2, 0)) < 0)
        perror(dev);
    if (ioctl(fd, FBIOGATTR, &fbattr) < 0 ||
        fbattr.fbtype.fb_type != FBTYPE_SUNFAST_COLOR)
        fprintf(stderr, "device %s is not a GX frame buffer0, dev);

    (void) close(fd);

    if ((pr = pr_open(dev)) == 0)
        fprintf(stderr, "pixrect open failed for %s0, dev);

    * (int *) (((struct cg6pr *) pr->pr_data)->cg6_fbc +
        THCOFFSET + THC_CURSOR) = 0xffe0ffe0;

    exit(0);
}

```

### GX with OpenWindows on Multiple Screens: Some Operations May Leave Cursor Invisible

When you run on a GX and have OpenWindows displayed on multiple screens, operations that warp the cursor to a new position (such as scrollbars and pop-ups) may leave the cursor invisible until you move the mouse. This behavior does not start until after the mouse cursor visits the non-GX screen. Once the disappearing cursor starts, it is non-deterministic (due to a race

condition), so it shows up about 50% of the time. Three workarounds follow:

- Set the Scrollbar Pointer Jumping and Pop-up Pointer Jumping properties in the “Mouse Settings” Workspace property sheet to off, then restart OpenWindows. This fixes the most common XView symptoms.
- Adjust the OpenWindows.PopupJumpCursor and Scrollbar.JumpCursor properties in the `~/.Xdefaults` file to be False. For example:

```
OpenWindows.PopupJumpCursor:  False
Scrollbar.JumpCursor:         False
```

This fixes the most common XView symptoms.

- Permanently disable the new hardware cursor tracking feature in the kernel by adjusting a kernel variable and rebooting:

```
% su
Password: [enter root password]
# adb -w /vmunix /dev/kmem
not core file = /dev/kmem
win_do_hw_cursor?W 0
$quit
# /etc/fastboot
```

Under this workaround, cursor tracking may lag behind mouse motion in some circumstances.

### BadAlloc Xerror Results from Large Number of Panel Items

In XView, the number of panel items is unlimited. However, after a certain point, the server will run out of virtual memory with the message:

```
BadAlloc Xerror
```

### CANVAS\_PAINTWINDOW\_ATTRS in the Create Call of a Canvas May Not Work Correctly

Due to a bug in the XView canvas package, trying to set some canvas paint window attributes using `CANVAS_PAINTWINDOW_ATTRS` in the create call of a canvas will not work correctly. The attributes that fail are `WIN_BIT_GRAVITY` and anything that adjusts the window’s event mask (e.g., `WIN_CONSUME_EVENT(S)` and `WIN_IGNORE_EVENT(S)`). The work around is to set the `CANVAS_PAINTWINDOW_ATTRS` after the canvas has been created.

For example, given:

```

canvas = (Canvas)xv_create(frame, CANVAS,
    ...,
    CANVAS_PAINTWINDOW_ATTRS,
        WIN_BIT_GRAVITY, SouthWestGravity,
        WIN_CONSUME_EVENT, LOC_DRAG,
        0,
    ...,
    0);

```

you would produce the following:

```

canvas = (Canvas)xv_create(frame, CANVAS,
    ...,
    ...,
    0);

xv_set(canvas,
    CANVAS_PAINTWINDOW_ATTRS,
        WIN_BIT_GRAVITY, SouthWestGravity,
        WIN_CONSUME_EVENT, LOC_DRAG,
        0,
    0);

```

### SunView Compatibility under OpenWindows Does Not Work with GS Systems (1048981)

If you have a SPARCstation with GS graphics, the SunView compatibility feature of OpenWindows does not work correctly. When running a SunView application under OpenWindows, the screen goes blank and the XNeWs server eventually dumps core and terminates.

### Using DNI

- If you run the X11/NeWS server over DNI, you will crash the server if you view an access list using `xhost` after adding a host to the list with `newshost`.
- You must unset the environment variable `DNI_X_ENABLE` before starting the server if DNI is not in a mode that accepts connections.
- If you turn off the network node state while the server is running, you will crash the server.
- You can cut text from an OpenWindows application into a DEC application, but not from a DEC application to OpenWindows.



## 10.16. OpenWindows Version 3 Product Notes

The following sections, from *OpenWindows Version 3 Product Notes* (Part Number 800-6387-10), contain important information about this release of the OpenWindows software. See the following documents for additional release and installation information:

- *OpenWindows Version 3 Release Manual*
- *OpenWindows Version 3 Installation and Start-Up Guide*

## Installation Issues

### Do Not Install SunOS 4.1.1 Patches

There are SunOS 4.1.1 patches available in `$/OPENWINHOME/lib/OSpatches`. Do *not* install these patches on your SunOS 4.1.2 system.

### Do Not Remove OpenWindows Version 2 During Installation of Version 3

During installation of OpenWindows Version 3 you are given the option of removing OpenWindows Version 2:

```
Do you want to remove the current contents of
the /usr/openwin directory? [y|n]
```

Do *not* respond “y” if you are doing the installation while running OpenWindows; the system will be in an unstable state if you do. If you exit OpenWindows before starting the installation and you decide to remove OpenWindows Version 2, be sure you have room for OpenWindows Version 3, as described below.

### Check Space Available Before Installing OpenWindows Version 3

OpenWindows Version 3 uses more space than OpenWindows Version 2 — over 85000 kilobytes for all packages vs. about 40000 kilobytes for all packages with OpenWindows Version 2. Be sure you have sufficient space available before beginning your SunOS 4.1.2 installation. See Table 10-2 below for information about how much space is needed for various OpenWindows configurations.

### Installation Cluster Sizes

The subset sizes for installation packages provided in the *OpenWindows Installation and Start-Up Guide* are incorrect. Table 10-1 below lists the sizes of the packages. Note that some of the packages are subsets of the others (see the *OpenWindows Installation and Start-Up Guide* for the contents of each package).

Table 10-1 *Installation Packages*

<b>Package and Content</b>	<b>Disk Space Required</b>
Required Package	27700 Kbytes
Online Man Pages	1300 Kbytes
Library Interface Man Pages	2100 Kbytes
Online Handbooks	2000 Kbytes
Optional MIT Release	2300 Kbytes
Demo Programs	2800 Kbytes
Demo Images	5100 Kbytes
Include Files	3000 Kbytes
Lint Libraries	600 Kbytes
Static Libraries	10700 Kbytes
Sample Source	7900 Kbytes
Optional Fonts	10100 Kbytes

Table 10-2 below lists the space used by the installation clusters. Note that the amount of space requested by the script (“Script Requests”) differs from what is actually used (“Space Used After Install”).

Table 10-2 *Installation Clusters*

<b>Cluster and Content</b>	<b>Script Requests</b>	<b>Space Used After Install</b>
Minimal Configuration	31000 Kbytes	31588 Kbytes
End User	41100 Kbytes	41233 Kbytes
Advanced User	43400 Kbytes	43412 Kbytes
Advanced User with Demos	51300 Kbytes	51168 Kbytes
Programmer*	67700 Kbytes	66556 Kbytes
Full OpenWindows	85700 Kbytes	74312 Kbytes

\* Does not include Demos.

If your disk has the space necessary for the cluster or package (see the column on the right in Table 10-2), but not the amount that the installation script requests (middle column), the script will not allow you to install. To solve this problem:

1. Use the `df` command to find a disk partition — `/var/tmp`, for example — that has extra space.
2. Move files from the disk partition that you are installing onto to the disk partition with extra space.
3. Move enough files so the disk partition that you are installing onto has the necessary space (listed in the middle column of Table 10-2 above).
4. Run the installation script.
5. Move the files back to the disk partition that you installed on.

### Missing F3 Fonts with the Minimal Configuration Cluster

If you have installed the OpenWindows software using the minimal configuration cluster, note that this configuration does not include the full set of F3 fonts. Many OpenWindows applications, including NeWSprint and FrameMaker for OpenWindows, cannot run without these fonts in place and will fail under the minimal configuration.

If you intend to run NeWSprint or other unbundled or third-party applications in the OpenWindows environment that require these fonts, you must install a cluster that includes the full F3 font set (i.e., the End User Cluster or one of the ones listed below it in Table 10-2). See the *OpenWindows Version 3 Installation and Start-Up Guide* for details.

### Installing the OpenWindows Software with cdmanager

It is possible to install the OpenWindows Version 3 release with `cdmanager` or `cdm` under an OpenWindows Version 2 session. As noted above, though, do *not* perform the installation in an OpenWindows Version 2 window if you are going to remove OpenWindows Version 2 during the installation.

If you perform the installation of OpenWindows Version 3 in an OpenWindows Version 2 session, you can then exit the OpenWindows Version 2 session, set `OPENWINHOME` to point to the newly installed OpenWindows Version 3 software, run `$OPENWINHOME/bin/install_openwin`, and start up the OpenWindows Version 3 environment.

As stated in the *CDManager and CDM User Instructions*, follow these steps to mount your CD-ROM containing the OpenWindows Version 3 software when you use `cdm` or `cdmanager`:

```
example% su
example# cd /
example# mkdir /cdrom (only if directory does not exist)
example# mount -r /dev/sr0 /cdrom
example# cd /cdrom
example# cdm (cdmanager if you are running openwin)
```

### Running cdmanager Remotely

When you attempt to run the `cdmanager` remotely, the following error message may appear:

```
ld.so: libxview.so.3 not found
```

To avoid this problem, make sure that you have set your environment correctly. See “Setting the Environment Correctly” below for more information. Also note that any time you run a program remotely, you must `xhost` the machine with the display that you are displaying the program on.

### ***OpenWindows Version 3 Installation and Start-Up Guide Corrections***

Note the following corrections to the *OpenWindows Version 3 Installation and Start-Up Guide*:

- page 1, Section 1.1:  
Replace the first bullet item with the following:  
“The system architecture must be Sun-4, Sun-4c, or Sun-4m.”
- page 23:  
Add the following to the last paragraph:  
“The `-nosunview` option is not supported on the GXplus, TC, GS, or GT graphics devices.”
- Chapter 6:  
Add the following installation troubleshooting note:  
“If you are running OpenWindows with a graphics accelerator, and performance seems poor, the OpenWindows software may have been improperly installed. Verify that the OpenWindows software was installed by `root`, that proper `suid` modes were preserved, and that `$OPENWINHOME/bin/install_openwin` was run by `root`. If you have access to XGL, run the XGL `install_check` program to verify that DGA is operational. Contact your system administrator for further assistance.”
- Appendix A, section A.3, pages 40-41:  
Replace this section with the following: “DGA is supported on all SBus device configurations in table A-1 on page 39. DGA is not supported on any VME or P4 device configurations in table A-1 on page 39.”

## **Start-up**

### **Setting the Environment Correctly**

The `openwin` script sets your executable path to have the location of the OpenWindows version of the DeskSet tools before the location of the SunView tools of the same name. If you are using the C shell, make sure you run the OpenWindows tools first when you start the server. Make sure that your `.cshrc` file does not reset your path to have `/bin` and `/usr/bin` before `$OPENWINHOME/bin` and `$OPENWINHOME/bin/xview`. In addition, the `openwin` script sets your library path. If you do not have your library path set correctly, the following error message may appear when you invoke a deskset tool from the command line:

```
ld.so: libxview.so.3 not found
```

You may want to add the following lines to your `.cshrc` file so that your environment is set up correctly for remote logins:

```
setenv OPENWINHOME (installed location of OpenWindows)
setenv LD_LIBRARY_PATH $OPENWINHOME/lib:/usr/lib
set path=($OPENWINHOME/bin $path)
```

## Frame Buffer Issues

Some frame buffers require special use under the OpenWindows Version 3 environment:

- GT

For OpenWindows applications using the TrueColor visual (for example, that use the same color value that is used on a GS framebuffer), the color will look brighter (or more washed out) on a GT. This is because the GT hardware provides *gamma correction* to account for the sensitivity of our eyes to low intensities. This give GT images that use shading and anti-aliasing a higher quality. The gamma correction feature is described in detail in the *SunPHIGS and XGL Programmer's Guides*. It can be turned off as follows:

```
example# gtconfig -G 1
```

Note that graphics (SunPHIGS and XG-based) programs will have lower quality images.

- TC (CG8)

If CG8 is the default frame buffer, you must specify the device when starting up the OpenWindows software with the `openwin` command:

```
example# openwin -dev /dev/cgeight0
```

## Screen Blank (Screensaver)

The OpenWindows Version 3 release has an automatic *screensaver* feature that makes the screen go blank after a period of no use. The *OpenWindows Version 3 Release Manual* incorrectly states that this feature is enabled by default. To enable this feature, you must type the following at the command line:

```
example% xset s on
```

When you exit the OpenWindows environment, the screensaver feature is turned off. For more information, see the `xset(1)` man page.

## realxfishdb Demo

The `realxfishdb` demo in `$OPENWINHOME/demo` is broken in the OpenWindows Version 3 release. Although the `root` window will be painted, no fishes will appear. You can obtain a working version from the February 1991 Catalyst CD.

## Compatibility with the OpenWindows Version 2 Release

### NeWSPrint

The NeWSprint 1.0 software does not work with the OpenWindows Version 3 release. Support for the OpenWindows Version 3 release will be in the next release of the NeWSprint software. To continue using the NeWSprint software, you must re-install it with `small_openwin`, a stripped-down version of the OpenWindows Version 2 software. For more information, see Chapter 12 of the *OpenWindows Version 3 Release Manual*.

### XView

Because error checking has improved with the OpenWindows Version 3 release, the XView libraries will catch some common programming errors that were not caught with the OpenWindows Version 2 XView libraries. For example, with the OpenWindows Version 2 release, not terminating attribute lists with a NULL was acceptable. In the OpenWindows Version 3 release, this error can cause an application to crash with a SIGSEV error.

The following is acceptable OpenWindows Version 2 code, but is not acceptable OpenWindows Version 3 code:

```
xv_set/sliderItem, PANEL_MAX_VALUE, 100);
xv_set/sliderItem, PANEL_VALUE, 100);
```

This is correct code, which will run under OpenWindows Version 3:

```
xv_set/sliderItem, PANEL_MAX_VALUE, 100, NULL);
xv_set/sliderItem, PANEL_VALUE, 100, NULL);
```

### Devguide 1.1

Using Devguide 1.1, the following pinnable menu code is not correct, even though it worked under the OpenWindows Version 2 release:

```
caddr_t *ip;
MENU_GEN_PIN_WINDOW, (Xv_opaque)ip[0], "",
```

The code should actually be:

```
MENU_GEN_PIN_WINDOW, owner, "",
```

In this example, the menu's owner was not set correctly to the tool base frame. Under the OpenWindows Version 2 release, even though the code is incorrect, it will work: the pinned menu unmaps when the tool is iconified. In the OpenWindows Version 3 release, XView is more rigorous about checking window ownership: when the menu is pinned and the tool is iconified, the menu stays up.

## SunView Compatibility

If you see the following message on your screen after you type `openwin`:

```
ld.so: libsunwindow.so.0: not found
giving up.
/usr/openwin/bin/xinit: No such file or directory (errno 2):
unable to connect to X server
```

do one of the following:

- install the SunOS Sunview-Users Subset from the SunOS media (if you want to run SunView applications)
- use the `-nosunview` option with the `openwin` command (if you do not want to run SunView applications)

## Sun Applications Not Supported Under the OpenWindows Version 3 Release

The following Sun applications are not supported under the OpenWindows Version 3 release:

- SparcWorks Ada 1.1
- NSE 1.3

## X11/NeWS Server Issues

### 4-bit Visuals

NeWS and TNT are not supported on a 4-bit visual.

### `XDrawImageString()` and `XDrawImageString16()`

The X11/NeWS server may crash on the Xlib function calls, `XDrawImageString()` and `XDrawImageString16()`, on an GS framebuffer, when the default depth is 24 (e.g., `openwin -dev /dev/cg12 defdepth 24`). This will only occur when the width or the height of the window being drawn into is an out-of-range integer value (e.g., a negative number).

The workaround is to verify the window width and height. Instead of that verification, you can use the Xlib function call `XFillRectangle()` with `XDrawString()`, which will print an error message if the window width or height is an out of range value.

## International (8-bit) User Issues

If you choose the Format option from the Extras Menu in a `textsw` or `termw`, any 8-bit characters (characters with diacritical marks, such as accents) in the subwindow will be lost if `LANG` is not set to a European language, or `LC_CTYPE` is set to `C`, or either one is not set at all. To work around this problem, set `LC_CTYPE` to `iso_8859_1` in your `.profile` or `.cshrc` file, as described in Chapter 9.

**Frame Menus and Dynamic Locale Changes**

The frame menus managed by `o1wm` do not always adhere to dynamic locale changes. For example, you may initially change the Display Language locale through the Workspace Properties sheet successfully. However, subsequent locale changes will not affect the frame menu. To work around the problem, restart `o1wm` by exiting the window system and restarting `openwin`.

**Incorrect Sorting in File Manager and Mailtool (1053980)**

Sorting of file and directory names in File Manager and sorting of mail in Mailtool may be incorrect for locales other than the C locale.

**DeskSet Issues****Running Deskset Application Remotely**

If you are starting a Deskset application from a remote system that does not have OpenWindows stored in `/usr/openwin`, you will get the error:

```
ld.so: libxview.so.3 not found
```

To work around this problem, set the `LD_LIBRARY_PATH` environment variable on your remote system to `$OPENWINHOME/lib`.

**Cutting and Pasting Unprintable Characters**

When you Cut and Paste, or directly manipulate a line with non-printable characters, it may get truncated if you paste or drag it to a text field. Pasting or dragging to a `textsw` works as expected.

**Tape-Reading Limitations**

There is a limitation when reading a tape using the Read Entire List or the Read Selected option from `tapetool`. The total buffer size of the `tar` command cannot exceed 1024 bytes. Therefore, in the case of the Read Selected Option, the total length of all selected pathnames from the scrolling list cannot exceed 1024 bytes. In the case of the Read Entire List option, the total length of all filenames in the scrolling list cannot exceed 1024 bytes. If the total length of all files to be read does exceed 1024 bytes, `tapetool` will only read those files up to the 1024 byte maximum and will terminate the Read option with an alert message displaying the last file that it attempted to read.

**Problem Returning to OpenWindows Version 2 from OpenWindows Version 3**

If you use the OpenWindows Version 2 software after you have been running the OpenWindows Version 3 software, the Version 3 Calendar Manager may cause problems for the Version 2 Calendar Manager. To avoid this, back up your `/usr/spool/calendar/callog.<user>` file before starting the Version 3 Calendar Manager. If you later go back to Version 2, you should restore your old file.

**Problem with Mailtool Attachments from OpenWindows Version 3 System**

Be careful when you mail attachments from Mailtool to people who are not using the OpenWindows Version 3 Environment. Their mailer will not be able to unbundle the attachments in the same way as Mailtool. The attachments must in this case be unbundled by hand by using an editor and sometimes `uudecode(1)`; `uudecode` can easily unbundle one attachment, but the file



must be broken up into pieces if more than one attachment is mailed.

### Mailtool Swap Space Problem

Mailtool may run out of swap space if you drag-and-drop a file that is too large into the Mailtool Attachment window. The workaround is to increase your swap space.

### Calendar Manager Problems

If the Calendar Manager is not working correctly, make sure you have run the `install_openwin` script (as root).

## Toolkit/XView Issues

### Problem if No SunView

If your system is installed without SunView files or you use the `-nosunview` option to start the OpenWindows environment, you may encounter the following warning when you start up a Version 2 XView-based application:

```
System warning: No such file or directory, extras menu file
/usr/lib/.text_extras_menu (Textsw package)
```

You will not be able to use the extras menu in the text subwindow. To work around this problem, add the following to your `.Xdefaults` file:

```
text.extrasMenuFilename:
OPENWINHOME/lib/locale/C/xview/.text_extras_menu
```

where `OPENWINHOME` is the value of `$OPENWINHOME`.

### Selection Transactions Not Complete Successfully

If selection transactions do not complete successfully (e.g., Mailtool attachments do not display data, or drag-and-drop operations fail), it may be because you are using an eight-megabyte system or a system that is too heavily loaded. This is caused by selection timeouts. The selection timeout value for XView-based applications is the time a requesting application will wait for a response from the selection holder. In general, the default three-second timeout is sufficient for a selection transaction to complete.

If your selection transactions are failing, increase the default selection timeout to 10 seconds by putting the following in the `.Xdefaults` file in your home directory and then restarting the OpenWindows session:

```
selection.timeout: 10
```

**Complete OPEN LOOK  
Mouseless Model**

The complete OPEN LOOK Mouseless Model is only enabled when `OpenWindows.KeyboardCommands` is set to `FULL`. Refer to the *XView Programming Manual* for information.

**Compiling an XView  
Application Statically**

If you compile an XView application statically and do not specify `-lc` as a library to link against, a compiler warning is issued:

```
ld: /lib/libc.a(flshbuf.o): _fflush: multiply defined
```

The workaround is to specify `-lc` as a library to link against.

**Toolkit/OLIT Issues****Enter Key on Numeric  
Keypad**

If you use the keypad Enter key when the NumLock key is depressed (that is, when the keyboard is in numeric mode), it will work as specified. If you try to use it when NumLock is not depressed (cursor mode), it will not work correctly. To eliminate this problem, add the following to the user's defaults file for your application:

```
Olitapplication*returnKey: <Return>, <KP_Enter>
```

where `Olitapplication` is the name of the application.

To make the adjustment for all applications, add the following to `.Xdefaults`:

```
*returnKey: <Return>, <KP_Enter>
```

**XInput Extension**

OLIT supports the X input extension. Sample code demonstrating its use is available on the SUCCESS library (token number 6186).

**Problems When Using  
Monochrome Monitors**

Several difficulties may arise when you use a monochrome monitor:

- `FlatExclusives` and `FlatNonExclusives` widget labels disappear when the `inputFocusColor` matches the background color. This problem is much more likely to appear on a monochrome monitor since there are only two colors.
- When you use a `RectButton` in mouseless mode on a monochrome monitor, you will not be able to tell if the button is selected or not. Normally, in 2D mode you can tell if a button is selected because its border becomes bold. However when a `RectButton` receives input focus in mouseless mode, the border of the button is removed, preventing you from telling what the button state is.

- When you use a `TextField` widget on a monochrome monitor, the line and the arrows at the end of each line in the widget may not show up.

**Setting the  
XtNconsumeEvent  
Resource on a  
ScrollingList Widget.**

If you set the `XtNconsumeEvent` resource on a `ScrollingList` widget, the callback will never get called. The `ScrollingList` does not currently propagate events to its `ListPane` child.

**Destroying Popup Widgets**

If you destroy an unrealized `Popup` widget which contains a `TextEdit` widget, the program will die with an `Xlib` error.

**Destroying DrawArea  
Widgets**

OLIT applications that destroy `DrawArea` Widgets will crash.

**Setting Checkbox Text Font  
Color**

Setting the font color in the `checkbox` widget does not have any effect.

**No Compose Key Light with  
OLIT Applications**

The Compose key will work correctly in OLIT applications, but the Compose LED on the keyboard will not illuminate when Compose is active.

**Toolkit/TNT Issues**

**TNT Demos**

TNT demos, such as `jed` and `jet`, are only available on the "Advanced User with Demos" and "Full OpenWindows" clusters. See the *OpenWindows Version 3 Installation and Start-Up Guide* for details.

**No NeWS or TNT with 4-bit  
Visuals**

NeWS and TNT are not supported on a 4-bit visual.

**Color Changes in Workspace  
Properties Pop-up**

Whenever the `colormap` compaction utility is run it writes a file, `.owcolors`. When the server is restarted `.owcolors` is read to initialize the colormap. After the file is read, TNT colormap initialization fails. The symptom of this failure is that TNT applications do not respond dynamically when workspace colors are changed using the `WorkSpace Properties` pop up.

**Menu Repainting**

At certain times, the `SaveBehind` feature of TNT menus is erroneously invalidated. This invalidation causes repainting of the area under the menu when the menu is brought down, instead of repairing the area using the `SaveBehind` feature.

A patch has been provided with the `OpenWindows Version 3` release to minimize this behavior. You must copy it manually from the CD-ROM before you can install

it. To copy it, make sure that you are running as `root` and have set `$OPENWINHOME` to the location where you installed the OpenWindows software. Then enter the following:

```
example# cp -p /cdrom/OpenWindows/sun4/etc/NEWS/tnt/menu_fix.ps $OPENWINHOME/etc/NEWS/tnt
```

Replace `/cdrom` with the appropriate directory if you mounted the CD-ROM elsewhere.

After you have copied the patch by hand, you may install it in one of the following ways:

- Do the following, as `root`:

```
example# psh $OPENWINHOME/etc/NEWS/tnt/menu_fix.ps
```

This loads the patch into your running server, but only affects TNT applications started after installing the patch. If you restart the server, the patch must be reinstalled.

- Append `$OPENWINHOME/etc/NEWS/tnt/menu_fix.ps` to your `$HOME/.user.ps` file.

This installs the patch each time your server is restarted and affects all TNT applications started after the server is restarted.

## ToolTalk Service Issues

Due to a problem in ToolTalk 1.0, you should create dynamic message patterns for all `psignatures` and `osignatures` (static message patterns) listed in your `ptypes` and `otypes`. Register these dynamic message patterns with `tt_pattern_register()`. Do not use `tt_ptype_declare()`, which will automatically create and register message patterns based on the `psignatures` and `osignatures` in your `ptypes` and `otypes`. There is no problem using `ptypes` or `otypes`. `Ptypes` provide a start "string" for the ToolTalk service to automatically start your application when it's needed.

The static message patterns in your `ptype` and `otype` may be used when a message is addressed to an `otype` or contains instructions to start an application or queue the message. If no running process matches the message, the ToolTalk service looks at the `ptypes` and `otypes` it has read in from the Classing Engine to look for a matching static message pattern.

If you use `tt_ptype_declare()`, the following problem could surface. If two processes have declared the same `ptype` (using `tt_ptype_declare()`) in a ToolTalk session and one of the processes quits the session, the ToolTalk service removes the message patterns for that `ptype` without checking to see if another process has also registered that `ptype`. This leaves the remaining process with no registered message patterns (except those registered dynamically) and as a result, the remaining process no longer receives messages that match the patterns contained in its `ptype`.

This problem also affects processes that have different ptypes but their ptypes contain a pattern with the same op (operation name). When a process that has declared a ptype containing a pattern with the common operation name quits the ToolTalk session, the pattern with the common operation name is removed from memory. The remaining processes that declared a ptype that also contains the common operation name will no longer receive messages that match that particular pattern.

The following instructions for creating dynamic patterns that match your static patterns are in the form of code templates. Templates are given for ptype and otype definitions. In each template strings in <>'s denote variables and names in []'s denote optional variables. Refer to the *ToolTalk 1.0 Programmer's Guide* for more information on ptype and otype syntax.

The general form of a ptype definition is:

```
ptype <ptype_name> {
  <cat>:
    <scope> <op>(<arg1>, ..., <argN>) => [start]
    [queue] [opnum=<opn>]; }
```

Repeat the following template for each signature listed in the ptype definition:

```

declare_ptype_signature()
{
    Tt_pattern      p;

    p = tt_pattern_create();
    if (<cat> == "handle") {
        tt_pattern_category_set(p, TT_HANDLE);
    } else if (<cat> == "observer") {
        tt_pattern_category_set(p, TT_OBSERVE);
    }
    if (<scope> == "session") {
        tt_pattern_scope_add(p, TT_SESSION);
    } else if (<scope> == "file") {
        tt_pattern_scope_add(p, TT_FILE);
    } else if (<scope> == "both") {
        tt_pattern_scope_add(p, TT_BOTH);
    } else if (<scope> == "file_in_session") {
        tt_pattern_scope_add(p,
            TT_FILE_IN_SESSION);
    }
    tt_pattern_op_add(p, <op>);
    if (opnum is listed) {
        tt_pattern_opnum_add(p, <opn>);
    }
    tt_pattern_address_add(p, TT_PROCEDURE);
    if (start is listed) {
        tt_pattern_disposition_add(p,
            TT_START);
    }
    if (queue is listed) {
        tt_pattern_disposition_add(p,
            TT_QUEUE);
    }
    /* optionally, you can add a callback since
    this is a dynamic pattern*/
    tt_pattern_register(p);
}

```

This example shows how to create a dynamic pattern for the third message pattern listed in Sun\_EditDemo's ptype. Sun\_EditDemo is a sample program stored in \$OPENWINHOME/share/src/ooltalk/. The ptype is listed in the edit.type file in this directory.

```

/*
 * pattern corresponding to ptype signature:
 *     file_in_session Sun_EditDemo_save_as(in
 *     string new_filename)
 *     => opnum=SUN_EDITDEMO_SAVE_AS;
 */
declare_ptype_signature()
{
    Tt_pattern      p3;

    p3 = tt_pattern_create();
    tt_pattern_category_set(p3, TT_HANDLE);
    tt_pattern_scope_add(p3,
        TT_FILE_IN_SESSION);
    tt_pattern_op_add(p3,
        "Sun_EditDemo_save_as");
    tt_pattern_opnum_add(p3,
        SUN_EDITDEMO_SAVE_AS);
    tt_pattern_address_add(p3, TT_PROCEDURE);
    tt_pattern_arg_add(p3, TT_IN, "string",
        (char *)0);

    tt_pattern_register(p3);
}

```

The otype message pattern definition is similar to the ptype definition. Use this template to provide otype message patterns.

```

otype <otype_name> {
    <cat>:
        <op>(<arg1>, ..., <argN>) => <ptid>
        [<scope>] [start] [queue] [opnum=<opn>]; };

```

Repeat the following for each signature listed in the otype.

```

declare_otype_signature()
{
    Tt_pattern      p;
    p = tt_pattern_create();
    if (<cat> == "handle") {
        tt_pattern_category_set(p, TT_HANDLE);
    } else if (<cat> == "observer") {
        tt_pattern_category_set(p, TT_OBSERVE);
    }
    if (<scope> == "session") {
        tt_pattern_scope_add(p, TT_SESSION);
    } else if (<scope> == "file") {
        tt_pattern_scope_add(p, TT_FILE);
    } else if (<scope> == "both") {
        tt_pattern_scope_add(p, TT_BOTH);
    } else if (<scope> == "file_in_session") {
        tt_pattern_scope_add(p,
            TT_FILE_IN_SESSION);
    }
    tt_pattern_op_add(p, <op>);
    if (opnum is listed) {
        tt_pattern_opnum_add(p, <opn>);
    }
    tt_pattern_otype_add(p, <otype_name>)
    tt_pattern_address_add(p, TT_OTYPE);
    tt_pattern_address_add(p, TT_OBJECT);
    if (start is listed) {
        tt_pattern_disposition_add(p,
            TT_START);
    }
    if (queue is listed) {
        tt_pattern_disposition_add(p,
            TT_QUEUE);
    }
    /* optionally, you can add a callback since
    this is a dynamic pattern*/

    tt_pattern_register(p);
}

```

This example shows how to create a dynamic pattern for the message pattern listed in Sun\_EditDemo's otype. The otype is listed in the edit.type file in the \$OPENWINHOME/share/src/tooltalk directory.



```

/*
 * pattern corresponding to otype signature:
 * hilite_obj(in string objid)
 * => Sun_EditDemo file_in_session start
 * opnum=SUN_EDITDEMO_HILITE_OBJ;
 */
declare_otype_signature()
{
    Tt_pattern    p5;

    p5 = tt_pattern_create();
    tt_pattern_category_set(p5, TT_HANDLE);
    tt_pattern_scope_add(p5,
        TT_FILE_IN_SESSION);
    tt_pattern_op_add(p5, "hilite_obj");
    tt_pattern_opnum_add(p5,
        SUN_EDITDEMO_HILITE_OBJ);
    tt_pattern_otype_add(p5,
        "Sun_EditDemo_object");
    tt_pattern_address_add(p5, TT_OTYPE);
    tt_pattern_address_add(p5, TT_OBJECT);
    tt_pattern_disposition_add(p5, TT_START);
    tt_pattern_arg_add(p5, TT_IN, "string",
        (char *)0);

    tt_pattern_register(p5);
}

```

If arguments are listed in either the ptype or otype signature, use this template to add them to the pattern:

Case 1:

```
<op> ()
```

This case specifies that argument matching is not required for this signature so no action needs to be taken.

Case 2:

```
<op> (void)
```

This case specifies that this signature takes no arguments. No action is required.

## Case 3:

```

<op>(<model> <type1> <name1>, ..., <modeN>
<typeN> <nameN>)
if (<mode> == "in") {
    tt_pattern_arg_add(p, TT_IN, <type>, (char
    *)0);
} else if (<mode> == "out") {
    tt_pattern_arg_add(p, TT_OUT, <type>, (char
    *)0);
} else if (<mode> == "inout") {
    tt_pattern_arg_add(p, TT_INOUT, <type>, (char
    *)0);
}

```

In this case, you need to add arguments to the dynamic pattern. For an example of this, see the `Sun_EditDemo` ptype example.

## Supplemental Documentation

The following sections are supplements to the OpenWindows Version 3 documentation.

### NeWS Programming: NeWS Operators

The correct definition of the `reshapecanvas` operator differs from the definitions given in the *NeWS 3.0 Programmer's Guide* and in the *OpenWindows Version 3 Release Manual*. Most of the changes occur in the paragraphs regarding the X Shape Extension. The following is the correct definition for `reshapecanvas`:

```

canvas reshapecanvas -
canvas path reshapecanvas -
canvas borderwidth reshapecanvas -
canvas path borderwidth reshapecanvas -

```

If a *canvas* argument alone is specified, this operator sets *canvas*'s shape to be the same as the current path, and sets *canvas*'s default coordinate system to be the same as the current coordinate system. This results in a new *canvas* transformation matrix. If the current path is empty, with *canvas* as the only argument, this operator has no effect.

If *canvas* is the current canvas, an implicit `initmatrix` and `initclip` are performed. `initmatrix` sets the current transformation matrix to be the same as the new *canvas*'s transformation matrix. `initclip` sets the current context clip path to be the same as the new *canvas*'s shape. An implicit `newpath` is always performed to clear the current path.

If the current path is non-rectangular and the only argument, *canvas*, is an X canvas, *canvas*'s X Shape Extension is activated by setting *canvas*'s client bounding region to be the same as the current path and *canvas*'s default bounding region to be the same as the bounding box of the current path. *Canvas*'s effective bounding and/or clip regions are recomputed properly.

*Canvas*'s default bounding region is the rectangular area that covers both *canvas*'s client drawable interior and its border (as described in the X core protocol). *Canvas*'s default clip region is *canvas*'s default bounding region without the border. *Canvas*'s client bounding region is the client-defined bounding region, which can be non-rectangular. *Canvas*'s client clip region is the client-defined clip region, which can be non-rectangular. *Canvas*'s effective bounding region is the intersection of *canvas*'s default bounding region and client bounding region. *Canvas*'s effective clip region is the intersection of *canvas*'s effective bounding region, default clip region, and client clip region. These regions maintain constant distances from *canvas*'s X default origin, which is the upper left corner of *canvas*'s default clip region. See the *X11 Non-Rectangular Window Shape Extension* (MIT X Consortium Standard Version 1.0, Copyright 1989 by the Massachusetts Institute of Technology) for more information.

If the current path is rectangular and the only argument (*canvas*) is an X canvas, then *canvas*'s default bounding region is set to be the same as the current path. In this case, if *canvas*'s X Shape Extension is activated (i.e. if *canvas* has a client bounding region and or a client clip region), then *canvas*'s effective bounding and or clip regions are recomputed properly.

The *path* and or *borderwidth* arguments can be used only if *canvas* is an X canvas, otherwise this operator results in a *typecheck* error.

A non-empty current path and *path* activate *canvas*'s X Shape Extension (if it is not activated already) by setting *canvas*'s client bounding region to be the same as *path*. If the current canvas is different from the context canvas used when forming *path*, then *path* is pre-translated such that its distance to the upper left corner of the current canvas's default bounding region is the same as its previous distance to the upper left corner of its context canvas default bounding region. A non-empty current path and empty *path* result in the removal of *canvas*'s client bounding restriction. *Canvas*'s effective bounding and/or clip regions are recomputed properly.

An empty current path and *path* activate *canvas*'s X Shape Extension (if it is not activated already) by setting *canvas*'s client clip region to be the same as *path*. If the current canvas is different from the context canvas used when forming *path*, then *path* is pre-translated such that its distance to the upper left corner of the current canvas's default bounding region is the same as its previous distance to the upper left corner of its context canvas default bounding region. An empty current path and an empty *path* result in *canvas*'s client clip restriction being removed. *Canvas*'s effective clip region is recomputed properly.

*borderwidth* resets *canvas*'s default clip region and recomputes *canvas*'s effective clip region properly.

## TNT: Drag-and-Drop Changes

The following sections contain changes to `ClassSelection`, and `ClassTextCanvas` that didn't get included in the *NeWS Toolkit Reference Manual*. In addition, a new class, `ClassTextSelection`, is introduced below.

### *ClassSelection*

`ClassSelection` has a new variable and a new method that are used to implement drag and drop animation.

#### **/DragCursorType** (Variable)

Determines how **/setdragimage** (below) combines the drag-and-drop pointer glyphs with an application-specific image. The legal values are:

**/SourceImage**: Center the application image around the pointer glyph

**/Boxed**: Place the application image in the 'flying punchcard' box. **/Boxed** is the default.

**/SourceImage** drags should be used when dragging a whole object, e.g., an entire file. **/Boxed** drags should be used when dragging a portion of a larger object (e.g., some text from a file, one object from a drawing).

#### cursorimage maskimage **/setdragimage -**

Builds a drag-and-drop pointer using `cursorimage` and `maskimage`. `cursorimage` and `maskimage` can only be 1-bit deep canvases. Both `cursorimage` and `maskimage` must be given, but they can be the same canvas. The value of **/DragCursorType** (above) determines how the application-specific image and the drag-and-drop pointer glyphs are combined. **/setdragimage** is typically called from the **/DragStart** method.

The Toolkit provides two ways to set the drag-and-drop pointers for an application selection:

- Use **/DragCursorType** and **/setdragimage** to combine an application-specific image with the default pointer glyphs. `ClassTextCanvas` uses this combination to provide the flying punchcard text drag cursor. This is also the easiest way to merge application-specific images with the OPEN LOOK drag-and-drop pointer glyphs.
- Override the drag-and-drop pointer class variables in your `ClassCanvas` subclass. There are eight pointer variables that you can override: **/InsertMoveCursor**, **/InsertCopyCursor**, **/ReplaceMoveCursor**, **/ReplaceCopyCursor**, **/NeutralMoveCursor**, **/NeutralCopyCursor**, **/InvalidMoveCursor**, and the **/InvalidCopyCursor**. Overriding the canvas variables is a good mechanism if your drag-and-drop cursors aren't likely to change from instance to instance. `OpenWindows` has an `OLcursor` font that contains predefined cursors that cover several of the most common source image cases.

For more information on drag-and-drop pointers see the *OpenWindows Version 3 Desktop Integration Guide*.

*ClassTextSelection*

ClassTextSelection subclasses ClassSelection to set defaults and provide definitions for several ClassSelection subclass responsibility methods. ClassTextSelection does the following:

- Sets the default **/DragCursorType** to be **/Boxed**
- Sets the default cursors to be empty ‘flying punchcard’ cursors.
- Creates the **/CanRenderAs** dictionary with **/ContentsAscii**, **/ContentsPostScript**, and **/SelectionObjsize** as the keys. The value for each key is a proc that obtains the requested value. The proc executes during a call to **/SingleRequest**.
- Creates a dictionary named **/ActionRequests**. **/ActionRequests** is similar to **/CanRenderAs** but contains those requests with side-effects instead of values. Default requests are **/DeleteContents** and **/Canvas**. The **/Canvas** key does not denote an action, but it can’t go into **/CanRenderAs** because of the possibility of creating a sticky window if the selection is copied to the clipboard.
- Overrides **/AllRequests** to return a dict that combines **/CanRenderAs** and **/ActionRequests**.
- Overrides **/Deselect** to forward **/Deselect** to the Holder of the selection.
- Overrides **/SingleRequest** to expect the value from **/CanRenderAs** or **/AllRequests** to be the proc to execute to obtain the value.

*ClassTextCanvas*

ClassTextCanvas has changed to reflect the new drag-and-drop implementation. ClassTextCanvas subclasses ClassCanvas to provide assistance for clients whose selections are character strings and who want use the standard OPEN LOOK semantics for dragging and dropping text. ClassTextCanvas is Selectable, sets **/SelectableType** to be **/Text**, and defines **/DragStart**. You still must provide the definitions for the other selectable subclass responsibility methods in ClassCanvas.

**Creation**

parentcanvas **/new** instance  
Creates a text canvas.

**Drag Methods**

event selection **/DragStart** -  
Creates a text drag cursor that contains part of the selected text.

selection **/CurrentText** string

Obtains the text to display in the overlay canvas. Subclassers will generally wish to override the **/CurrentText** method for greater efficiency (the default uses the normal **/query** mechanism whereas individual subclasses can usually obtain the text by more direct methods).

## Running OpenWindows via xdm

`xdm(1)`, the X display manager, is used to control a collection of both local and remote displays, and is typically started by root. When OpenWindows runs via `xdm`, you see a login window on the screen, welcoming you to the system and asking for your username and UNIX password. The login capability is very rudimentary, in that a user's configuration files, such as `.login` and `.cshrc` or `.profile` are not executed.

After you successfully log in, `xdm` starts up your X environment, typically by reading the executable file named `.xinitrc` in your home directory, and runs an Xsession. On exiting from the window system, the window server is restarted by the `xdm` daemon.

A configuration file is usually provided to `xdm` at start-up. This file is used for initial `xdm` set-up. This file also contains pointers to other shell scripts used in by `xdm`. `xdm` is typically started by root. The user environment for root must be set-up correctly, as follows:

```
# set $OPENWINHOME /usr/openwin <or other location of OpenWindows>
# set path = ( $OPENWINHOME/bin $path )
# setenv LD_LIBRARY_PATH $OPENWINHOME/lib:/usr/lib
```

A typical command line, where the configuration file is provided to `xdm` at start-up is as follows:

```
# xdm -config $OPENWINHOME/lib/xdm/xdm-config
#
```

By default, `xdm` looks in `/usr/openwin/lib/xdm` to find the configuration files it requires. The configuration file and other shell scripts used by `xdm` are in `$OPENWINHOME/lib/xdm`. It is suggested that you start with this set of files if you wish to change the behavior of `xdm`. From its original X11R4 version, `xdm` has been modified slightly to allow it to work more efficiently with the OpenWindows X11/NeWS server. When started by root, `xdm` executes a window server which is listed in the file, `Xservers`. Once the server has initialized itself, the login window through which a user may log into the system displays.

### *xdm* Files

The following are some of the files used in conjunction with `xdm`:

- `$OPENWINHOME/lib/xdm/xdm-config`  
This file is used by `xdm`; see the man page for more information.
- `$OPENWINHOME/lib/xdm/Xservers`  
This file lists which server to run and whether to run local or remote:  
:0 Local local \$OPENWINHOME/lib/xdm/StartOW :0
- `$OPENWINHOME/lib/xdm/StartOW`  
This file contains the shell script used to pass the correct arguments to the OpenWindows server.
- `$OPENWINHOME/lib/xdm/Xsession`  
This file runs as the client for the display manager. It attempts to run a per-

user `.xinitrc`. It also forces the window server to change its `userid` to that of the new user.





---

## Notes

---

# Notes

---

## Notes

---

# Notes

---

## Notes

---

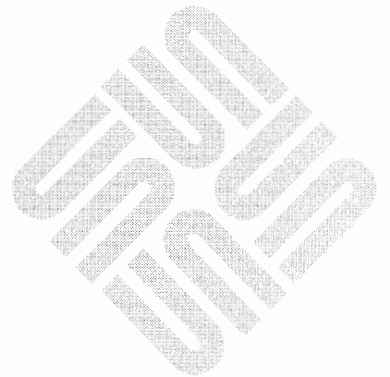
# Notes

---

## Part 5 — Appendixes

Part 5 consists of the following appendixes:

- Appendix A, “Documents Available for Use with SunOS 4.1.2”
- Appendix B, “Addendum to Writing Device Drivers”
- Appendix C, “Sundiag version 2.3 for SunOS 4.1.2”
- Appendix D, “X/Open Conformance Statement (XCS-QUE-3.106)”







---

## Documents Available for Use with SunOS 4.1.2

An extensive collection of software manuals is made available for use with SunOS 4.1.2. The manuals are shipped in binders that may contain one or more documents. The following describes the contents of the binders following the general order of use. This release manual and *Installing SunOS System Software* were described in Chapter 1, and are not listed here.

**Books:** *User's Guides*

These brief manuals are easy to read, written for the user with little or no UNIX experience. *Getting Started* is for those new to the Sun environment. Other titles are *Doing More with SunOS*, *Basic Troubleshooting*, *SunView User's Guide*, and *Customizing Your Environment*. The *SunDiag User's Guide* is in the same box, but is written for users who will be testing memory, drives, boards, and board-level devices with SunDiag.

**Binder:** *System and Network Administration*

The *System and Network Administration* manual is written for system administrators, but is a valuable resource for *all* SunOS users. It deals with adding hardware, disk maintenance, networking and electronic mail service, and advanced UNIX administration.

**Books:** *OpenWindows End User Manuals*

This is a set of four guides for the user who is new to OpenWindows:

*OpenWindows Version 2 User's Guide*  
*DeskSet Environment Reference Guide*  
*OpenWindows Version 2 Release Notes*  
*OpenWindows Version 2 Installation & Start-Up Guide*

**Binders:** *SunOS Reference Manual (3 Volumes)*

This set of binders contains the SunOS version of the Berkeley UNIX man pages, alphabetically arranged descriptions of commands, functions and other aspects of SunOS 4.1.\* Numbered tabs divide the *Reference Manual* into a range of topics, as listed below:

1. User Commands

---

\* New man pages for SunOS 4.1.2 are included in the release box.

2. System calls and error numbers
3. User-level library functions
4. Devices, drivers, protocols and network interfaces
5. File formats used or read by various programs
6. Games and demos
7. Public files, tables and TROFF macros
8. System maintenance and operations commands

**Binder: *Global Index***

Provides an index to SunOS 4.1 software documentation. It is not updated for SunOS 4.1.1, SunOS 4.1.1 Rev B, or SunOS 4.1.2.

**Binder: *SunOS Documentation Tools***

*Editing Text Files* covers the editors *vi*, *ex*, *ed*, and *sed*. *Formatting Documents* explains special formatting macros that work with TROFF such as *refer*, indexing, table formatting and equation setting. *Using TROFF and NROFF* explains the use of those text processing utilities.

**Binder: *Programmer's Guides***

This group of manuals focuses on programming within the Sun environment, for both systems and applications level interests.

*C Programmer's Guide* describes how to write C programs that interface with SunOS. *Assembly Language Reference Manual for Sun-3* and *Assembly Language Reference Manual for Sun-4* covers syntax and usage of the assembler for some of the microprocessors used in Sun workstations and servers. *A RISC Tutorial* looks at the aspects of RISC and open systems architectures as they pertain to SPARC architecture. *Porting Software to SPARC Systems* briefly describes machine level SPARC architecture, and porting C, FORTRAN, and Pascal programs to a SPARC system. The *Debugging Tools Manual* describes the debuggers *dbx*, *dbxtool*, and *adb* for experienced programmers.

**Binder: *Programmer's Overview Utilities and Libraries***

The *System Services Overview* contains details of various specialized aspects of SunOS operating system, including internationalization, security features, networking, and UNIX standards compatibility. *Programming Utilities and Libraries* is written primarily for applications programmers to provide an overview of the Sun environment, and the system facilities, utilities, and libraries supported.

**Binder: *Network Programming Guide***

The *Network Programming Guide* provides an overview of NFS, pipes, sockets, network commands, Sun online database service, and network managers and monitors.

**Binder:** *Writing Device Drivers/STREAMS Programming*

*Writing Device Drivers* is a guide to adding drivers for new hardware devices to the SunOS kernel. The *STREAMS Programming* manual covers the theory of STREAMS programming, the SunOS-specific implementation, and catalogs STREAMS functions and data structures.

**Binder:** *SunView Programmer's Guide*

*SunView 1 Programmer's Guide* is written for applications programmers to support interactive, graphics-based applications running within windows. The *SunView 1.80 Update Appendix* updates the *SunView 1 Programmer's Guide* with the latest information.

**Binder:** *SunView 1 System Programmer's Guide*

The *SunView 1 System Programmer's Guide* describes how SunView works from the inside, and how to structure applications. The *Pixrect Reference Manual* describes the *Pixrect* graphics library routines that manipulate arrays of pixel values, and *RasterOps* used by applications programs to manipulate bit-mapped displays.



---

## Addendum to Writing Device Drivers

This addendum has two parts:

- Sun-4m Device Driver Developer Notes
- SunOS 4.1.1 Changes

### B.1. Sun-4m Device Driver Developer Notes

Due to architectural differences between 600MP and previous Sun machines, there are few key points that device driver developers must be aware of:

- 600MP supports both VME and SBus. There is also an I/O cache (IOC) (which actually behaves more like a write buffer) on the VME bus interface.
- 600MP has a separated IOMMU which is capable of doing "coherent DVMA" with respect to the CPU caches.

With a separated IOMMU, DVMA mappings are now independent of the host SRMMU mappings. This means that by default a DVMA address should be considered as only valid from the point of view of devices that will be doing DVMA, and it should be considered invalid from the point of view of a CPU (and your driver, since it runs on a CPU). Similarly, a host SRMMU virtual address should not be used by a device to perform DVMA.

As a result of these new features, 600MP supports at least 16MB (also default) of DVMA space. This DVMA space is divided into 4 regions (maps):

vme24map: despite its name, a vme32 device can use this map too. This is the default map for all VME devices. This map is about 1MB.

vme32map: only vme32 devices can use this map. This map is about 6MB.

sbusmap: default map for SBus devices. This map is about 1MB.

bigsbmap: an optional map for SBus devices. This map is about 8MB.

NOTE: although there are now four maps, for compatibility reasons, a VME driver should still use `mb_hd.mh_map` as an argument to the `mb_XXX` routines. Similarly, SBus devices should use `dvmapap` to interface with the `mb_XXX` routines. See "Performance Tuning" below for information about using the larger vme32map and bigsbmap.

## Compatibility

Most drivers can probably run on a 600MP series machine without recompilation, but some may require some modifications. The following is a list of points to consider to determine if a existing driver is portable to a 600MP series machine:

- Does the driver use only standard driver supporting routines provided by the kernel, such as `mb_mapalloc/mb_mapfree` (and their `mb_XXX` variations) to manage DVMA mappings?

Note: `segkmem_XXX` and `hat_XXX` are not considered as standard driver supporting routines.

If your driver uses non-standard routines to set up DVMA mappings, it most likely will NOT work on a 600MP series machine (since `mb_XXX` routines are the only routines that set up DVMA mappings on the IOMMU) These drivers MUST be converted over to use the standard `mb_XXX` routines if they are to be portable.

- Does the driver use `DVMA[some_offset]` to reference/modify data inside the DVMA data transfer buffers?

If the answer is yes to this question, then:

- if it is a VME device, it will run as is. However, it is advised that the driver should use `bp_mapin()` and `bp_mapout()` routines for better portability.
- if it is an SBus device, you need to convert it to use `bp_mapin()/bp_mapout()`. Otherwise, the driver will pick up random values.

This is due to the fact that DVMA addresses are now independent of the host SRMMU virtual address. For compatibility reasons, the DVMA mappings in `vme24map` (default map for vme devices) is set up such that `DVMA[]` reference would still work. However, the same thing can not be done on the SBus maps, so it is no longer compatible with other Sun machines.

- Does your VME device run with IOC off?  
IOC only works on VME devices. IOC is turned on/off, flushed automatically by the `mb_mapalloc()/mb_mapfree()`. Currently, there is no easy way for a driver to disable IOC.
- Does the driver use the `ioc_flush()` macro to flush IOC?

If yes, the driver needs to be recompiled on a SPARCsystem 600MP series machine; the binary will not be portable to other machines.

Other than the points listed above, a driver should be written exactly the same way as if it were written for other machines. A driver written for 600MP series machines should work for the other machines but not necessarily vice versa since the rules for 600MP are more restrictive. The only exception to this rule is if your driver uses non-default, large DVMA maps which are not supported on other machines (see "Performance Tuning").

## Interrupts

Hardware interrupt levels for Sun-4m are different than for Sun-4 or Sun-4c. Refer to the Sun-4m System Architecture document for details.

The Sun-4m mapping of SBus and VMEbus interrupt levels to SPARC interrupt request level (IRL) is different from previous Sun-4 and Sun-4c architectures. A device which interrupts at SBus level *x*, which is mapped by the onboard interrupt logic to SPARC IRL *y* on Sun-4 or Sun-4c, will now be mapped to SPARC IRL *z* on Sun-4m and *z* may be greater than *y*. This may introduce bugs which typically manifest themselves as corrupted data structures leading to kernel crashes.

For Sun-4m, device drivers should not make Networking or STREAMS framework function calls while operating at SPARC IRL levels higher than IRL 7 (`splimp`) for networking, or IRL 10 (`splstr`) for STREAMS. Doing so circumvents the interrupt masking being done by the networking and STREAMS subsystems themselves and risks data structure corruption. Device drivers should do minimal processing at high interrupt levels and schedule a software interrupt for further interrupt processing, including interacting with other portions of the kernel.

For Sun-4m, it is OK to call networking and mbuf utility routines while servicing a device interrupt at VME and SBus levels 4 and below. It is OK to call STREAMS utility routines at VME and SBus levels 5 and below.

In general, device drivers which support multiple devices which interrupt at more than one hardware interrupt level must take precautions to service only those interrupts at the "current" interrupt priority level (via `spltoipl()`) in order to avoid race conditions which result in the error message "Level XXX BBB interrupt not serviced."

## Write Buffers

Write buffers are used to accelerate writes and reduce bus occupancy for better overall system performance. Write buffers exist both for programmed I/O and DVMA activity. Use of the `mb_XXX()` routines guarantees correct operation. All write buffers in the Sun-4m architecture follow these rules:

- Once a write buffer has accepted a write, it must either guarantee that the write can occur without error, or the write buffer is responsible for reporting those errors,
- Write buffers are read-stall; that is, after a write buffer has accepted a write, any subsequent access to that device must wait for the write operation to complete (order is maintained). Although write buffers are not visible to device drivers, their effect may be. While order is maintained, the relative timing of writes to the device may be significantly different from the issuing (CPU) timing.

## Performance Tuning

To optimize the performance for drivers running on 600MP series machines, drivers can do the following:

**vme32 devices**

If the driver does not peek inside the DVMA data buffers without `bp_mapin()` / `bp_mapout()`, it may set the `MDR_VME32` flag to use the much larger `vme32map`, instead of the smaller default `vme24map`. If the DVMA request size is not larger than 1MB, adding this flag should not cause problems in terms of portability. It will be simply ignored by machines that do not have this map.

**SBus Devices**

SBus devices can set the `MDR_BIGSBUSMAP` flag to use the big 8MB map. If it uses the big map, it must NOT add the DVMA base to form the DVMA address passed to the device. `Mb_mapalloc()` already returns a correct "ready to use" DVMA address. Usage of this map makes the driver NON-portable.

NOTE: driver should not use `vme32map` or `bigsbusmap` as arguments to the `mb_XXX` routines. Instead, use flags as described.

**IOMMU Bypass Mode**

If a driver uses IOMMU bypass mode, it will be responsible for its own DVMA mappings. The standard DVMA supporting routines described above will not be useful for them.

**SBus Slot Configuration Register**

There is an SBus slot configuration register for each SBus slot in the system. Each SBus slot configuration register provides information about the slave device in that slot (slave support for 64-, 32-, 16-, and 8-byte bursts), and is also used for IOMMU bypass management for that slot. The boot code is expected to configure the slot based upon FCodes associated with the SBus device. Refer to the Open Boot PROM V2 Reference Manual. Failure of the device firmware FCode to support this property will result in less than optimal slave access performance for the device as only 4-byte word sized slave transfers will be used.

**Open Boot PROM**

SPARCsystem 600MP is released with version 2 of the open boot PROM firmware. Refer to the OBP V2 Reference Manual for details.



## B.2. SunOS 4.1.1 Changes

This appendix is an addendum to *Writing Device Drivers for Sun Workstations*; it replaces a similar document which was distributed with the SunOS 4.1 PSR A Release Manual. It contains important information if you are planning to implement an I/O device driver for a DVMA device that runs on a machine with an I/O Cache (IOC). Machines with an I/O cache include the Sun3/4X0 series (Sun-3/460, Sun-3/470, Sun-3/480) and the SPARCsystem 4X0 series (SPARCstation470, SPARCserver 470, SPARCserver 490).

The IOC provides a Write-Back buffer for VME I/O devices capable of initiating accesses to Main Memory. On Sun machines, I/O devices use virtual addresses for memory access. Direct Memory Access using virtual addresses is referred to as DVMA, and devices capable of initiating DVMA are called DVMA devices.

The SPARCsystem 4X0 IOC consists of 128 sets of 32-byte-wide data buffers. The Sun-3/4X0 IOC consists of 128 sets of 16-byte-wide data buffers. Except for this difference, the two caches are identical. The following description will assume the SPARCsystem 4X0 IOC.

Each data buffer corresponds to an 8K page, so that serial transfers through a particular DVMA page are buffered into 32-byte blocks on their way to or from memory. For example, a DVMA write to memory will use the same IOC buffer repeatedly as it serially transfers a page: each 32-byte block on the page is flushed to memory by the next 32-byte block on the same page. At the end of each such DVMA write there will be one line in the IOC for each page of the transfer that will contain valid data that has yet to be written to memory.

The IOC is designed to buffer data into blocks to make long serial transfers more efficient. When an IOC line is flushed, all 32 bytes are flushed, even if some bytes of the line were not modified. Despite its name, the IOC is really a buffer. When transferring from a device to memory the cache line is **not** filled from main memory before the line is updated with data from the device. This makes using the IOC appropriate for buffering data transfers, but inappropriate for buffering I/O control block information which is shared between the device and the CPU.

Drivers allocate and release DVMA using mb routines described in the *Writing Device Drivers* manual. The kernel mb routines generally account for the presence of the IOC in ways that will be transparent to the driver. This mechanism is detailed here so an implementer can decide if it is suitable for a particular driver.

The kernel mb routines enable the Sun3/4X0 and SPARCsystem 4X0 IOCs for a particular buffer using the criteria shown below.

```

if (bp->b_flags & B_READ) {
    if (((u_int)bp->b_un.b_addr & IOC_LINEMASK) == 0) &&
        ((bp->b_bcount & IOC_LINEMASK) == 0))
        set_ioc = 1;
    else /* misaligned */
        set_ioc = 0;
} else /* B_WRITE */
    set_ioc = 1;

```

If the starting address of the transfer is on an I/O cache line boundary, i.e., the address is a multiple of the constant `IOC_LINESIZE` and the length is also a multiple of `IOC_LINESIZE`, the IOC is used. Note that all writes are I/O cached as writes correspond to reads by DVMA devices, which entail no flushes and have no alignment requirements (extra bytes may be loaded from memory into the IOC, but the DVMA device gets only bytes from the addresses it requests from the IOC and there is no chance for corruption of data).

Upon release of the I/O cached DVMA space the kernel `mb` routines will flush any remaining data left in the IOC. This means that a driver for a device that performs DVMA in a serial fashion to aligned buffers will automatically make use of the IOC without requiring any code changes.

There are some special cases:

- What if I want a driver to use the IOC even though transfers aren't properly aligned?

In some cases, for instance DVMA to a buffer in user space, the alignment of the buffer into which data is received can't be changed and you can not force the IOC to be used. In other cases, for instance DVMA to a buffer managed by the driver itself, the buffer can be appropriately manipulated so the criteria for use of the IOC is met. This is generally done by allocating `IOC_LINESIZE-1` more bytes of DVMA than needed and rounding up the starting address to be aligned with `IOC_LINESIZE`. Also the size of the DVMA requested is rounded up to `IOC_LINESIZE` even though the amount actually DVMA'd may be less.

- What if I don't want the IOC to be used?

The I/O cache can be turned off for a DVMA transfer by making sure the `pg_ioc` bit in a DVMA page's Page Table Entry (PTE) is set to zero. This should be done after DVMA space is allocated with the kernel `mb` routines but before any DVMA has occurred. The `pg_ioc` bit should be cleared for each page of DVMA involved in a transfer.

Note that on a sun3x, this is the `pte_iocache` bit.

- What if DVMA space is reused without releasing and reallocating it with the kernel `mb` routines?

In this case either the transfer can be set up so the IOC is not used or the IOC can be manually flushed at whatever points are necessary to insure all data has reached memory. Flushing the I/O cache is accomplished with use of the `ioc_flush()` macro. The argument to `ioc_flush()` is the page offset into DVMA for the transfer. If the transfer spans more than one page then `ioc_flush()` will need to be called for each page.

## Header File Inclusion

Drivers that use either the `IOC_LINESIZE` constant or the `ioc_flush()` macro should include the `machine/iocache.h` header file. Proper inclusion is shown in the code fragment example below.

**Manual IOC Flush Example**

In the following code fragment example, the IOC is being manually flushed for a range of DVMA addresses following a transfer. `dvma_addr` is the byte offset into DVMA for the transfer. `dvma_count` is the number of bytes the device transferred.

```
#ifdef IOC
#include <machine/iocache.h>
#endif IOC
...
#ifdef IOC
    if (ioc) {
        end_addr = dvma_addr + dvma_count;

        for (addr = dvma_addr; addr < end_addr; addr += MMU_PAGESIZE) {
            linenum = (addr >> MMU_PAGESHIFT) & (IOC_CACHE_LINES - 1);
            ioc_flush(linenum);
        }
    }
#endif IOC
```

**Changes Since Earlier Releases**

As of SunOS 4.1 and SunOS 4.1 PSR A, the code for managing the Sun-3/470 IOC and the SPARCsystem 4X0 IOC was inconsistent. In SunOS 4.1.1, Sun3x IOC management was changed to be compatible with Sun4 IOC management. This means that drivers compiled to use the SunOS 4.1 `B_IOCTL_CACHE` flag to use the Sun3x IOC will not compile under 4.1.1.



---

## Sundiag version 2.3 for SunOS 4.1.2

The Sundiag diagnostic program, version 2.3, is shipped with SunOS 4.1.2. A brief description of the version 2.3 enhancements follows; see the *Sundiag 2.3 User's Guide - SunOS 4.1.2* (Part Number 800-6020-10) for details.

NOTE: Sundiag currently does not run with the OpenWindows window interface. If you are running OpenWindows, exit the window system before starting Sundiag. Sundiag will work with Sunview windows.

The Sundiag diagnostics program includes a variety of new tests and features for SunOS 4.1.2. Tests for Sun's new hardware include the following:

- Multi-Processing CPUs (details below)
- Graphics Tower
- SunVideoPix
- Sunlink HSI
- DRI/S boards
- Serial Parallel Controller
- Prestoserve
- SBus Printer cards
- cg12 (GS) Graphics Accelerator Board

The following existing tests have been enhanced:

- kmem
- nettest
- rawtest
- fstest
- sunlink

The following options have been added to the Sundiag program:

- w Writes the system hardware configuration to the `/usr/adm/sundiaglog/sundiag.conf` file.

- b Allows you to run Sundiag in batch mode. In order to use this option, you must first create a `batch_file` in `/usr/diag/sundiaglog/configs` before using Sundiag. The `batch_file` must use the following format:

#option file	runtime	delay_before_loading_next_option_file (min.)
optfile1	60	3
optfile2	1020	5
optfile1	60	16
optfile3	0	0

`optfile1` and `optfile2` are files containing Sundiag tests. They will run for the times specified (in minutes) in the runtime column. Files with a runtime of 0 display the final status of tests that have already run. This feature can be used to give the status of some or all of the option files in the `options_file`.

The `delay_before_loading_next_option_file` is required to ensure that all tests have been stopped before the next option file is loaded.

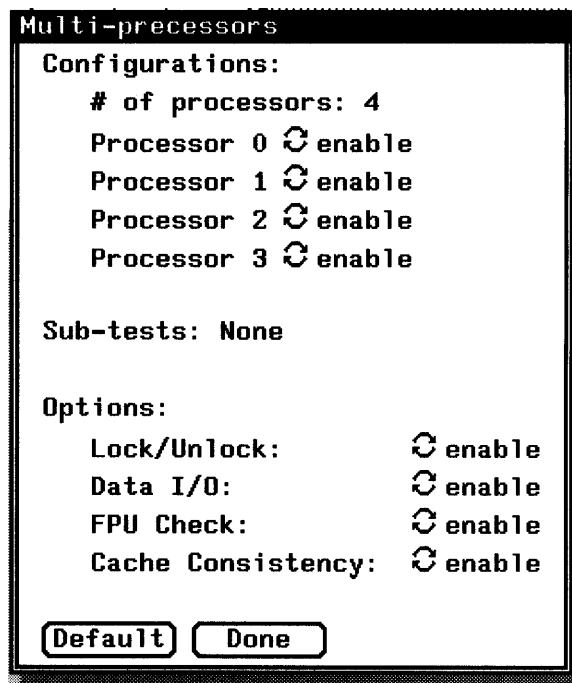
The settings in the Sundiag Options Menu override the settings in the batch files. Use larger values in the Options Menu when using batch files and avoid using the single pass values.

### C.1. Sundiag for the Multiprocessing Chip

The new Sundiag multi-processor test, `mptest`, verifies the functionality of Sun's multi-processing CPUs. There are four test Options to `mptest`: Lock/Unlock, Data I/O, FPU Consistency, and Cache Consistency. As with other Sundiag tests, you can specify options in either of two ways — from the SunView Sundiag window interface or with command-line arguments.

#### Running Sundiag From Window Interface

Sundiag automatically probes for multiprocessing CPUs when it is brought up. If a multiprocessing CPU is found, `mptest` is displayed as an available test in the Sundiag SunView window. Next to the `mptest` entry is an Options button. If that button is selected with the Left mouse button, the following Option Menu appears:

Figure C-1 *mptest* Option Menu**Configurations**

The processors available for test are listed here. The multi-processing test can be enabled/disabled for individual processors on this menu. The enable/disable toggle switch next to each processor is set by clicking the left mouse button on the arrow circle. Subtests There are no subtest currently supported by this test. Options The following options can be run singly or concurrently.

**Subtests**

No subtests are currently supported.

**Options**

The following options are available for *mptest*:

**Lock/Unlock**

Each processor uses the SPARC atomic instruction `ldstub` as a lock/unlock to write to a shared physical memory page. While one processor is attempting the write, the other processors are free spinning for their turn to write.

**Data I/O**

Each processor, in turn, writes data to the disk. The other processors immediately read and verify the data pattern.

**FPU Consistency**

The processors under test do floating point and integer numeric computations simultaneously, and verify identical results.

**Cache Consistency**

This option requires two or more processors to access the same physical address. This option verifies that a change in physical address by one processor is confirmed by another.

**Command-line Syntax**

Sundiag `mptest` can be run from a SunView window or a TTY port; `mptest` can run from the command line using the following syntax:

```
/usr/diag/sundiag/mptest T=n C=n M=x standard_arguments
```

Insert arguments for T, C, and M from the following table; see the *Sundiag 2.3 User's Guide - SunOS 4.1.2* for a description of *standard\_arguments*.

<b>Argument</b>	<b>Description</b>
<b>T=<i>n</i></b>	<p><i>n</i> is one of the following integers, representing the bit pattern of the test options to be enabled. If T= is not specified, the default is to enable all options</p> <ul style="list-style-type: none"> <li>1 = Shared Memory Enabled</li> <li>2 = Data I/O Enabled</li> <li>3 = Data I/O and Shared Memory Enabled</li> <li>4 = FPU Check Enabled</li> <li>5 = FPU Check and Shared Memory Enabled</li> <li>6 = FPU Check and Data I/O Enables</li> <li>7 = FPU Check, Data I/O, and Shared Memory Enabled</li> <li>8 = Cache Consistency Enabled</li> <li>9 = Cache Consistency and Shared Memory Enabled</li> <li>10 = Cache Consistency and Data I/O Enabled</li> <li>11 = Cache Consistency, Data I/O, and Shared Memory Enabled</li> <li>12 = Cache Consistency and FPU Chick Enabled</li> <li>13 = Cache Consistency, FPU Check, and Shared Memory Enabled</li> <li>14 = Cache Consistency, FPU Check, and Data I/O Enabled</li> <li>15 = Cache Consistency, FPU Check, Data I/O, and Shared Memory Enabled.</li> </ul>
<b>C=<i>n</i></b>	<p><i>n</i> is one of the following integers, representing the bit pattern of the processors to be enabled. If C= is not specified, the default is to enable all available processors.</p> <ul style="list-style-type: none"> <li>1 = processor0 Enabled</li> <li>2 = processor1 Enabled</li> <li>3 = processor2 Enabled</li> <li>4 = processor3 Enabled</li> <li>5 = processor0 and processor3 Enabled</li> <li>6 = processor1 and processor2 Enabled</li> <li>7 = processor0, processor1, and processor2 Enabled</li> <li>8 = processor3 Enabled</li> <li>9 = processor0 and processor3 Enabled</li> <li>10 = processor1 and processor3 Enabled</li> <li>11 = processor0, processor1, and processor3 Enabled</li> <li>12 = processor2 and processor3 Enabled</li> <li>13 = processor0, processor2, and processor3 Enabled</li> <li>14 = processor1, processor2, and processor3 Enabled</li> <li>15 = processor0, processor1, processor2, and processor3 Enabled</li> </ul>
<b>M=<i>x</i></b>	<p><i>x</i> is the test pattern expressed as a hexadecimal number. The range is from 0 to 0xff. The default is 0x10.</p>



## C.2. Error

`mpctest` may return one of the following error messages. These messages specify problem with specific command-line arguments:

ERROR: Must be super-user  
User is not in super-user mode.

Cannot open <file>  
The disk is full. The Data I/O test file can not create a /tmp directory, or open it if it exists.

Couldn't map address space to device, <system error message>

The mmap process, which maps the Data I/O test file to memory, has failed.

Fail get share memory, <error number>  
Failure to get the shared memory segment.

Fail to fork!  
A new process could not be spawned.

Fail <single/double> precision FPU test, process <number>, processor <number>, observed = <pattern>, expected = <pattern>  
FPU test failed on a specify process, and processor.

Fail to open /dev/kmem  
/dev/kmem could not be opened for Cache Consistency.

Fail to open /dev/mem  
/dev/mem could not be opened for Cache Consistency.

Ioctl error, errno = <number>, sys\_errlist = <error message>  
The kernel support ioctl failed.

Please specify the 'T=' argument again  
The integer specified for the "T=n" argument was not acceptable;  
enter another

Please specify the 'C=n' argument again  
The integer specified for the "C=n" argument was not acceptable;  
enter another.

Can not run Multi-Processors test! You need to enable at least two processors.

There was an error in the non Multi-Processors environment. Add another processor from the `mpctest` Option Menu, or by using the "C=n" option.



---

## X/Open Conformance Statement (XCS-QUE-3.106)

This appendix provides the X/Open Conformance Statement for all Sun and Sun-based software products that have received the XPG3 Component brand. X/Open's publication procedure prints the conformance statement as a series of chapters, one for each component. Each component is assigned a specific chapter number. This appendix starts with the conformance statement for the Commands and Utilities component, which X/Open assigns to Chapter 3. The following XCS-QUE-3.106 conformance chapters are printed here:

- Chapter 3: Commands and Utilities
- Chapter 4: C Language
- Chapter 5: Indexed Sequential Access Method
- Chapter 8: Pascal Language
- Chapter 9: Fortran Language
- Chapter 10: SQL
- Chapter 11: Terminal Interfaces
- Chapter 12: Window Management
- Chapter 14: Inter-Process Communication
- Chapter 15: Source Code Transfer
- Chapter 16-1: Ada Language (from Telesoft AB)
- Chapter 16-2: Ada Language (from Verdix Corporation)

## Chapter 3: COMMANDS AND UTILITIES

PRODUCT IDENTIFICATION	Product Identification	SunOS
	Version/Release No.	4, from 4.1 on
	If you do not supply this component yourself, please identify below the supplier you reference.	
CONFORMANCE REFERENCE	Indicator of Compliance	None
ENVIRONMENT SPECIFICATION	Enter below details of the hardware and software environment in which testing took place, including compilation routines and installation procedures (if any). Sufficient detail must be supplied to enable conformant behaviour and any test results to be reproduced.	
	SunOS 4.1 running on a Sun4	
CONFORMANCE EXPECTATIONS	Volume 1 of XPG3 recognises that convergence of implementations towards a common specification for commands and utilities is not yet complete and therefore does not require a vendor to supply all of the commands and utilities (and individual options) specified in XPG3.	
	This chapter explicitly identifies those commands and utilities not supplied by the vendor and any supplied which do not conform to the published specification. (Reference : XPG3 Volume 1 Page 1).	

## Section 3.1: BASIC UTILITIES

### 3.1.1 SUPPORTED COMMANDS

**Question 1:** *Which of the basic utilities (non-development utilities) defined in the XPG are not provided with the implementation?*

**Answer:**

The **iconv**, **mailx** and **mkfifo** commands are not provided.

Rationale

The XPG Volume 1 states that “this volume in its current form is useful only as a guide to portability, but it is not possible to precisely define or test conformance to it”. This question determines whether or not the implementation provides a command of the name specified in the XPG, it does not attempt to determine whether it supports the semantics of that command. The (optional) development utilities are excluded from this question and are dealt with in the next section of the questionnaire.

Reference

XPG3 Volume 1 Page 1 - Introduction

### 3.1.2 COMMAND BEHAVIOR

**Question 2:** *In what ways do the commands provided by the implementation behave differently from the specifications contained in the XPG?*

**Answer:**

Command	Behaviour Differences from XPG3
at	Uses /var/spool/cron/* instead of /usr/lib/cron/*
batch	Uses /var/spool/cron/* instead of /usr/lib/cron/*
crontab	Uses /var/spool/cron/* instead of /usr/lib/cron/*
awk	Not completely 8-bit clean
pack	Accepts file names up to 12 bytes long
sh	The ulimit built-in is not provided

Rationale

This question provides a greater degree of granularity than the previous question, requiring the semantic differences associated with the commands to be specified. Again, the question relates to the basic utilities rather than the development utilities. The question only relates to the semantics of the options specified within the XPG; implementation specific extensions should not be documented.

## Section 3.2: DEVELOPMENT UTILITIES

### 3.2.1 SUPPORTED COMMANDS

**Question 3:** *Which of the development utilities defined in the XPG are not provided with the implementation?*

**Answer:**

The optional **dis** and **sdb** utilities are not provided.

**Rationale**

The XPG Volume 1 states that “The development utilities might not be present in all X/Open compliant systems; in designated (**DEVELOPMENT**) systems all of the development utilities must be present and must conform to the published definition”.

**Reference**

XPG3 Volume 1 Page 2 - Status of Interfaces

### 3.2.2 COMMAND BEHAVIOR

**Question 4:** *In what ways do the development utilities provided by the implementation behave differently from the specifications contained in the XPG?*

**Answer:**

Command	Option	Behaviour Differences from XPG3
admin	-a!name -fi[str] -flist	Not supported Optional string argument not supported Does not provide all generality specified by XPG3
cc	-f	Requires option argument specifying type of floating point support to be provided
lex		Not 8-bit clean
lint	-o LIB	Creates library with name <b>llib-LIB.in</b> as in all known historical implementations instead of with name <b>LIB</b> as specified by XPG3
make		Search order for makefiles is makefile, SCCS/s.makefile, Makefile, SCCS/s.Makefile instead of makefile, Makefile, s.makefile, s.Makefile
nm	-e -f	Not supported Not supported

**Rationale**

This question provides a greater degree of granularity than the previous question, requiring the semantic differences associated with the development utilities to be specified.

### Section 3.3: INTERNATIONALISATION OPTION

#### 3.3.1 COMMANDS AND UTILITIES

**Question 5:** *Is an internationalised environment, reflecting changes in the locale setting as described in XPG Volume 1 - XSI Commands and Utilities, supported?*

**Answer:**

Sun does not claim support for internationalised commands and utilities in SunOS 4, although the following table does indicate that some of the options described in XPG3 are supported in this release.

Command	Behaviour Specified in XPG3	Supported
ar	LC_TIME affects date format	No
awk	LC_COLLATE, LC_CTYPE affect regular expression matching	No
	LC_COLLATE affects the behaviour of string comparisons	No
	LC_NUMERIC affects the behaviour of the radix character	No
comm	LC_COLLATE affects sorting sequence	No
cp,ln,mv	LANG affects yes string	No
cpio	LC_COLLATE, LC_CTYPE affect filename pattern matching	No
	LC_TIME affects date format	No
date	LC_TIME affects date formatting options	No
ed,red	LC_COLLATE, LC_CTYPE affect regular expression matching	No
	LC_CTYPE is used to determine whether characters are printable	Yes
egrep	LC_COLLATE, LC_CTYPE affect regular expression matching	No
	LC_CTYPE is used to determine character classification (alphabetic, upper-case, lower case)	Yes
expr	LC_COLLATE, LC_CTYPE affect regular expression matching	No
	LC_COLLATE affects the behaviour of relational operators	No
fgrep	LC_CTYPE is used to determine character classification (alphabetic, upper-case, lower case)	Yes
find	LANG affects yes string	No
	LC_COLLATE, LC_CTYPE affect filename pattern matching	No
grep	LC_COLLATE, LC_CTYPE affect regular expression matching	No
	LC_CTYPE is used to determine character classification (alphabetic, upper-case, l to discover case)	Yes
join	LC_COLLATE affects sorting sequence	No
lpstat	LC_TIME affects date format	No
ls	LC_COLLATE affects sorting sequence	No
	LC_CTYPE is used to determine whether a character is printable	Yes

Command	Behaviour Specified in XPG3	Supported
	LC_TIME affects date format	No
mail	LC_TIME affects date format	No
mailx	LC_COLLATE, LC_CTYPE affect filename pattern matching	No
	LC_TIME affects date format	No
pg	LC_COLLATE, LC_CTYPE affect filename pattern matching	No
pr	LC_TIME affects date format	No
	LC_CTYPE is used to determine whether a character is printable	Yes
ps	LC_TIME affects date format	No
rm,rmdir	LANG affects yes string	No
sed	LC_COLLATE, LC_CTYPE affect regular expression matching	No
	LC_CTYPE is used to determine whether a character is printable	Yes
sh	LC_COLLATE, LC_CTYPE affect filename pattern matching	No
	LC_CTYPE is used to determine whether a character is alphabetic	Yes
sort	LC_COLLATE affects sorting sequence	No
	LC_CTYPE affects character classification (alphabetic, upper case, printing)	Yes
	LC_NUMERIC affects the determination of the radix character	Yes
tar	LC_TIME affects date format	No
	LANG affects yes string	No
tr	LC_COLLATE, LC_CTYPE affect bracketed expressions	No
	LC_CTYPE affects the definition of the character universe	Yes
uniq	LC_COLLATE affects sorting sequence	No
uucp	LC_TIME affects date format	No
uustat	LC_TIME affects date format	No
wc	LC_CTYPE is used to determine white-space characters	Yes
who	LC_TIME affects date format	No
yacc	LC_CTYPE is used to determine character classification	Yes

#### Rationale

This behaviour is collectively optional; that is, it should be provided for all commands listed (subject to sections 3.1 and 3.2, which identify those commands not supplied by the vendor and those which do not fully support the X/Open specification).

#### Reference

XPG3 Volume 1 Pages 4-5 - Status of Interfaces.



### 3.3.2 REGULAR EXPRESSIONS IN COMMANDS

**Question 6:** *Which form of regular expression syntax is supported by those commands which use regular expressions?*

**Answer:**

Command	Regular Expression Syntax Supported
awk	Extended
csplit	Simple
ed	Simple
egrep	Extended
ex	Simple
expr	Simple
grep	Simple
lex	Extended
pg	Simple
sed	Simple
vi	Simple

#### Rationale

The XPG Volume 3 - XSI Supplementary Definitions requires that an internationalised set of commands will provide regular expression syntax for the above commands in one of the forms specified for that command. The XPG encourages the implementation of internationalised regular expressions for all of the above utilities. It should be noted that the **sdb** command is an optional development utility and may not be available on all XPG conforming systems.

#### Reference

XPG3 Volume 3 Pages 49-51 - Regular Expressions

## Chapter 4: C LANGUAGE

### PRODUCT IDENTIFICATION

Product Identification C compiler bundled with SunOS  
Version/Release No. 4, from 4.1 on

If you do not supply this component yourself, please identify below the supplier you reference.

### CONFORMANCE REFERENCE

Indicator of Compliance  
VSX Test Suite Release 3.203  
Testing Agency Name Sun Microsystems  
Address 2550 Garcia Avenue  
Mountain View CA 94043

### ENVIRONMENT SPECIFICATION

Enter below details of the hardware and software environment in which testing took place, including compilation routines and installation procedures (if any). Sufficient detail must be supplied to enable conformant behaviour and any test results to be reproduced.

SunOS 4.1 running on a Sun4

### TEMPORARY WAIVERS

List below references to any temporary waivers granted by X/Open in respect of minor errors in the product referenced above. This should include the X/Open reference and the waiver expiry date. The waivers as granted shall be made available with this document on request.

### 4.1 IMPLEMENTATION LIMITS

**Question 1:** *What limits does the implementation impose on the significant part of a identifier?*

**Answer:**

External identifiers 32 characters  
Non-External identifiers 127 characters

#### Rationale

The XPG states that, while there is no limit to the length of an identifier, only a certain number of characters are significant. The XPG points out that there must be at least eight characters for a non-external name, but may be less for external names.

#### Reference

XPG 3 Volume 4 Page 3 - Lexical Conventions

## 4.2 GENERAL

**Question 2:** *What truncation rules are applied when a floating value is converted to an integral value?*

**Answer:**

Truncation of floating point values is always towards zero.

**Rationale**

The XPG states that such conversions are machine dependent. In particular, the XPG points out the differences related to the truncation of negative numbers.

**Reference**

XPG Volume 4 Page 10 - Conversions

**Question 3:** *What truncation rules are applied when using the division operator and either of the operands is negative?*

**Answer:**

Truncation towards zero

**Rationale**

The XPG states that such truncations are machine dependent.

**Reference**

XPG Volume 4 Page 16 - Expressions

## Chapter 5: INDEXED SEQUENTIAL ACCESS METHOD

### PRODUCT IDENTIFICATION

Product Identification NetISAM  
Version/Release No. 1.0 and subsequent releases

If you do not supply this component yourself, please identify below the supplier you reference.

### CONFORMANCE REFERENCE

Indicator of Compliance  
VSX Test Suite Release 3.201  
Testing Agency Name Sun Microsystems  
Address 2550 Garcia Avenue  
Mountain View CA 94043

### ENVIRONMENT SPECIFICATION

Enter below details of the hardware and software environment in which testing took place, including compilation routines and installation procedures (if any). Sufficient detail must be supplied to enable conformant behaviour and any test results to be reproduced.

SunOS 4.1 running on a Sun4

### TEMPORARY WAIVERS

List below references to any temporary waivers granted by X/Open in respect of minor errors in the product referenced above. This should include the X/Open reference and the waiver expiry date. The waivers as granted shall be made available with this document on request.

### 5.1 IMPLEMENTATION LIMITS

**Question 1:** *What limits does the implementation impose for the following aspects?*

**Answer:**

Number of key parts in an index	NPARTS	8
Maximum size of a key	MAXKEYSIZE	150
Maximum number of indices per file	—	16
Maximum size of data record	—	8192
Maximum number of concurrently locked records	—	5000
Maximum number of ISAM files that can be concurrently open	—	128

### Rationale

The XPG states that the values for NPARTS and MAXKEYSIZE are implementation specific but must be at least 8 and 120 respectively. The XPG states that the definition puts no limit on the number of alternative indices that can be created for a file. However, it is known that some implementations impose a limit rather than allowing for a potentially infinite number of indices. The XPG also remains silent about the maximum length of a data record, though it is again known that some implementations impose a limit. The XPG states that the number of records that may be concurrently locked when a file has been opened with ISMANULOCK set is system dependent. This number may depend on the locks that are concurrently applied to other files and the number specified in the answer should be the maximum number of concurrently locked records that could be attained. The XPG states that an application program must be able to have at least ten ISAM files concurrently open. This limit assumes that the process has no other files open. The limit given in answer to the above question must also be based upon this assumption.

### Reference

XPG 3 Volume 5 Page 61 - The <isam.h> Header  
XPG 3 Volume 5 Page 7 - ISAM Overview  
XPG 3 Volume 5 Page 20 - Manual Record Locking  
XPG 3 Volume 5 Page 71 - Files.

## Chapter 8: PASCAL LANGUAGE

### PRODUCT IDENTIFICATION

Product Identification Sun Pascal  
Version/Release No. 2.0 and subsequent releases

If you do not supply this component yourself, please identify below the supplier you reference.

### CONFORMANCE REFERENCE

Indicator of Compliance  
NIST Test Suite Release 5.1  
Testing Agency Name NIST  
Address Gaithersburg, MD  
NIST Certificate Number NIST-89/2663

### ENVIRONMENT SPECIFICATION

Enter below details of the hardware and software environment in which testing took place, including compilation routines and installation procedures (if any). Sufficient detail must be supplied to enable conformant behaviour and any test results to be reproduced.

SunOS 4.0 on a Sun4

### TEMPORARY WAIVERS

List below references to any temporary waivers granted by X/Open in respect of minor errors in the product referenced above. This should include the X/Open reference and the waiver expiry date. The waivers as granted shall be made available with this document on request.

A temporary waiver has been granted for Sun Pascal, because the NIST testing was performed on SunOS 4.0 rather than SunOS 4.1. The waiver is granted until December 6, 1991. The X/Open reference number on the waiver request is:

PG3.083

The waiver request follows:

Pascal is not tested by VSX; instead, X/Open accepts a NIST certificate as the indicator of compliance. As the Trade Mark License Agreement is written, the NIST certificate must be from a test run on a system that is XPG branded. We have a NIST certificate from a test run on a version of the operating system prior to the version that will be XPG branded.

We are requesting a temporary waiver for the Pascal component because the NIST certificate for our Pascal product is from a test run on a version of the operating system prior to the version that will be XPG branded. The Pascal product executes correctly on either version of the operating system; that is, the identical binary image of the Pascal compiler works correctly on the

version of the operating system that NIST used to run their tests, and on the version of the operating system that will be XPG branded. This testing discrepancy has zero impact on application portability. Note that this temporary waiver is removed not by fixing the product, but by asking NIST to re-test the product on the version of the operating system with the XPG brand.

For example, suppose that Acme Computers had earned an XPG brand for ACME O/S 6.3, but had asked NIST to test ACME PASCAL on ACME O/S 6.2 (which does not have an XPG brand). Acme Computers would not be able to get an XPG brand for ACME PASCAL until NIST re-tested the compiler on ACME O/S 6.3. Since the identical binary image of ACME PASCAL runs on both ACME O/S 6.2 and ACME O/S 6.3, it is clear that the lack of the XPG brand for ACME PASCAL does NOT reflect product quality, but only the order in which testing was planned. Therefore, Acme Computers would request a temporary waiver allowing them to use the XPG brand on ACME PASCAL until such time as NIST completed the re-test of ACME PASCAL on ACME O/S 6.3.

## Chapter 9: FORTRAN LANGUAGE

### PRODUCT IDENTIFICATION

Product Identification Sun Fortran  
Version/Release No. 1.3 and subsequent releases

If you do not supply this component yourself, please identify below the supplier you reference.

### CONFORMANCE REFERENCE

Indicator of Compliance  
NIST Test Suite Release 2.0  
Testing Agency Name NIST  
Address Gaithersburg, MD  
NIST Certificate Number NIST-89/2004

### ENVIRONMENT SPECIFICATION

Enter below details of the hardware and software environment in which testing took place, including compilation routines and installation procedures (if any). Sufficient detail must be supplied to enable conformant behaviour and any test results to be reproduced.

SunOS 4.0 running on a Sun4

### TEMPORARY WAIVERS

List below references to any temporary waivers granted by X/Open in respect of minor errors in the product referenced above. This should include the X/Open reference and the waiver expiry date. The waivers as granted shall be made available with this document on request.

A temporary waiver has been granted for Sun Fortran, because the NIST testing was performed on SunOS 4.0 rather than SunOS 4.1. The waiver is granted until December 6, 1991. The X/Open reference number on the waiver request is:

PG3.084

The waiver request follows:

Fortran is not tested by VSX; instead, X/Open accepts a NIST certificate as the indicator of compliance. As the Trade Mark License Agreement is written, the NIST certificate must be from a test run on a system that is XPG branded. We have a NIST certificate from a test run on a version of the operating system prior to the version that will be XPG branded.

We are requesting a temporary waiver for the Fortran component because the NIST certificate for our Fortran product is from a test run on a version of the operating system prior to the version that will be XPG branded. The Fortran product executes correctly on either version of the operating system; that is, the identical binary image of the Fortran compiler works correctly on



the version of the operating system that NIST used to run their tests, and on the version of the operating system that will be XPG branded. This testing discrepancy has zero impact on application portability. Note that this temporary waiver is removed not by fixing the product, but by asking NIST to re-test the product on the version of the operating system with the XPG brand.

For example, suppose that Acme Computers had earned an XPG brand for ACME O/S 6.3, but had asked NIST to test ACME FORTRAN on ACME O/S 6.2 (which does not have an XPG brand). Acme Computers would not be able to get an XPG brand for ACME FORTRAN until NIST re-tested the compiler on ACME O/S 6.3. Since the identical binary image of ACME FORTRAN runs on both ACME O/S 6.2 and ACME O/S 6.3, it is clear that the lack of the XPG brand for ACME FORTRAN does NOT reflect product quality, but only the order in which testing was planned. Therefore, Acme Computers would request a temporary waiver allowing them to use the XPG brand on ACME FORTRAN until such time as NIST completed the re-test of ACME FORTRAN on ACME O/S 6.3.

## Chapter 10: SQL

### PRODUCT IDENTIFICATION

Product Identification Informix OnLine  
Version/Release No. 4.0

If you do not supply this component yourself, please identify below the supplier you reference.

Informix Software Inc  
4100 Bohannon Drive  
Menlo Park CA 94025

### CONFORMANCE REFERENCE

Indicator of Compliance  
None

### ENVIRONMENT SPECIFICATION

Enter below details of the hardware and software environment in which testing took place, including compilation routines and installation procedures (if any). Sufficient detail must be supplied to enable conformant behaviour and any test results to be reproduced.

SunOS 4.1 running on a Sun4

### TEMPORARY WAIVERS

List below references to any temporary waivers granted by X/Open in respect of minor errors in the product referenced above. This should include the X/Open reference and the waiver expiry date. The waivers as granted shall be made available with this document on request.

## Chapter 11: TERMINAL INTERFACES

### PRODUCT IDENTIFICATION

Product Identification SunOS

Version/Release No. 4, from 4.1 on

If you do not supply this component yourself, please identify below the supplier you reference.

### CONFORMANCE REFERENCE

Indicator of Compliance  
None

### ENVIRONMENT SPECIFICATION

Enter below details of the hardware and software environment in which testing took place, including compilation routines and installation procedures (if any). Sufficient detail must be supplied to enable conformant behaviour and any test results to be reproduced.

SunOS 4.1 running on a Sun4

### TEMPORARY WAIVERS

List below references to any temporary waivers granted by X/Open in respect of minor errors in the product referenced above. This should include the X/Open reference and the waiver expiry date. The waivers as granted shall be made available with this document on request.

## Chapter 12: WINDOW MANAGEMENT

### PRODUCT IDENTIFICATION

Product Identification    OpenWindows

Version/Release No.    1.0 and subsequent releases

If you do not supply this component yourself, please identify below the supplier you reference.

### CONFORMANCE REFERENCE

Indicator of Compliance  
None

### ENVIRONMENT SPECIFICATION

Enter below details of the hardware and software environment in which testing took place, including compilation routines and installation procedures (if any). Sufficient detail must be supplied to enable conformant behaviour and any test results to be reproduced.

SunOS 4.1 running on a Sun4

### TEMPORARY WAIVERS

List below references to any temporary waivers granted by X/Open in respect of minor errors in the product referenced above. This should include the X/Open reference and the waiver expiry date. The waivers as granted shall be made available with this document on request.

## Chapter 14: INTER-PROCESS COMMUNICATION

### PRODUCT IDENTIFICATION

Product Identification SunOS

Version/Release No. 4, from 4.1 on

If you do not supply this component yourself, please identify below the supplier you reference.

### CONFORMANCE REFERENCE

Indicator of Compliance  
None

### ENVIRONMENT SPECIFICATION

Enter below details of the hardware and software environment in which testing took place, including compilation routines and installation procedures (if any). Sufficient detail must be supplied to enable conformant behaviour and any test results to be reproduced.

SunOS 4.1 running on a Sun4

### TEMPORARY WAIVERS

List below references to any temporary waivers granted by X/Open in respect of minor errors in the product referenced above. This should include the X/Open reference and the waiver expiry date. The waivers as granted shall be made available with this document on request.

## Chapter 15: SOURCE CODE TRANSFER

### 15.1 UTILITIES

#### PRODUCT IDENTIFICATION

Product Identification SunOS

Version/Release No. 4, from 4.1 on

If you do not supply this component yourself, please identify below the supplier you reference.

#### CONFORMANCE REFERENCE

Indicator of Compliance  
None

#### ENVIRONMENT SPECIFICATION

Enter below details of the hardware and software environment in which testing took place, including compilation routines and installation procedures (if any). Sufficient detail must be supplied to enable conformant behaviour and any test results to be reproduced.

For Source Code Transfer software:

SunOS 4.1 running on a Sun4

For floppy disk hardware:

SPARCstation with internal floppy disk drive (part number X554H) or with external floppy disk subsystem (part number 550D-S)

For magnetic tape hardware:

SPARCserver Office server or Datacenter server (Sun 4/260, Sun 4/280, Sun 4/390 or Sun 4/490) with 1/2-inch tape drive subsystem (part number 675A)

#### TEMPORARY WAIVERS

List below references to any temporary waivers granted by X/Open in respect of minor errors in the product referenced above. This should include the X/Open reference and the waiver expiry date. The waivers as granted shall be made available with this document on request.

#### FORMATS

**Question 1:** Which exchange media format(s) may be **written** by the system?

**Answer:**

80 track floppy disk	Yes
40 track floppy disk	No
1600bpi PE magnetic tape	Yes

Rationale

XPG3 states that standards are referenced for transfer of floppy discs and magnetic tapes between machines. Because of the different nature of X/Open conformant systems, it is not possible to define a single portable medium which is supported across the whole range of systems.

Reference

XPG3 Volume 3 Chapters 15, 16 and 17

**Question 2:** Which exchange media format(s) may be read by the system?

**Answer:**

80 track floppy disk	Yes
40 track floppy disk	No
1600bpi PE magnetic tape	Yes

Rationale

XPG3 states that standards are referenced for transfer of floppy discs and magnetic tapes between machines. Because of the different nature of X/Open conformant systems, it is not possible to define a single portable medium which is supported across the whole range of systems. In addition, some systems can read a wider range of formats that they can write.

Reference

XPG3 Volume 3 Chapters 15, 16 and 17

UTILITIES

**Question 3:** Which utilities are used to create and read the archive formats specified in XPG Volume 3 - XSI Supplementary Definitions?

**Answer:**

Format	Creating	Reading
Extended tar	<b>pax</b>	<b>pax</b>
Cpio	<b>pax and cpio</b>	<b>pax and cpio</b>

Rationale

There is no explicit definition as to the commands that must be used to create and retrieve these archives. On most systems this will be achieved by the **tar** and **cpio** commands. There are other commands available which produce these archives. On some implementations the command may need a special option to enable reading of the specified formats with the "standard" option being to create archives which are backwards compatible with previous versions of the command.

Reference

XPG3 Volume 3 Chapter 18

INVALID FILE NAMES

**Question 4:** *What file name is used to contain data from the archive in the case that the file name on the archive is invalid for the system on which the file hierarchy is being created?*

**Answer:**

Format	File Name
Extended tar	All legal file names in a USTAR archive are legal in the filesystem.
Cpio	All legal file names in a cpio archive are legal in the filesystem.

Rationale

Because an archive can contain non-portable file names it is necessary for an archive reading utility to be able to generate a file and store the data associated with a non-portable file name when this is encountered on the archive. There may be a need to generate a number of such file names in the same directory and the specification should detail the algorithm used to generate these file names.

Reference

XPG3 Volume 3 Pages 151,155

MULTI VOLUME ARCHIVES

**Question 5:** *How does the archive reading utility determine which file to read as the next volume when an end-of-file or end-of-media condition is encountered?*

**Answer:**

Format	Method
Extended tar	The <b>pax</b> utility prompts the user for the pathname of the next file in the archive. (The path need not name a device.)
Cpio	The <b>cpio</b> and <b>pax</b> utilities prompt the user for the pathname of the next file in the archive. (The path need not name a device.)

Options:

Description of method used by each utility.

Refer to: POSIX.1 Conformance Document Section 10.1.3.

Rationale:

In many cases the utility will prompt the user for the path name of the device to use for the next volume. There may be extensions to the utility syntax which allow the definition of alternate addresses for subsequent volumes.



## Chapter 16: ADA LANGUAGE

### PRODUCT IDENTIFICATION

Product Identification    TeleGen2 Ada Development System  
Version/Release No.    1.4

If you do not supply this component yourself, please identify below the supplier you reference.

TeleSoft  
CONTACT: Joan Giannetta  
5959 Cornerstone Court West  
San Diego CA 92121

### CONFORMANCE REFERENCE

Indicator of Compliance

ADA Joint Program Office  
Test Suite Release    1.10

Testing Agency Name    Wright-Patterson AFB Ada Validation Facility  
Address    Wright-Patterson Air Force Base  
Dayton OH

ADA Joint Program Office  
Certificate Number    890801W1.10134

### ENVIRONMENT SPECIFICATION

Enter below details of the hardware and software environment in which testing took place, including compilation routines and installation procedures (if any). Sufficient detail must be supplied to enable conformant behaviour and any test results to be reproduced.

SunOS 4.0 running on a Sun4

### TEMPORARY WAIVERS

List below references to any temporary waivers granted by X/Open in respect of minor errors in the product referenced above. This should include the X/Open reference and the waiver expiry date. The waivers as granted shall be made available with this document on request.

A temporary waiver has been granted for TeleSoft TeleGen2, because the AJPO testing was performed on SunOS 4.0 rather than SunOS 4.1. The waiver is granted until December 6, 1991. The X/Open reference number on the waiver request is:

PG3.085

The waiver request follows:

Ada is not tested by VSX; instead, X/Open accepts an AJPO certificate as the indicator of compliance. As the Trade Mark License Agreement is

written, the AJPO certificate must be from a test run on a system that is XPG branded. We have an AJPO certificate from a test run on a version of the operating system prior to the version that will be XPG branded.

We are requesting a temporary waiver for the Ada component because the AJPO certificate for our Ada product is from a test run on a version of the operating system prior to the version that will be XPG branded. The Ada product executes correctly on either version of the operating system; that is, the identical binary image of the Ada compiler works correctly on the version of the operating system that AJPO used to run their tests, and on the version of the operating system that will be XPG branded. This testing discrepancy has zero impact on application portability. Note that this temporary waiver is removed not by fixing the product, but by asking AJPO to re-test the product on the version of the operating system with the XPG brand.

For example, suppose that Acme Computers had earned an XPG brand for ACME O/S 6.3, but had asked AJPO to test ACME ADA on ACME O/S 6.2 (which does not have an XPG brand). Acme Computers would not be able to get an XPG brand for ACME ADA until AJPO re-tested the compiler on ACME O/S 6.3. Since the identical binary image of ACME ADA runs on both ACME O/S 6.2 and ACME O/S 6.3, it is clear that the lack of the XPG brand for ACME ADA does NOT reflect product quality, but only the order in which testing was planned. Therefore, Acme Computers would request a temporary waiver allowing them to use the XPG brand on ACME ADA until such time as AJPO completed the re-test of ACME ADA on ACME O/S 6.3.

## Chapter 16: ADA LANGUAGE

**PRODUCT IDENTIFICATION**

Product Identification    Verdex Ada Development System VAda-110-4040  
 Version/Release No.    6.0

If you do not supply this component yourself, please identify below the supplier you reference.

Verdex Corporation  
 CONTACT: Roger Baker  
 14130-A Sullyfield Circle  
 Chantilly VA 22021

**CONFORMANCE REFERENCE**

Indicator of Compliance

ADA Joint Program Office  
 Test Suite Release            1.10

Testing Agency Name        Wright-Patterson AFB Ada Validation Facility  
 Address                        Wright-Patterson Air Force Base  
    Dayton OH

ADA Joint Program Office  
 Certificate Number            890216W1.10030

**ENVIRONMENT SPECIFICATION**

Enter below details of the hardware and software environment in which testing took place, including compilation routines and installation procedures (if any). Sufficient detail must be supplied to enable conformant behaviour and any test results to be reproduced.

SunOS 4.0 running on a Sun4

**TEMPORARY WAIVERS**

List below references to any temporary waivers granted by X/Open in respect of minor errors in the product referenced above. This should include the X/Open reference and the waiver expiry date. The waivers as granted shall be made available with this document on request.

A temporary waiver has been granted for Verdex VADS, because the AJPO testing was performed on SunOS 4.0 rather than SunOS 4.1. The waiver is granted until December 6, 1991. The X/Open reference number on the waiver request is:

PG3.085

The waiver request follows:

Ada is not tested by VSX; instead, X/Open accepts an AJPO certificate as the indicator of compliance. As the Trade Mark License Agreement is written, the AJPO certificate must be from a test run on a system that is XPG



branded. We have an AJPO certificate from a test run on a version of the operating system prior to the version that will be XPG branded.

We are requesting a temporary waiver for the Ada component because the AJPO certificate for our Ada product is from a test run on a version of the operating system prior to the version that will be XPG branded. The Ada product executes correctly on either version of the operating system; that is, the identical binary image of the Ada compiler works correctly on the version of the operating system that AJPO used to run their tests, and on the version of the operating system that will be XPG branded. This testing discrepancy has zero impact on application portability. Note that this temporary waiver is removed not by fixing the product, but by asking AJPO to re-test the product on the version of the operating system with the XPG brand.

For example, suppose that Acme Computers had earned an XPG brand for ACME O/S 6.3, but had asked AJPO to test ACME ADA on ACME O/S 6.2 (which does not have an XPG brand). Acme Computers would not be able to get an XPG brand for ACME ADA until AJPO re-tested the compiler on ACME O/S 6.3. Since the identical binary image of ACME ADA runs on both ACME O/S 6.2 and ACME O/S 6.3, it is clear that the lack of the XPG brand for ACME ADA does NOT reflect product quality, but only the order in which testing was planned. Therefore, Acme Computers would request a temporary waiver allowing them to use the XPG brand on ACME ADA until such time as AJPO completed the re-test of ACME ADA on ACME O/S 6.3.

---

# Index

## *Special Characters*

FORTRAN 1.2, installation of  
compatibility with SunOS release 4.1.2, 7-15

## **A**

about SPARCsystem 600MP systems, 2-6  
add\_client command  
if command fails, use rm\_client before retrying, 10-3  
add\_services command  
adding new release requires 350 KB in /usr, 10-2  
aliases in open boot PROM for device path names, 5-1  
ALM-2 serial ports in GENERIC kernel file, 8-7  
application and kernel architecture, 1-3  
application architecture, 1-3  
architecture, application and kernel, 1-3  
assembler, SPARC  
optimization level -O2 produces incorrect code, 10-9  
automounter  
may cover /home on system mounting /home from local  
disk, 10-2  
Sun386i server kit causes all clients to automount /home,  
10-18

## **B**

boxes, use of, 1-5

## **C**

CD-ROM  
actions causing system to hang, 10-21  
eject command may fail after read failure, 10-21  
must be at sr0 for SunInstall, 10-21  
some error messages can be disregarded, 10-22  
CG6 frame buffer  
screen errors with some SPARC 390/4X0 system PROMs,  
10-17  
compatibility of releases, 1-1  
compiler, SPARC  
incorrect code at optimization level -O2, 10-9  
conventions, documentation, 1-5

## **D**

DeskSet, 3-1  
Desktop SPARCsystem  
definition, 1-4  
/home on second disk, 8-10  
rebuilding the kernel, 8-3

Desktop SPARCsystem, *continued*  
swap space on second disk, 8-10

desktop tutorial, 3-5

devalias command, 5-6

device driver  
for SPARCsystem 4X0, B-5  
for Sun-3/4X0, B-5

device names  
open boot PROM, 5-4

differences  
SunOS 4.1.1, SunOS 4.1.1 Rev B, & SunOS 4.1.2, 2-1

disk drives  
moving from SunOS 4.1/4.1.X to system under earlier release,  
10-3

documentation  
conventions, 1-5  
guide, 1-5

Domain Name Service (DNS)  
hostname required in "host" files, 10-7

## **E**

error messages  
Alignment Error, 10-13  
assertion failed panic:bn! UFS\_hole, 10-18  
Bus Error, 10-13  
eject: Open fail on cd, 10-21  
fatal IO error 32 (Broken pipe), 10-23  
have you run install\_cmgr?, 3-4  
Inappropriate ioctl for device, 10-23  
ld.so: libsuntool.so.0.## not found, 7-4  
Memory address not aligned, 10-14  
mount: [...] I/O error; mount: [...] giving up on, 10-8  
No label found - attempting boot anyway, 10-14  
Not enough space in sd0a, 10-2  
panic: insufficient virtual space, 10-4  
panic: data fault, 10-14  
RPC program not registered, 10-6  
rpc.cmsd is not responding..., 3-4  
sense key(0x1): soft error, 10-22  
sr0a: read recoverable, 10-22  
su: uucico: illegal option, 10-8  
time out on RFS primary name server, 10-4  
watchdog reset, 10-18  
Watchdog Reset!, 10-4  
extract\_unbundled  
problem installing software on multiple release server, 7-1

**F**

framebuffer  
 problem redirecting I/O to ttya on SPARCstation 2 with CG6,  
 10-14

**G**

guide to publications, 1-5

**H**

hierarchical device naming, 5-4

**I**

I/O cache, SPARCsystem 4X0, B-5  
 install\_cmgr command, 3-4

**K**

kadb  
 causes console tty to hang, 10-5  
 kernel  
 and application architecture, 1-3  
 architecture, 1-3  
 customized for large servers, 8-6  
 new configuration file for Desktop SPARCsystems, 8-3

**L**

language products, 7-11

**M**

machine names & system architecture, 1-3  
 maxcontig parameter  
 cannot be greater than 7 with tunefs -a, 10-8  
 maxusers, changing value in kernel for large systems, 8-6  
 memory  
 8MB required by OpenWindows, 7-3  
 miniroot  
 can't be booted with some SPARC 390/4X0 system PROMs,  
 10-15  
 sun-4/330 cannot boot from internal disk with some PROMs,  
 10-14  
 MP, how it works, 2-6  
 multiprocessor, how it works, 2-6

**N**

NeWS (Network extensible Window System) window system, 3-1  
 NeWS Toolkit, 3-1  
 NSE (Network Software Environment)  
 NSE Release 1.2 not usable with SunOS 4.1.2, 7-5  
 NVRAM, 5-4

**O**

online desktop tutorial, 3-5  
 online Openwindows tutorial, 3-5  
 OPEN LOOK  
 graphical user interface, 3-1  
 Intrinsic Toolkit, 3-1  
 open boot PROM, 4-4, 5-4  
 OpenWindows, 3-1  
 deskset environment, 3-1  
 requires large swap partitions, 7-3  
 running install\_cmgr, 3-4

OpenWindows

tutorial, 3-5

OpenWindows

using as default window system, 3-3

OpenWindows window system, 7-3

exiting window system may generate error messages, 10-23

foreground color not reset, 10-23

SunView libraries required, 7-4

too large for 104MB system disk, 7-3

**P**

ports, AlM-2 serial \_\_\_\_\_ in GENERIC kernel file, 8-7

PROM

open boot, 5-4

open boot PROM, 4-4

problem booting miniroot from internal disk on Sun-4/330,  
 10-14

some prevent booting miniroot on SPARC 390/4X0 systems,  
 10-15

publications guide, 1-5

**R**

release compatibility, 1-1

RFS

problem starting primary name server only, 10-4

**S**

SCSI bus

support for five on SPARCsystem 600MP series systems, 4-4

show-devs command, 5-5

software features, new, 2-1

SPARC 390/4X0 systems

problem booting miniroot with some PROMs, 10-15

some PROMs cause CG6 screen errors, 10-17

SPARCprinter

do not install SunOS 4.1 patch under 4.1.1, 7-4

SPARCstation 1

problem upgrading from SunOS 4.0.3 to SunOS 4.1.1, 10-13

SPARCstation 2

system with CG6 has problem redirecting I/O to ttya, 10-14

SPARCsystem 600MP system description, 2-6

Sun386i

correction to *Sun386i SunOS 4.0.2 Installation Guide*, 10-21

server kit causes all clients to automount /home, 10-18

SunInstall

can only find CD-ROM at sr0, 10-21

SunLink BSC3270

mistake in *SunLink BSC3270 System Administration Guide*,  
 7-10

SunOS

differences in 4.1.1, 4.1.1 Rev B, 4.1.2, 2-1

SunView window system

SunView\_Users category required for OpenWindows, 7-4

swap space

20MB required by OpenWindows, 7-3

using a UNIX file for local swap, 10-18

system architecture & machine names, 1-3

**T**

- ttys io command
  - fails on SPARCstation 2 with CG6 frame buffer, 10-14
- tutorial
  - desktop, 3-5
  - OpenWindows, 3-5
- type styles, 1-5
- Type-4 national keyboard compatibility with unbundled products, 7-15

**U**

- unbundled products
  - FORTRAN 1.2 compatibility, 7-15
  - installation problem for client of server running different release, 7-1
- uucp password file problem at bootup, 10-8

**W**

- warnings, 1-6
- window system
  - NeWS, 3-1
  - OpenWindows, 3-1, 7-3
  - X11/NeWS, 3-1

**X**

- X11 window system, 3-1
- X11/NeWS window system, 3-1
- XView Toolkit, 3-1

**Y**

- ypinit command
  - generates error message on slave server, 10-6





---

## Notes

---

# Notes

---

## Notes

---

# Notes

---

## Notes

---

# Notes

---

## Notes



---

# Notes