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.

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

Part No: 800-6643-10
Revision A of December 1991

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.

9

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 nonexistent 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: registerss 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 `biod`(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
%
```

10

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)

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.

add_services(8) Requires 350KB in /usr (1032894)

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 reenter the same clients.

`add_client` Does Not Set Up Multiple Hostnames for Multiple Ethernets (1017238)

A server with multiple ethernets 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 ethernets 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 ethernets on a server in a client's `/export/root/`*client*`/etc/hosts` file. You must manually enter the hostnames of any additional ethernets.

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 'ctlr_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 workaround 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 ethernets 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 `tftpd`, 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`:

```
% su
Password: [root password]
# kill 153
# /usr/etc/inetd
```

ypinit on Slave Server Generates Error Message (1029284)

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 SPARCsystems 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 tranceiver 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] -Qoption 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 sd1
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 ttya 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 xxxxxxxx != 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

   L1/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: ctlr 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_fddi0 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 tranport-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: 0xlebc9c (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 Version3

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*

Package and Content	Disk Space Required
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*

Cluster and Content	Script Requests	Space Used After Install
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 `termsw`, 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 olwm 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 olwm 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(flsbuf.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 otypes 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/tooltalk/. 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>(<mode1> <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. *Canvass* 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