READ THIS FIRST
SunOS™ Release 4.1.1

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

Revision A of 30 October 1990, Part Number 800-5566-10

Read This First

READ THIS FIRST
SunOS Release 4.1.1

This *READ THIS FIRST* (RTF) provides information not available in earlier SunOS 4.1.1 documentation. There are three main sections. The first describes problems with SunOS 4.1.1 release software that were identified after the SunOS 4.1.1 Release Manual went into production. The second section describes problems with OpenWindows and goes over recommended procedures for installing and using OpenWindows. The third section is an OpenWindows Tutorial.

After reading this RTF, you should turn to the *SunOS 4.1.1 Release Manual* before attempting to run or install SunOS 4.1.1.

Getting Help

If you have problems installing or using SunOS 4.1.1, call Sun Microsystems with the information outlined below. In the United States you can call 1-800-USA-4-SUN; outside the U.S., contact your local Sun Answer Center or your Sun sales representative for assistance.

You Will Need to Provide the Following Information:

- Your name and electronic mail address (if any)
- Your company name, address, and phone number
- The model and serial number of your workstation
- Additional information provided by the `showrev`(8) command (described in Chapter 3, Section 3.11 of the *SunOS 4.1.1 Release Manual*):

  – your system's hostname
  – hostid
  – kernel and application architecture
  – kernel revision
  – SunOS release number

- Any information that may help to diagnose the problem.

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

SunOS Release 4.1.1

This section goes over problems with SunOS 4.1.1. When available, bug ID's are given in parentheses after headings. The ID's can be used to reference problems if you need to contact a Sun Answer Center or sales representative.

"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.

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`:

- 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.

- Become superuser, terminate the process (`153` in the example), and restart `inetd`:

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

`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","");
```

GS Systems Require 4.1.1–GFX Rev.1

Systems configured with the GS graphics accelerator/frame buffer board require the 4.1.1-GFX Rev.1 software supplement to SunOS 4.1.1. The 4.1.1-GFX Rev.1 tape and documentation are included with every GS system or upgrade kit. The tape should be installed as the final step in setting up your system. Consult the 4.1.1-GFX Rev.1 documentation for further information.

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.

Misprint in Instructions for Loading and Booting the Miniroot from Remote CD–ROM

There is a misprint in the second box on page 195 of *Installing the SunOS*. The last line in the box should not show a "/" before `miniroot_`*k–arch*. The line should appear as follows:

```
# cp miniroot_k-arch /export/exec/kvm/k-arch.sunos.release/miniroot
```

Sun Database Excelerator (DBE) 1.0 Not Supported under SunOS 4.1.1

Do not use DBE 1.0 with SunOS 4.1.1; it is not supported.

Sun Cross Compilers 3.0 Not Supported under SunOS 4.1.1

Do not use Sun Cross Compilers 3.0 with SunOS 4.1.1; they are not supported.

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
#
```

The root prompt (#) is now displayed, and you are ready to proceed to other matters.

C++ Header Files Must be Modified for Use under SunOS 4.1.1

Eight C++ header files under the `incl` directory of the C++ 2.0 patch need to be modified in order to be fully compatible with SunOS 4.1.1. Three of the files require important changes that are described in detail, below. The remaining files need to have new macro definitions added, and also, in one case, a new struct definition.

Changes to be Made in `incl/mon/openprom.h`

Change lines 167 to 195 for `struct sunromvec` in the file `incl/mon/openprom.h` of your C++ patch directory to appear as shown below (changes are in boldface):

```
    void        (*op_chain)(caddr_t virt, u_int size, caddr_t entry,
                            caddr_t argaddr, u_int arglen);
    void        (*op_release)(caddr_t virt, u_int size);
#else
    phandle_t   (*op_phandle)(/* ihandle_t ihandle */);
                                        /* Convert ihandle to phandle */

    caddr_t     (*op_alloc)(/* caddr_t virthint, u_int size */);
                                        /* Allocate physical memory */

    void        (*op_free)(/* caddr_t virt, u_int size */);
                                        /* Deallocate physical memory */

    caddr_t     (*op_map)(/* caddr_t virthint, u_int space, u_int phys, u_int size */);
                                        /* Create device mapping */

    void        (*op_unmap)(/* caddr_t virt, u_int size */);
                                        /* Destroy device mapping */


    ihandle_t   (*op_open)  (/* char *name */);
    u_int       (*op_close)(/* ihandle_t fileid */);
    int         (*op_read)  (/* ihandle_t fileid, caddr_t buf, u_int len */);
    int         (*op_write)(/* ihandle_t fileid, caddr_t buf, u_int len */);
    int         (*op_seek)  (/* ihandle_t fileid, u_int offsh, u_int offsl */);

    void        (*op_chain)(/* caddr_t virt, u_int size, caddr_t entry,
                            caddr_t argaddr, u_int arglen */);
    void        (*op_release)(/* caddr_t virt, u_int size */);
#endif
    int                 *v_reserved[15];
```

Change to be Made in `incl/pixrect/cg8var.h`

Add `int real_windowfd;` as the last field for `struct cg8_data`, as shown below:

```
struct cg8_data
{
        struct  mprp_data       mprp;           /* memory pixrect simulator */
        int                     flags;          /* misc. flags */
        int                     planes;         /* current group and mask */
        int                     fd;             /* file descriptor */
        short                   active;         /* active no. */
        struct  cg4fb           [CG8_NFBS];     /* frame buffer info */
        int                     windowfd;       /* if 8-bit indexed pw */
        struct  colormapseg     cms;            /* if 8-bit indexed pr */
        int                     real_windowfd;  /* if 8-bit indexed pw */
};
```

Change to be Made in `incl/sundev/scsi.h`

Add `struct scsi_unit *c_drainun;` as the fifth field of `struct scsi_ctlr`, as shown below:

```
struct scsi_ctlr {
        int     c_flags;                /* misc state flags */
        int     c_reg;                  /* controller registers in I/O space */
        int     c_intpri;               /* controller interrupt priority */
        struct  scsi_unit *c_un;        /* scsi unit using the bus */
        struct  scsi_unit *c_drainun;   /* scsi unit being drained */
        struct  scsi_ctlr_subr *c_ss;   /* scsi device subroutines */
```

Changes to be Made in Additional C++ 2.0 Header Files

New macro definitions should be added to the following header files:

```
incl/pixrect/cg12_var.h
incl/pixrect/gplcmds.h
incl/pixrect/pr_dblbuf.h
incl/pixrect/pr_planegroups.h
incl/sun/autoconf.h
```

In addition, `struct dev_path_ops` should be added to `incl/sun/autoconf.h`. To make the changes, carry out the following steps:

- `diff` each header file in the above list with the corresponding file in `/usr/include`. For example:

```
%diff [C++_patch_directory]/incl/pixrect/cg12_var.h \
/usr/include/pixrect/cg12_var.h > header.diff
```

- Examine the differences and copy new macro definitions (plus the new `struct`) to the appropriate header files.

OPENWINDOWS VERSION 2

This section recommends procedures for installing and using OpenWindows and goes over known problems with OpenWindows Version 2.

Reporting Bugs

Please inform us of any bugs you find in the operation of OpenWindows. You can do this by filling out a bug report and sending it to us, as described below. If you need more immediate assistance, you can contact us directly, as described under "Getting Help" at the beginning of this RTF.

To help us process bug reports as quickly and smoothly as possible, please:

1. Use the template bug form in `/usr/openwin/share/bugform`. If you are uncertain about the entry for bug category, you can leave that part of the template blank, or put in a tentative category, which we will review in any event.
2. Send bugs via electronic mail to `sun.com!sunbugs`, not the `windowsbug` alias given in the template. (Customers with software support contracts can also send email to `sun!hotline` or call Sun Microsystems directly.)

Desktop SPARCsystems with Two 104MB Disks: Using Your Second Disk for OpenWindows

If you use the first 104MB disk of a Desktop SPARCsystem as the system disk for SunOS 4.1.1 release software, there will not be enough room for OpenWindows. This section describes how to use the `g` partition of your second disk (`sd1g`) for OpenWindows.

The SunOS 4.1.1 Release Manual describes how to use `sd1g` as `/home`. Following the instructions given here or in the Release Manual, you cannot use `sd1g` for both `/home` and OpenWindows, and must chose between one use or the other. If you put OpenWindows in `sd1g`, you will have about 21MB of space in `/usr` on your first disk for use by `/home`. Experienced users who want to have more options in using their second disk should refer to the *System and Network Administration* manual, which provides detailed information on disk partitioning, mounting partitions, and using symbolic links.

The procedures you need to follow in using `sd1g` for OpenWindows depend on whether or not you use SunInstall to install SunOS 4.1.1 release software. If you use SunInstall, you can set up a `/usr/openwin` filesystem in `sd1g` from the start (See *Installing the SunOS* for instructions on setting up file systems on a second disk). After using SunInstall you can load OpenWindows in `sd1g` by following the instructions given in the section "Installing the OpenWindows Software in `sd1g`", below.*

If you have the preinstalled version of SunOS 4.1.1 on your first disk or you use re-preinstall or Quick Install, you will need to create a `/usr/openwin` filesystem in `sd1g` and carry out additional steps before your can load OpenWindows into `sd1g`.

Systems with SunOS 4.1.1 through Preinstallation, Quick Install, or Re-preinstall

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

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

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

```
%su
Password: [enter root password]
#newfs /dev/rsd1g
/dev/rsd1g: 160230 sectors in 763 cylinders of 6 tracks, 35 sectors
    82.0MB in 48 cyl groups (16 c/g, 1.72MB/g, 768 i/g)
super-block backups (for fsck -b #) at:
 32, 3440, 6848, 10256, 13664, 17072, 20480, 23888, 26912,
 30320, 33728, 37136, 40544, 43952, 47360, 50768, 53792, 57200,
 60608, 64016, 67424, 70832, 74240, 77648, 80672, 84080, 87488,
 90896, 94304, 97712, 101120, 104528, 107552, 110960, 114368, 117776,
 121184, 124592, 128000, 131408, 134432, 137840, 141248, 144656, 148064,
 151472, 154880, 158288,
```

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

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

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

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

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

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

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

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

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

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

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

7. Mount `sd1g` on `/usr/openwin`:

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

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

Installing the OpenWindows Software in `sd1g`

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

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

– Replace *devicenumber* with the tape device number, if you are installing from tape, or with `sr0` if you are installing from CD–ROM.

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

OpenWindows Calendar Manager May Use Incorrect Time Zone When First Run (1045264)

If you are planning to use OpenWindows and you did not install OpenWindows using SunInstall's "Custom installation", the first time you boot your system you should log in as root and run `tzsetup`. You can then exit, and log in under your user name:

* In addition to the man page for `extract_files`, see Chapter 3, Section 3.6, of the SunOS 4.1.1 Release Manual for information on using `extract_files`.

```
login: root
Password: [enter root password]
#/usr/etc/tzsetup
#exit
login: [enter username]
```

`tzsetup` is necessary for giving the OpenWindows Calendar Manager the correct time zone. Without it, the Calendar Manager assumes Greenwich Mean Time. Unless you carry out a SunInstall "Custom installation" to load OpenWindows, `tzsetup` will not run the first time your system boots, and you need to run it manually before using the Calendar manager. `tzsetup` runs automatically whenever your system reboots.

`.xinitrc` File in `/home` Must Be Removed or Edited

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.

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 4.1.1, 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.

`app-defaults` Directory in `/usr/openwin/lib/app-defaults` Must Be Moved

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.

LED's on Type–4 Keyboards Flash When Cursor Moves Across SunView Programs Running under OpenWindows

On Type–4 keyboards, all four keyboard LEDs (the green lights on the upper–right part of the keyboard) flash briefly whenever the cursor moves into or out of a SunView program that is running in an OpenWindows window.

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 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 sub–windows smaller than the top of the bottommost split causes the program that owns the text sub– window 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 Textedit, 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.

Information on Keyboard Map File for Japanese International Keyboard

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

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 filesystem 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.

Use SunView for Maximum Security Levels

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

Application Programs Can Move Windows in Front of the Screen Lock on Monochrome Monitors

On monochrome monitors, application programs can move new or updated windows in front of the Screen Lock. 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.

OpenWindows Supports 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 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 [`*`LTWH`*`]`

  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 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
```

Using `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);
```

Three Methods for Creating XView Fonts

XView fonts can be created by the following methods:

- Creation by specifying the font name:

```
font = xv_create(frame, FONT,
        FONT_NAME, "lucidatypewriter-bold-14",
                NULL);
```

- Creation by resizing a previously created font:

```
font = xv_create(frame, FONT,
        FONT_RESCALE_OF, other_font, WIN_SCALE_LARGE,
                NULL);
```

- Creation by specifying family, style, and scale:

```
font = xv_create(frame, FONT,
            FONT_FAMILY, FONT_FAMILY_LUCIDA,
            FONT_STYLE, FONT_STYLE_BOLD,
            FONT_SIZE, 10,
                        NULL);
```

The third method, font creation by specify family, style, and scale, is the only method that always guarantees correct information about `FONT_SIZE`, `FONT_STYLE`, and `FONT_FAMILY`. However, there are circumstances in which the other two methods will reliably return correct values:

- Creating by resizing will return correct values if the original font was created by specifying family/style/scale.
- If the font name is specified in the X Logical Font Description (XLFD) format, XView will decrypt the font name to extract the correct `FONT_SIZE`, `FONT_STYLE`, and `FONT_VALUE` values.
- Some other well known names, such as *lucida–12*, *lucidatypewriter–bold–12*, and *cour.r.18*, are also decrypted correctly, but is only for compatibility purposes.

In any event, one should always check the values returned by `xv_get` before using them.

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.

OPENWINDOWS V2 ON–LINE TUTORIAL

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

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

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

There are two ways to view the on–line tutorial:

- With animated demonstrations

and

- Without the animated demonstrations

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

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

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

Accessing the On–Line Tutorial with Animated Demonstrations

To access the on–line tutorial with animated demonstrations:

1. Exit OpenWindows.

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

   If you are using the SunOS 4.1.1 default `.login` file, exiting OpenWindows will log you off the system. In this case, you need to log back in and press Control-C when you see the prompt:

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

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

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

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

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

Returning to OpenWindows

To return to OpenWindows when you finish the on–line tutorial:

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

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

Accessing the On–Line Tutorial without Animated Demonstrations

To access the on–line tutorial without animated demonstrations, from within OpenWindows enter:

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

An alert appears on screen:

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

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

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

Moving Around in the Tutorial

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

Closing the Tutorial

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

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

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

Quitting the On–Line Tutorial

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

Training Courses

Sun Educational Services offers advanced training on OpenWindows as well as other Sun hardware and software, system administration, and end–user training. You can call us or send electronic mail to request information or a catalog of our courses and class schedules.

Customers who have a large number of employees to train may wish to arrange for a course to be taught on–site. Courses are offered in all parts of the world—check with your local Sun sales office for more information and complete details. We respond by phone, electronic mail, or fax machine.

- For more information on Sun training courses at a training center in the United States:

  Sun U.S. Training Courses
  1–800–422–8020

- If you are calling from outside the United States:

  Call 1–408–276–3630

- For more information on Sun training courses at a training center in Canada:

  Sun Canadian Training Courses
  1–416–477–6745, Extension 2723