

# Direct Xlib User's Guide

2550 Garcia Avenue  
Mountain View, CA 94043  
U.S.A.



*SunSoft*  
A Sun Microsystems, Inc. Business

© 1995 Sun Microsystems, Inc. 2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A.

All rights reserved. This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Portions of this product may be derived from the UNIX<sup>®</sup> system, licensed from UNIX Systems Laboratories, Inc., a wholly owned subsidiary of Novell, Inc., and from the Berkeley 4.3 BSD system, licensed from the University of California. Third-party software, including font technology in this product, is protected by copyright and licensed from Sun's Suppliers.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

#### TRADEMARKS

Sun, Sun Microsystems, the Sun logo, SunSoft, the SunSoft logo, Solaris, SunOS, OpenWindows, Direct Xlib, DeskSet, ONC, ONC+, and NFS are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd. OPEN LOOK is a registered trademark of Novell, Inc. PostScript and Display PostScript are trademarks of Adobe Systems, Inc.

All SPARC trademarks are trademarks or registered trademarks of SPARC International, Inc. in the United States and other countries. SPARCcenter, SPARCcluster, SPARCcompiler, SPARCdesign, SPARC811, SPARCengine, SPARCprinter, SPARCserver, SPARCstation, SPARCstorage, SPARCworks, microSPARC, microSPARC-II, and UltraSPARC are licensed exclusively to Sun Microsystems, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK<sup>®</sup> and Sun<sup>™</sup> Graphical User Interfaces were developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUI's and otherwise comply with Sun's written license agreements.

X Window System is a trademark of X Consortium, Inc.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN, THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAMS(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.



# *Contents*

---

Preface . . . . .	v
<b>1. Introduction . . . . .</b>	<b>1</b>
Overview . . . . .	1
System Requirements . . . . .	2
Supported Graphics Primitives . . . . .	3
<b>2. Installing the Direct Xlib Software . . . . .</b>	<b>5</b>
File Locations . . . . .	5
Installation Overview . . . . .	6
Determining If Volume Management Is Running . . . . .	6
Installing Direct Xlib Using Volume Management . . . . .	7
Local Installation . . . . .	7
Remote Installation . . . . .	9
Installing Direct Xlib without Volume Management . . . . .	11
Local Installation . . . . .	11
Remote Installation . . . . .	13

---

<b>3. Using the Direct Xlib Software</b> . . . . .	<b>17</b>
Getting Started . . . . .	17
General Setup . . . . .	18
Invoking Direct Xlib 3.2 . . . . .	18
Using Direct Xlib on Your Entire Desktop . . . . .	20
Usage Notes . . . . .	21
Server Connections . . . . .	21
Frame Buffers . . . . .	22
Font Performance . . . . .	22
Determining How Applications Have Been Linked. . . . .	22
dxlib Script. . . . .	23
Developer Notes . . . . .	25
General Application Compiling with Direct Xlib . . . . .	25
Direct Xlib Errors . . . . .	26
Supporting Your Own X11 Extensions with Direct Xlib . . . . .	26
Setting Environment Variables . . . . .	27
Direct Xlib Demonstration Programs . . . . .	27
pointfish . . . . .	28
DXspeed . . . . .	28
xspheres . . . . .	28
Index . . . . .	31

## *Preface*

---

The *Direct Xlib User's Guide* describes how to install and use Direct Xlib on your Sun system.

This document is intended for end-users who are responsible for installing and using Direct Xlib on their Sun system.

---

## What Typographic Changes Mean

The following table describes the typographic changes used in this book.

Table P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name%</code> You have mail.
<b>AaBbCc123</b>	What you type, contrasted with on-screen computer output	<code>machine_name% su</code> Password:
<i>AaBbCc123</i>	Command-line placeholder: replace with a real name or value	To delete a file, type <code>rm filename</code> .
<b><i>AaBbCc123</i></b>	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be root to do this.

## Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

Table P-2 Shell Prompts

Shell	Prompt
C shell prompt	<code>machine_name%</code>
C shell superuser prompt	<code>machine_name#</code>
Bourne shell and Korn shell prompt	<code>\$</code>
Bourne shell and Korn shell superuser prompt	<code>#</code>

# Introduction

---



## Overview

Direct Xlib™ is Sun's accelerated implementation of the Xlib Application Programmers Interface (API). It improves rendering performance for graphics-intensive Xlib applications drawing on the local host's display.

Direct Xlib improves Xlib application performance by eliminating the setup, transport, context switch and parsing overhead involved in communicating graphics commands from the Xlib client to the X11 server. Direct Xlib allows Xlib applications to perform graphics operations directly to the graphics device. Direct Xlib uses the services provided by Direct Graphics Access (DGA), available in OpenWindows™, to synchronize rendering operations with the server and other direct access applications (like those using XGL or XIL). Since this improved performance comes from the elimination of this communication overhead, performance is dependent upon the ratio of overhead to rendering time.

Direct Xlib provides the most improvement when rendering small individual primitives or a large list of batched primitives. In the former case, the performance benefits are due to the elimination of the communication time which is significant when compared to the rendering time for a single small graphics primitive. In the latter case, the performance benefits are due to the elimination of the data copy required to transfer the client's primitive data between the client's address space and the server's.

The applications most likely to benefit from Direct Xlib are those in the following industry segments: mechanical design, electronic design, drafting, mapping, and scientific visualization.

You can use the Direct Xlib software with any existing or newly developed application that is already linked to Xlib as a shared library. Recompiling or relinking is not necessary.

---

**Note** – Direct Xlib does not support statically linked Xlib applications or applications that load `libX11` by calling `dlopen`.

---

Direct Xlib supports the X11R5 API with level 4 Internationalization. The package name for Direct Xlib is `SUNWdxlib`.

## *System Requirements*

The Direct Xlib 3.2 software requires the following:

- Sun-4™ or SPARC® systems
- SunOS™ 5.5 software
- OpenWindows Version 3.5 software



---

## *Supported Graphics Primitives*

The following primitives are accelerated by the Direct Xlib software:

XCopyArea and XCopyPlane  
XGetImage and XPutImage  
XSetClipRectangles and XSetClipMask  
XDrawLine and XDrawLines  
XDrawSegments  
XDrawPoint and XDrawPoints  
XFillRectangle and XFillRectangles  
XDrawString and XDrawImageString  
XDrawString16 and XDrawImageString16  
XDrawText and XDrawText16  
XmbDrawString and XmbDrawImageString  
XwcDrawString and XwcDrawImageString  
XmbDrawText and XwcDrawText  
XFillPolygon  
XDrawRectangle and XDrawRectangles  
XClearWindow and XClearArea  
XDrawArc and XDrawArcs  
XFillArc and XFillArcs

All other Xlib operations are supported by the Direct Xlib software through standard Xlib server rendering. Direct Xlib supports shared fonts, thereby minimizing wasted memory.



## *Installing the Direct Xlib Software*

---



This chapter shows you how to install the Direct Xlib software. The supported system configurations include a SPARC-based standalone or server workstation. The installation procedures include installing Direct Xlib on systems with Volume Management installed, as well as for systems without Volume Management.

The Direct Xlib 3.2 package requires about four megabytes of disk space. Make sure you have enough disk space before you attempt to install Direct Xlib.

Install Direct Xlib using `pkgadd` after you have successfully installed Solaris 2.5 software.

### *File Locations*

The installation script installs the specific product's subdirectories in a standard directory (`/opt`). You can change the standard directory by providing a new location at the appropriate prompt. The subdirectories are then installed under the specified location.

## *Installation Overview*

The following is an overview of the basic installation steps:

1. Determine if your system is running Volume Management.
2. Mount the CD-ROM.
3. Install Direct Xlib.
4. Eject the CD-ROM.

## *Determining If Volume Management Is Running*

You must first determine if Volume Management is running on the system with the CD-ROM drive attached. If Volume Management is running, the system automatically mounts the CD-ROM when you insert it into the CD-ROM drive. If the system is not running Volume Management, then you must manually mount the CD-ROM.

To determine if Volume Management is running,

- 1. Log in as `root` on the system with the CD-ROM drive attached.**
- 2. Enter the following to run the `grep` program for locating Volume Management, and press Return:**

```
hostname% /usr/bin/ps -e | grep vold
```

If you find `vold`, then Volume Management is running on your system. Proceed to "Installing Direct Xlib Using Volume Management" on page 7 in this chapter.

If you do *not* find `vold`, then Volume Management is not running on your system. Proceed to "Installing Direct Xlib without Volume Management" on page 11 in this chapter.

## Installing Direct Xlib Using Volume Management

This procedure describes how to install Direct Xlib on a system using Volume Management. You can install Direct Xlib on either your *local* workstation or on your system from a *remote* workstation.

### Local Installation

To install Direct Xlib on your local system,

1. **Place the Solaris 2.5 CD-ROM disc into its caddy and insert the caddy into the drive.**



**Caution** – The CD drive is not the same as a CD player for audio discs that you may be accustomed to from home. Never put the CD-ROM into the drive without **first** placing it in the caddy. When you place the CD-ROM in the caddy, make sure that it fits securely under the caddy-tray lip before you insert the caddy into the drive.

2. **Log in as `root`. Enter the following and press Return after each entry:**

```
hostname% su root
Password: root-password
```

3. **Verify that you have mounted the CD-ROM. To do this, enter the following and press Return:**

```
hostname# mount
```

A line similar to the following should appear after you enter `mount`.

```
/cdrom/solaris_2_5/s0 on /vol/dev/dsk/ct06/solaris_2_5/s0 read only
Thur Mar 4 19:16:48 1995
```

4. Execute `pkgadd` to install the Direct Xlib software. Enter the following and press Return:

```
hostname# /usr/sbin/pkgadd -a none -d /cdrom/solaris_2_5/s0/Solaris_2.5
```

5. Follow the instructions that then appear on the screen.

When asked to select a package to add, select the `SUNWdxlib` package.

---

**Note** - To remove `SUNWdxlib` at any time, use `pkgrm`.

---

6. Specify the default installation path, `/opt`, or any path that contains enough disk space for the installation.

Direct Xlib 3.2 installation requires approximately four megabytes of space.

7. Eject the CD-ROM to unmount and retrieve the CD-ROM. To do this, enter the following and press Return:

```
hostname# eject sr0
```

8. Log out.

Installation is now complete.

## Remote Installation

To install Direct Xlib from a CD-ROM attached to a remote system,

1. **On the remote system, place the Solaris 2.5 CD-ROM disc into its caddy and insert the caddy into the drive.**



**Caution** – The CD drive is not the same as a CD player for audio discs that you may be accustomed to from home. Never put the CD-ROM into the drive without **first** placing it in the caddy. When you place the CD-ROM in the caddy, make sure that it fits securely under the caddy-tray lip before you insert the caddy into the drive.

2. **Log in to the remote system as `root`. Enter the following and press Return after each entry:**

```
hostname% su root
Password: root-password
```

3. **Export the CD-ROM filesystem. To do this, enter the following and press Return:**

```
hostname# share -F nfs -o ro=localhost /cdrom/solaris_2_5/s0
```

4. **Log out of the remote system.**

5. **Log in to the local system as `root`, create `/cdrom`, and mount the CD-ROM. To do this, enter the following on the local system. Press Return after each entry:**

```
hostname% su root
Password: root-password
hostname# mkdir /cdrom
hostname# mount -r remote_machinename:/cdrom/solaris_2_5/s0 /cdrom
```

- 6. Verify that you have mounted the CD-ROM. To do this, enter the following and press Return:**

```
hostname# mount
```

A line similar to the following should appear after you enter `mount`:

```
/cdrom on syaubi:/cdrom/solaris_2_5/s0 read only/remote  
Thur Mar 4 19:16:48 1995
```

- 7. Execute `pkgadd` to install the Direct Xlib software. Enter the following and press Return:**

```
hostname# /usr/sbin/pkgadd -a none -d /cdrom/Solaris_2.5
```

- 8. Follow the instructions that then appear on the screen.**

When asked to select a package to add, select the `SUNWdxlib` package.

---

**Note** - To remove `SUNWdxlib` at any time, use `pkgrm`.

---

- 9. Specify the default installation path, `/opt`, or any path that contains enough disk space for the installation.**

Direct Xlib 3.2 installation requires approximately four megabytes of space.

- 10. Unmount the CD-ROM. To do this, enter the following and press Return:**

```
hostname# umount /cdrom
```

- 11. Log out of the local system.**

- 12. Log in to the remote system as `root`.**



- 13. Eject the CD-ROM to unmount and retrieve the CD-ROM. To do this, enter the following on the remote system, and press Return:**

```
hostname# eject sr0
```

- 14. Log out.**  
Installation is now complete.

## *Installing Direct Xlib without Volume Management*

This procedure describes how to install Direct Xlib on a system that does not use Volume Management. You can install Direct Xlib on either your *local* workstation or on your system from a *remote* workstation.

### *Local Installation*

To install Direct Xlib on your local system,

- 1. Place the Solaris 2.5 CD-ROM disc into its caddy and insert the caddy into the drive.**



**Caution** – The CD drive is not the same as a CD player for audio discs that you may be accustomed to from home. Never put the CD-ROM into the drive without **first** placing it in the caddy. When you place the CD-ROM in the caddy, make sure that it fits securely under the lip before you insert the caddy into the drive.

- 2. Log in as `root`, create the directory `/cdrom`, and mount the CD-ROM. To do this, enter the following and press Return after each entry:**

```
hostname% su root
Password: root-password
hostname# mkdir /cdrom
hostname# mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /cdrom
```

- 3. Verify that you have mounted the CD-ROM. To do this, enter the following and press Return:**

```
hostname# mount
```

A line similar to the following should appear after you enter `mount`:

```
/cdrom on /dev/dsk/c0t6d0s0 read only Thur Mar 4 19:16:48 1995
```

- 4. Execute `pkgadd` to install the Direct Xlib software. To do this, enter the following and press Return:**

```
hostname# /usr/sbin/pkgadd -a none -d /cdrom/Solaris_2.5
```

- 5. Follow the instructions that then appear on the screen.**

When asked to select a package to add, select the `SUNWdxlib` package.

---

**Note** - To remove `SUNWdxlib` at any time, use `pkgrm`.

---

- 6. Specify the default installation path, `/opt`, or any path that contains enough disk space for the installation.**

Direct Xlib 3.2 installation requires approximately four megabytes of space.

- 7. Unmount the CD-ROM. To do this, enter the following and press Return:**

```
hostname# umount /cdrom
```

- 8. Eject and retrieve the CD-ROM. To do this, enter the following and press Return:**

```
hostname# eject sr0
```

**9. Log out.**

Installation is now complete.

## *Remote Installation*

To install Direct Xlib from a CD-ROM attached to a remote system,

- 1. On the remote system, place the Solaris 2.5 CD-ROM disc into its caddy and insert the caddy into the drive.**



**Caution** – The CD drive is not the same as a CD player for audio discs that you may be accustomed to from home. Never put the CD-ROM into the drive without **first** placing it in the caddy. When you place the CD-ROM in the caddy, make sure that it fits securely under the lip before you insert the caddy into the drive.

- 2. Log in as `root`, create the directory `/cdrom`, and mount the CD-ROM. To do this, enter the following and press Return after each entry:**

```
hostname% su root
Password: root-password
hostname# mkdir /cdrom
hostname# mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /cdrom
```

- 3. Export the CD-ROM filesystem. To do this, enter the following and press Return:**

```
hostname# share -F nfs -o ro=localhost /cdrom
```

- 4. Log out of the remote system.**

- 5. Log in on the local system as `root`, create `/cdrom`, and mount the CD-ROM. To do this, enter the following on the local system and press Return after each entry:**

```
hostname% su root
Password: root-password
hostname# mkdir /cdrom
hostname# mount -r remote_machinename:/cdrom /cdrom
```

- 6. Verify that the CD-ROM has been mounted. To do this, enter the following, and press Return:**

```
hostname# mount
```

A line similar to the following should appear after you enter `mount`:

```
/cdrom on syaubi:/cdrom read only/remote Thur Mar 4 19:16:48 1995
```

- 7. Execute `pkgadd` to install the Direct Xlib software. To do this, enter the following and press Return:**

```
hostname# /usr/sbin/pkgadd -a none -d /cdrom/Solaris_2.5
```

- 8. Follow the instructions that then appear on the screen.**

When asked to select a package to add, select the `SUNWdxlib` package.

---

**Note** - To remove `SUNWdxlib` at any time, use `pkgrm`.

---

- 9. Specify the default installation path, `/opt`, or any path that contains enough disk space for the installation.**

Direct Xlib 3.2 installation requires approximately four megabytes of space.

**10. Unmount the CD-ROM. To do this, enter the following and press Return:**

```
hostname# umount /cdrom
```

**11. Log out of the local system.**

**12. Log in to the remote system as `root`.**

**13. Unmount and eject the CD-ROM to retrieve the CD-ROM. To do this, enter the following on the remote system and press Return after each entry:**

```
hostname# umount /cdrom  
hostname# eject sr0
```

**14. Log out.**

Installation is now complete.



## *Using the Direct Xlib Software*

---

3 

This chapter shows you how to use the Direct Xlib software with your individual applications, and with your entire desktop. Usage Notes provide additional information about using Direct Xlib 3.2.

Also, Developer Notes are included to provide additional information about the Direct Xlib software for developing applications or for writing your own X extension libraries.

### *Getting Started*

This section describes the basic instructions for using Direct Xlib on an application-by-application basis. These instructions assume that your application is explicitly dynamically linked to `libX11`. If your application is statically linked to `libX11`, loads `libX11` by invoking `dlopen`, or if you are not sure, see "Determining How Applications Have Been Linked" on page 22.

## General Setup

By default, Direct Xlib places shared memory files in `/tmp`. Therefore, you may need more swap space than normally required when using Direct Xlib. This is especially true if you use a large number of 16-bit fonts.

You can set a different path for Direct Xlib to use for shared memory with the following command when starting OpenWindows. To do this, replace `/home/mymachine` with any path with plenty of disk space where you have write permission:

```
hostname% openwin -sharedretainedpath /home/mymachine &
```

It is only necessary to use the `-sharedretainedpath` option with Direct Xlib if you have limited space in `/tmp`.

If you are upgrading from Direct Xlib 2.0 and used the `-sharedretainedpath` path option, you should remove all files in the specified directory. The shared font mechanism has changed between Direct Xlib 2.0 and 3.x. Old shared font files created through use of Direct Xlib 2.0 will not work correctly with Direct Xlib 3.2

## Invoking Direct Xlib 3.2

In this section, `/mydir` refers to the directory where you installed Direct Xlib. With the standard installation, this directory is `/opt`. Substitute the directory path on your machine that points to the location of Direct Xlib.

- 1. Make sure you are running OpenWindows Version 3.5 or later software.**
- 2. In a Shell Tool or Command Tool, enter the following to start a single application using Direct Xlib.**

```
hostname% /mydir/SUNWdxlib/bin/dxlib myapp
```

where `myapp` is the name of the application you want to start. Follow the application name with the command-line arguments, if any, you would normally use when starting that application.



---

Remember that when you activate Direct Xlib in this manner, it is only active for the single application started with the `dxlib` script. You must execute the `dxlib` script with each application you want to use with Direct Xlib. If you want your entire desktop to use Direct Xlib, see "Using Direct Xlib on Your Entire Desktop" on page 20.

---

**Note** - The `LD_LIBRARY_PATH` environment variable should already be set to `$OPENWINHOME/lib` before executing the above command where `$OPENWINHOME` is set to `/usr/openwin` under SunOS5.x.

---

You may want to display general information about how to use the Direct Xlib software and its current status. To do this, enter `dxlib` with no application name, as shown in the following example:

---

**Note** - The release number in the following example corresponds to the major MIT release from which this version of the Direct Xlib software was derived. The version number is the version of the Direct Xlib software.

---

```
hostname% /mydir/SUNWdxlib/bin/dxlib

usage: dxlib [application]

Looking for /opt/SUNWdxlib/lib/dxlib/SunOS5.x/libX11.so.4

Direct Xlib Version 3.2 Release 5 is active.

For full desktop use, add the following two lines to your .xinitrc
just before the window manager startup line
(e.g. just before "olwm -syncpid $pid & olwmpid=$!"):

LD_LIBRARY_PATH=/opt/SUNWdxlib/lib/dxlib/SunOS5.x:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH

See the Direct Xlib User's Guide for more information.
```

If your environment is not correctly set up to use Direct Xlib when you use the `dxlib` script, the software returns an error message. Refer to "Usage Notes" on page 21.

## Using Direct Xlib on Your Entire Desktop

This section describes how to use Direct Xlib on your entire desktop.

1. **Add the following lines in your `.xinitrc` file just before you start your window manager (`o1wm` in the default case):**

---

**Note** – The `LD_LIBRARY_PATH` environment variable should already be set to `$OPENWINHOME/lib` before executing the following command where `$OPENWINHOME` is set to `/usr/openwin` under SunOS5.x.

---

```
LD_LIBRARY_PATH=/mydir/SUNWdxlib/lib/dxlib/SunOS5.x:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

**where `/mydir` is `/opt` in the default case. If no `.xinitrc` file exists in your home directory**

- a. **Copy `$OPENWINHOME/lib/Xinitrc` to `.xinitrc` in your home directory.**

- b. **Edit the file, and add the above lines just before the following line:**

```
if [ "$OW_WINDOW_MANAGER" ]; then #Alternate Window Manager
```

**If `$OPENWINHOME` is not equal to `/usr/openwin`, then modify the `LD_LIBRARY_PATH` to include `$OPENWINHOME/server/lib` as shown below:**

```
LD_LIBRARY_PATH=/mydir/SUNWdxlib/lib/dxlib/SunOS5.x:$OPENWINHOME/server/lib
:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

2. **Start the OpenWindows server as you would normally.**

Now, after starting the OpenWindows server, if you enter

```
env
```

the following appears in front of your old `LD_LIBRARY_PATH`:

```
/mydir/SUNWdxlib/lib/dxlib/SunOS5.x
```

or

```
/mydir/SUNWdxlib/lib/dxlib/SunOS5.x:$OPENWINHOME/server/lib
```

---

Leave your `LD_LIBRARY_PATH` set to this as long as you want Direct Xlib to be automatically used for dynamically linked applications when you start them from shells within the OpenWindows software or from the desktop menu. You can disable Direct Xlib for particular applications by setting the `DIRECTX_DISABLE` environment variable to any non-zero value before starting the application as discussed in "Setting Environment Variables" on page 27.

---

**Note** – DO NOT SET `LD_LIBRARY_PATH` environment variable in your `.cshrc` to include `/mydir/SUNWdxlib/lib/dxlib/SunOS5.x`. If you do, you will not be able to run OpenWindows.

---

If you update your `LD_LIBRARY_PATH` in your `.cshrc`, then Direct Xlib will not be enabled when you run Xlib applications from within a shell. You will have to explicitly run `/opt/SUNWdxlib/bin/dxlib myapp` (your Xlib application), or update your `.cshrc` file and remove references to `LD_LIBRARY_PATH`.

## Usage Notes

This section contains detailed information on using Direct Xlib. Unless you encounter problems when using Direct Xlib, or plan to develop applications using it, you may wish to skip this section and the more detailed sections that follow.

### Server Connections

The following message may appear while running Direct Xlib:

```
Xlib: connection to "display:X.X" refused by server
Xlib: Maximum number of clients reached
```

If this message appears, you need to shutdown some of your Direct Xlib applications before you can again connect to the server. Direct Xlib consumes server connection resources more quickly than standard Xlib.

### *Frame Buffers*

Direct Xlib usage on devices that do not have hardware cursor support is not recommended. The Cursor image data may corrupt underlying window data. These devices include CG3, CG8, and the onboard frame buffer for the SPARCStation 4.

On GX, GXplus, TurboGX, Turbo GXplus, SX, ZX and TurboZX frame buffers, Direct Xlib forces the cursor to fit in the hardware cursor, which is only 32 by 32 pixels. Therefore, when running Direct Xlib on this type of frame buffer, you may notice that drag and drop cursors may not display completely.

### *Font Performance*

Direct Xlib cannot detect fixed width fonts that can normally be accelerated within the X server. Due to this limitation, applications that are dominated by fixed width fonts may have their performance reduced rather than improved through Direct Xlib usage. Applications that mix fixed width fonts with non-fixed width fonts or other operations should not be greatly impacted by this problem.

### *Determining How Applications Have Been Linked*

Applications already dynamically linked to `libX11` automatically start using the Direct Xlib software when it is enabled. An application that is statically linked to `libX11` needs relinking to take advantage of Direct Xlib.

---

**Note** - Direct Xlib does not support statically linked Xlib applications.

---

You can determine whether an application is statically or dynamically linked to libX11. To do this, enter `ldd myapp` at your prompt, and press Return. The following appears:

```
hostname% ldd myapp (where myapp is the name of the application)
libX11.so.4 => /opt/SUNWdxlib/lib/dxlib/SunOS5.x/libX11.so.4
libsocket.so.1 => /usr/lib/libsocket.so.1
libnsl.so.1 => /usr/lib/libnsl.so.1
libc.so.1 => /usr/lib/libc.so.1
libdl.so.1 => /usr/lib/libdl.so.1
libdga.so.1 => /usr/openwin/lib/libdga.so.1
```

If after using `ldd`, you see that `libX11` is one of the libraries listed, then this application is dynamically linked to Xlib. If `libX11` is not listed, then either the application does not use Xlib, calls `dlopen` to load Xlib, or it is statically linked to Xlib.

### *dxlib* Script

If the `dxlib` script is run with no application name and it displays the message "Direct Xlib is not active," the Direct Xlib software has probably not been correctly installed on your system. See Chapter 2, "Installing the Direct Xlib Software," for more details. When Direct Xlib is run on dumb frame buffers, this message states that Direct Xlib is active but will not do direct rendering. It does not perform any direct rendering on these frame buffers, but continues to send X11 requests to the OpenWindows server as standard Xlib does.

For more specific information about the state of Direct Xlib, including the state of the environment variables that you can use to control it (explained in "Setting Environment Variables" on page 27), enter the following:

```
hostname% /mydir/SUNWdxlib/bin/dxlib
/mydir/SUNWdxlib/bin/dxlib_status -v

Looking for /opt/SUNWdxlib/lib/dxlib/SunOS5.x/libX11.so.4

Direct Xlib Version 3.2 Release 5 is active.

    Minimum direct window width is 96.
    Minimum direct window height is 64.
    Direct font memory limit is 65536 bytes.
    Direct retained windows are supported.
    Direct pixmaps are supported.
```

*mydir* refers to the directory where you installed Direct Xlib.

If the `dxlib` script cannot find Direct Xlib, a message to that effect appears. It suggests that you set the `DXLIBHOME` environment variable to the correct location. This usually only happens if the `dxlib` script has been moved after installation.

You can still use the script from this location if you set the `DXLIBHOME` environment variable to the directory path containing Direct Xlib.

To do so on an application-by-application basis, set the `DXLIBHOME` environment variable to where Direct Xlib is located before using the `dxlib` script. For example, if you have installed Direct Xlib in `/home/mydxlib`, enter the following:

```
hostname% setenv DXLIBHOME /home/mydxlib/SUNWdxlib
hostname% dxlib myapp
```

To get status information on Direct Xlib, enter the following:

```
hostname% setenv DXLIBHOME /home/mydxlib/SUNWdxlib
hostname% dxlib $DXLIBHOME/bin/dxlib_status -v
```

## *Developer Notes*

This section covers topics specific to developers using Direct Xlib. Direct Xlib gives your application process direct access to system hardware that you did not previously have when using the X11 API. Because of this, errors in applications that accidentally corrupt system memory can have more severe effects on the system than before. In particular, accidentally writing to graphics accelerator hardware registers can cause system crashes. This is, however, a highly unlikely event for well-behaved applications.

### *General Application Compiling with Direct Xlib*

When compiling X11 applications, continue to use the standard `include/X11` header files. You only need to use the header files supplied with Direct Xlib under the following conditions:

- If you are developing your own X extension libraries
- If your application code directly manipulates the Xlib X11 request buffer pointed to by the Display pointer
- If it modifies pixels in X11 drawables in a manner unknown to Xlib

If any of these conditions exist, you should contact the Sun Microsystems Computer Corporation (SMCC) US Answer Center at (800) 872-4786 for further assistance on developing your code using Direct Xlib.

If you are currently dynamically linked to `libX11`, continue to link to the *standard* `libX11` libraries. Linking dynamically to Direct Xlib causes your executable to reference libraries that may not be available through standard `libX11`. This makes it more difficult to move between standard `libX11` and the Direct Xlib software.

If you are statically linked to `libX11` and want to use Direct Xlib, you will have to dynamically link to `libX11`. Direct Xlib no longer supplies static versions of its libraries.

If your application library both uses and is statically linked to `libXext`, you must dynamically link to `libXext`.

### *Direct Xlib Errors*

Direct Xlib may not report as many X11 error conditions as standard X11. This is especially apparent when applications attempt to send numeric information out of the acceptable range. For this reason, do primary application development with standard `libX11`, and use Direct Xlib after application code is considered to be relatively bug free.

### *Supporting Your Own X11 Extensions with Direct Xlib*

Normal X11 applications do *not* need to perform any of the operations discussed in this section; only developers of applications or X11 extensions that reach below the X11 API or attempt to extend it need to pay attention to this section. If your application directly manipulates the Xlib X11 request buffer or if you are developing your own X extension libraries, you need to adhere to the following rules when using your application or extension with Direct Xlib.

Your application and/or extension must be compiled with the `Xlibint.h` header file from OpenWindows 3.5.

In addition, you need to synchronize your Xlib request buffer manipulation and any pixel modifications performed by your extension but unknown to Xlib with Direct Xlib by using additional macros. For more information on how to develop applications or extensions that reach below the Xlib API, contact Sun Microsystems Computer Corporation (SMCC).



## Setting Environment Variables

The following environment variables control memory usage and allow portions of Direct Xlib to be disabled.

DIRECTX_MIN_WIN_WIDTH	Limits the number of file descriptors by specifying a minimum window size below which the Direct Xlib software will not attempt direct access for that window. They default to 96 and 64 respectively.
DIRECTX_MIN_WIN_HEIGHT	
DIRECTX_FONT_MEMORY_LIMIT	Turns shared fonts on or off.
DIRECTX_DISABLE	Disables Direct Xlib functionality.

Use the following to determine the current values of the above environment variables:

```
/opt/SUNWdxlib/bin/dxlib /opt/SUNWdxlib/bin/dxlib_status -v
```

## Direct Xlib Demonstration Programs

Direct Xlib provides three demonstration programs. `pointfish`, `DXspeed`, and `xospheres` illustrate the performance of Direct Xlib.

These programs are installed in the following subdirectory:

```
/mydir/SUNWdxlib/demo/bin
```

`DXspeed` and `pointfish` depend upon images that are part of the OpenWindows packages. The particular packages, `SUNWoldim` and `SUNWxwdim`, are not installed by default. They are only installed if the entire Solaris 2.5 software distribution is installed. You will have to install this package to use `DXspeed` and `pointfish` to their fullest capabilities.

In the following examples, */mydir* refers to the directory where you installed Direct Xlib. With the standard installation, this directory is `/opt`.

- 1. Make sure the OpenWindows Version 3.5 software is running as described previously.**
- 2. In a Shell Tool or Command Tool, enter the following commands to start the demonstration programs.**

### ***pointfish***

The `pointfish` program uses `XDrawPoints` to draw an aquarium background with many fish. The background and the fish are sorted by color. To run this program, enter the following:

```
hostname% /mydir/SUNWdxlib/bin/dxlib /mydir/SUNWdxlib/demo/bin/pointf
```

The `-t` option to `pointfish` gives a points-per-second measure. To compare the performance, run the `pointfish` demonstration program without the Direct Xlib software.

### ***DXspeed***

The `DXspeed` program demonstrates the performance of the Direct Xlib software using `XPutImage`, `XCopyArea`, `XDrawLines`, `XDrawSegments`, `XFillPolygon`, and `XDrawPoints`. `DXspeed` uses four fish and the mandrill baboon for the image and area tests. For vectors, polygons, and points, `DXspeed` allows you to pick from four different math functions. These math functions generate a 125x125 matrix with a depth associated at each vertex. To run this program, enter the following:

```
hostname% /mydir/SUNWdxlib/bin/dxlib  
/mydir/SUNWdxlib/demo/bin/DXspeed
```

To compare the performance, try running the `DXspeed` demonstration without the Direct Xlib software.

### ***xspheres***

The `xspheres` program demonstrates the performance of the Direct Xlib software using `XPutImage` and `XCopyArea`. To run this program, enter the following:

```
hostname% /mydir/SUNWdxlib/bin/dxlib  
/mydir/SUNWdxlib/demo/bin/xspheres
```

An `xspheres` manpage is available in the `/demo/bin` directory which explains its use. Source code is also available as a compressed `tar` file. Read the `xspheres.man` man page for usage instructions. `xspheres` runs fastest on frame buffers with offscreen pixmap memory.



# Index

---

## A

applications  
  compiling, 25  
  linking, 17  
applications, linked to `libX11`, 22

## C

compiling general applications, 25

## D

demonstration programs, 27  
determining if volume management is  
  running, 6  
Direct Graphics Access (DGA), 1  
Direct Xlib  
  description, 1  
  graphics primitives supported, 3  
  installing on local system  
    with volume management, 7  
    without volume management,  
      11  
  installing on remote system  
    with volume management, 9  
    without volume management,  
      13  
Direct Xlib 3.1

  compiling general applications, 25  
  errors, 26  
  general setup, 18  
  getting started, 17  
  invoking, 18  
  supporting X11 extensions, 26  
  X11 applications, 25  
`DIRECTX_DISABLE`, 21, 27  
`DIRECTX_FONT_MEMORY_LIMIT`, 27  
`DIRECTX_MIN_WIN_HEIGHT`, 27  
`DIRECTX_MIN_WIN_WIDTH`, 27  
`dlopen`, 17, 23  
`dxlib` script, 19, 23  
`DXLIBHOME`, 24  
`DXspeed` program, 27, 28

## E

environment variables  
  `DIRECT_MIN_WIN_WIDTH`, 27  
  `DIRECTX_DISABLE`, 21, 27  
  `DIRECTX_FONT_MEMORY_LIMIT`, 27  
  `DIRECTX_MIN_WIN_HEIGHT`, 27  
  `DXLIBHOME`, 24  
  `LD_LIBRARY_PATH`, 20  
environment variables, setting, 27  
errors, Direct Xlib 3.1, 26

---

## F

file locations, 5  
frame buffers, 22

## G

general setup, Direct Xlib 3.1, 18  
getting started, Direct Xlib 3.1, 17  
graphics primitives supported, 3

## I

installation  
    overview of steps, 6  
installing Direct Xlib, 5  
    using volume management, 7  
    without volume management, 11  
invoking Direct Xlib 3.1, 18

## L

LD\_LIBRARY\_PATH, 19, 21  
ldd myapp command, 23

## N

notes, developers, 25

## O

OpenWindows  
    setting a different path, 18

## P

pkgadd program, 5, 8, 10, 12, 14  
pkgrm program, 8, 10, 12, 14  
pointfish program, 28

## S

server connections, 21  
SUNWdxlib, 2  
    demonstration programs, 27  
supported graphics primitives, 3

system configurations supported, 5  
system requirements, 2

## U

using Direct Xlib 3.1, 17  
using Direct Xlib 3.1 on your entire  
    desktop, 20

## V

volume management  
    determining if on system, 6  
    Direct Xlib installation, 5

## X

X11 drawables, 25  
X11 extensions  
    supporting with Direct Xlib, 26  
X11R5, Direct Xlib support, 2  
xspheres manpage, 29  
xspheres program, 28

Copyright 1995 Sun Microsystems, Inc., 2550 Garcia Avenue, Mountain View, Californie 94043-1100 USA.

Tous droits réservés. Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie et la décompilation. Aucune partie de ce produit ou de sa documentation associée ne peuvent être reproduits sous aucune forme, par quelque moyen que ce soit sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il en a.

Des parties de ce produit pourront être dérivées du système UNIX<sup>®</sup>, licencié par UNIX Systems Laboratories Inc., filiale entièrement détenue par Novell, Inc. ainsi que par le système 4.3. de Berkeley, licencié par l'Université de Californie. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

LEGENDE RELATIVE AUX DROITS RESTREINTS : l'utilisation, la duplication ou la divulgation par l'administration américaine sont soumises aux restrictions visées à l'alinéa (c)(1)(ii) de la clause relative aux droits des données techniques et aux logiciels informatiques du DFAR 252.227- 7013 et FAR 52.227-19.

Le produit décrit dans ce manuel peut être protégé par un ou plusieurs brevet(s) américain(s), étranger(s) ou par des demandes en cours d'enregistrement.

#### MARQUES

Sun, Sun Microsystems, le logo Sun, Solaris SunSoft, the SunSoft logo, Solaris, SunOS, OpenWindows, Direct Xlib, DeskSet, ONC, ONC+, and NFS sont des marques déposées ou enregistrées par Sun Microsystems, Inc. aux Etats-Unis et dans certains autres pays. UNIX est une marque enregistrée aux Etats-Unis et dans d'autres pays, et exclusivement licenciée par X/Open Company Ltd. OPEN LOOK est une marque enregistrée de Novell, Inc., PostScript et Display PostScript sont des marques d'Adobe Systems, Inc.

Toutes les marques SPARC sont des marques déposées ou enregistrées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. SPARCcenter, SPARCcluster, SPARCcompiler, SPARCdesign, SPARC811, SPARCengine, SPARCprinter, SPARCserver, SPARCstation, SPARCstorage, SPARCworks, microSPARC, microSPARC II et UltraSPARC sont exclusivement licenciées à Sun Microsystems, Inc. Les produits portant les marques sont basés sur une architecture développée par Sun Microsystems, Inc.

Les utilisateurs d'interfaces graphiques OPEN LOOK<sup>®</sup> et Sun<sup>™</sup> ont été développés par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique, cette licence couvrant aussi les licences de Sun qui mettent en place OPEN LOOK GUIs et qui en outre se conforment aux licences écrites de Sun.

Le système X Window est un produit du X Consortium, Inc.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" SANS GARANTIE D'AUCUNE SORTE, NI EXPRESSE NI IMPLICITE, Y COMPRIS, ET SANS QUE CETTE LISTE NE SOIT LIMITATIVE, DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DES PRODUITS A REPENDRE A UNE UTILISATION PARTICULIERE OU LE FAIT QU'ILS NE SOIENT PAS CONTREFAISANTS DE PRODUITS DE TIERS.

CETTE PUBLICATION PEUT CONTENIR DES MENTIONS TECHNIQUES ERRONEES OU DES ERREURS TYPOGRAPHIQUES. DES CHANGEMENTS SONT PERIODIQUEMENT APPORTES AUX INFORMATIONS CONTENUES AUX PRESENTES, CES CHANGEMENTS SERONT INCORPORES AUX NOUVELLES EDITIONS DE LA PUBLICATION. SUN MICROSYSTEMS INC. PEUT REALISER DES AMELIORATIONS ET/OU DES CHANGEMENTS DANS LE(S) PRODUIT(S) ET/OU LE(S) PROGRAMME(S) DECRITS DANS CETTE PUBLICATION A TOUS MOMENTS.



Adobe PostScript

