

SGI® OpenGL Vizserver™
User's Guide

Version 3.4.2



007-4245-013



CONTRIBUTORS

Written by Ken Jones and Jenn McGee

Illustrated by Chrystie Danzer

Production by Karen Jacobson

Engineering contributions by Michael Boccara, Phil Keslin, Younghee Lee, Peter Ostrin, Chikai Ohazama, Sonya Rikhtverchik, and Yochai Shefi-Simchon

COPYRIGHT

© 2001–2004 Silicon Graphics, Inc. All rights reserved; provided portions may be copyright in third parties, as indicated elsewhere herein. No permission is granted to copy, distribute, or create derivative works from the contents of this electronic documentation in any manner, in whole or in part, without the prior written permission of Silicon Graphics, Inc.

LIMITED RIGHTS LEGEND

The software described in this document is “commercial computer software” provided with restricted rights (except as to included open/free source) as specified in the FAR 52.227-19 and/or the DFAR 227.7202, or successive sections. Use beyond license provisions is a violation of worldwide intellectual property laws, treaties and conventions. This document is provided with limited rights as defined in 52.227-14.

TRADEMARKS AND ATTRIBUTIONS

Silicon Graphics, SGI, the SGI logo, IRIX, InfiniteReality, Octane, Onyx, Onyx2, OpenGL, and Tezro are registered trademarks and GLX, InfinitePerformance, InfiniteReality2, InfiniteReality3, InfiniteReality4, Octane2, Onyx4, OpenGL Vizserver, SGI ProPack, Silicon Graphics Fuel, Silicon Graphics Prism, and UltimateVision are trademarks of Silicon Graphics, Inc., in the United States and/or other countries worldwide.

Linux is a registered trademark of Linus Torvalds, used with permission by Silicon Graphics, Inc. Red Hat is a registered trademark of Red Hat, Inc. Solaris is a trademark of Sun Microsystems, Inc. UNIX and the X device are registered trademarks of The Open Group in the United States and other countries. Windows and Windows NT are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks mentioned herein are the property of their respective owners.

New Features in This Guide

In addition to miscellaneous corrections, this revision of the guide includes changes to support new OpenGL Vizserver servers. The primary changes are in the following sections:

- “Remote Servers” on page xi
- “Software Platforms” on page xii

Record of Revision

Version	Description
002	March 2001 Updated to support OpenGL Vizserver 1.2.
003	September 2001 Updated to support OpenGL Vizserver 1.3.
004	February 2002 Updated to support OpenGL Vizserver 2.0.
005	March 2002 Updated to support OpenGL Vizserver 2.0.1.
006	September 2002 Updated to support OpenGL Vizserver 3.0.
007	November 2002 Updated to support OpenGL Vizserver 3.0.1.
008	March 2003 Updated to support OpenGL Vizserver 3.1.
009	August 2003 Updated to support OpenGL Vizserver 3.2.
010	October 2003 Updated to support OpenGL Vizserver 3.2.1.
011	July 2004 Updated to support OpenGL Vizserver 3.4.

- 012 October 2004
 Updated to support OpenGL Vizserver 3.4.1.
- 013 November 2004
 Updated to support OpenGL Vizserver 3.4.2.

Contents

Figures	ix
About This Guide.	xi
Remote Servers.	xi
Software Platforms	xii
Related Publications	xii
Obtaining Publications	xii
Conventions	xiii
Reader Comments.	xiv
1. Running OpenGL Vizserver	1
Connecting to a Server	1
Running a Single-User Session	4
Starting a Session	5
Configuring a Session	9
Starting an Application	11
Stopping an Active Session	14
Running a Collaborative Session.	14
Starting a Local Collaborative Session	15
Starting a Remote Collaborative Session	18
Joining in a Collaborative Session	21
Input Control	27
Leaving a Collaborative Session	28
Stopping a Collaborative Session	28
Using Command-Line Options	29
Logging Out	30
Reserving Graphics Pipes.	31
Interacting with a Rendered Image	32
Using an Application with Stereo	32

A. Reservation Web Interface35
Common Header35
Login Screen36
Daily View37
Day Table37
Day Navigation Links.39
Add Reservation Form.39
Reservation View40
Login Timeout41
Index.43

Figures

Figure 1-1	OpenGL Vizserver Icon	2
Figure 1-2	OpenGL Vizserver Main Window	2
Figure 1-3	OpenGL Vizserver Login Window	3
Figure 1-4	OpenGL Vizserver Main Window (after Logging In)	3
Figure 1-5	OpenGL Vizserver Main Window (Session Starting)	5
Figure 1-6	OpenGL Vizserver Session Start Window (Single-User Session)	6
Figure 1-7	OpenGL Vizserver Session Control Window (Single-User Session)	7
Figure 1-8	OpenGL Vizserver Console Window	8
Figure 1-9	OpenGL Vizserver Main Window (Session Active)	8
Figure 1-10	Starting an Application in the OpenGL Vizserver Console Window	12
Figure 1-11	Displaying a Rendered Image in an Application Window	13
Figure 1-12	OpenGL Vizserver Main Window (Session Starting)	15
Figure 1-13	OpenGL Vizserver Session Start Window (Local Collaborative Session)	16
Figure 1-14	OpenGL Vizserver Session Control Window (Collaborative Session).	17
Figure 1-15	OpenGL Vizserver Console Window	18
Figure 1-16	OpenGL Vizserver Session Start Window (Remote Collaborative Session).	19
Figure 1-17	OpenGL Vizserver Session Control Window (Collaborative Session).	21
Figure 1-18	OpenGL Vizserver Main Window (Session Joining)	22
Figure 1-19	OpenGL Vizserver Session Join Window	23
Figure 1-20	OpenGL Vizserver Main Window (Waiting for Master's Acceptance)	24
Figure 1-21	OpenGL Vizserver Session Control Window (Nonmaster: After Joining)	25
Figure 1-22	Refusal Message	25

Figure 1-23	Asking Acceptance26
Figure 1-24	OpenGL Vizserver Session Control Window (Master: After Joining)27
Figure A-1	Header36
Figure A-2	Login Screen36
Figure A-3	Day Table38
Figure A-4	Day Navigation39
Figure A-5	Add Reservation Form40
Figure A-6	Reservation View41

About This Guide

Welcome to OpenGL Vizserver!

Remote Servers

With OpenGL Vizserver, you can interact with images and three-dimensional models utilizing the capabilities of any of the following remote SGI servers:

- SGI Onyx 3000 series with InfiniteReality3 or InfiniteReality4 graphics
- SGI Onyx 3000 series with InfinitePerformance graphics
- SGI Onyx 300 systems with InfinitePerformance graphics
- SGI Onyx 300 systems with InfiniteReality3 or InfiniteReality4 graphics
- SGI Onyx 350 systems with InfinitePerformance graphics
- SGI Onyx 350 systems with InfiniteReality3 or InfiniteReality4 graphics
- Silicon Graphics Onyx2 systems with InfiniteReality2, InfiniteReality3, or InfiniteReality4 graphics
- Silicon Graphics Octane or Octane2 systems
- Silicon Graphics Onyx4 UltimateVision systems
- Silicon Graphics Fuel systems
- Silicon Graphics Tezro systems
- Silicon Graphics Prism systems

OpenGL Vizserver transmits rendered images over a high-speed data pipe, running OpenGL applications with large RGB datasets at up to 20 Hz. Using your local workstation, you can take advantage of any of the above mentioned servers without leaving your desk.

There are two types of OpenGL Vizserver sessions: collaborative sessions and single-user sessions. Single-user sessions involve one client user (usually remote) using a free pipe on the server host. Collaborative sessions allow multiple distant users, one of them being a local client on the server host running an X server, to display and interact with the same applications in real time.

This user's guide consists of one chapter, "Running OpenGL Vizserver." This chapter provides step-by-step instructions for starting OpenGL Vizserver and interacting with images and three-dimensional models.

Software Platforms

OpenGL Vizserver is supported on the following software platforms:

- IRIX 6.5.11 or later
- Red Hat Linux 6.2 or later
- SGI ProPack 3 Service Pack 3
- Solaris 2.5.1 or later
- Windows NT 4.0, Windows 2000, and Windows XP

The images that illustrate this user's guide are taken from snapshots of the IRIX version. The shape and appearance of the window might vary from one platform to another, but the windows and options are the same.

Related Publications

The following document contains additional information that may be helpful:

- *SGI OpenGL Vizserver Administrator's Guide*

Obtaining Publications

You can obtain SGI documentation in the following ways:

- See the SGI Technical Publications Library at <http://docs.sgi.com>. Various formats are available. This library contains the most recent and most comprehensive set of online books, release notes, man pages, and other information.
- If it is installed on your SGI system, you can use InfoSearch, an online tool that provides a more limited set of online books, release notes, and man pages. With an IRIX system, select **Help** from the Toolchest, and then select **InfoSearch**. Or you can type `infosearch` on a command line.
- You can also view release notes by typing either `grelnotes` or `relnotes` on a command line.
- You can also view man pages by typing `man <title>` on a command line.

Conventions

The following conventions are used throughout this document:

Convention	Meaning
<code>command</code>	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.
function	This bold font indicates a function or method name. Parentheses are also appended to the name.
<i>variable</i>	Italic typeface denotes variable entries and words or concepts being defined.
user input	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. (Output is shown in nonbold, fixed-space font.)
[]	Brackets enclose optional portions of a command or directive line.
...	Ellipses indicate that a preceding element can be repeated.
<code>manpage(x)</code>	Man page section identifiers appear in parentheses after man page names.
GUI element	This font denotes the names of graphical user interface (GUI) elements such as windows, screens, dialog boxes, menus, toolbars, icons, buttons, boxes, fields, and lists.

Reader Comments

If you have comments about the technical accuracy, content, or organization of this document, contact SGI. Be sure to include the title and document number of the manual with your comments. (Online, the document number is located in the front matter of the manual. In printed manuals, the document number is located at the bottom of each page.)

You can contact SGI in any of the following ways:

- Send e-mail to the following address:
techpubs@sgi.com
- Use the Feedback option on the Technical Publications Library webpage:
<http://docs.sgi.com>
- Contact your customer service representative and ask that an incident be filed in the SGI incident tracking system.
- Send mail to the following address:
Technical Publications
SGI
1500 Crittenden Lane, M/S 535
Mountain View, CA 94043-1351

SGI values your comments and will respond to them promptly.

Running OpenGL Vizserver

There are two types of OpenGL Vizserver sessions: collaborative sessions and single-user sessions. Single-user sessions involve one client user (usually remote) using graphics pipes available on the OpenGL Vizserver server. Collaborative sessions allow multiple distant users, one of them being a master of the session and the rest of them being nonmasters, to display and interact with the same applications in real time.

This chapter provides step-by-step instructions for running OpenGL Vizserver. The following topics are covered:

- “Connecting to a Server” on page 1
- “Running a Single-User Session” on page 4
- “Running a Collaborative Session” on page 14
- “Using Command-Line Options” on page 29
- “Logging Out” on page 30
- “Reserving Graphics Pipes” on page 31
- “Interacting with a Rendered Image” on page 32
- “Using an Application with Stereo” on page 32

Connecting to a Server

The OpenGL Vizserver server can be any of the systems mentioned in “About This Guide” with one or more graphics subsystems. OpenGL applications installed on the server perform all image rendering. The server compresses a rendered image and transmits it to your workstation for viewing.

To connect to the OpenGL Vizserver server, follow these four steps:

Note: The connection procedures are the same for both a single-user and a collaborative session.

1. Click the OpenGL Vizserver icon on your desktop, as shown in Figure 1-1.



Figure 1-1 OpenGL Vizserver Icon

Note: You can also enter the command `vizserver` in a command-line window to display the **OpenGL Vizserver** main window.

The **OpenGL Vizserver** main window appears, as shown in the example in Figure 1-2. The **Host name** field is blank. The **Status** field indicates that you are currently disconnected from the OpenGL Vizserver server.



Figure 1-2 OpenGL Vizserver Main Window

2. Type the hostname of the OpenGL Vizserver server that you want to connect to in the **Host name** field.
3. Click the **Log in** button.

If authentication is required, the **OpenGL Vizserver Login** window appears, as shown in the example in Figure 1-3. In some cases, OpenGL Vizserver automatically logs you in, and you do not have to enter a user ID and password.



Figure 1-3 OpenGL Vizserver Login Window

4. Log in to the OpenGL Vizserver server by entering your username, password, and any other required information, then click the **OK** button.

In the **OpenGL Vizserver** main window, the **Status** field now indicates that you are connecting to the OpenGL Vizserver server. Once the **Status** field confirms that you are connected, the **Start Session**, **Join Session**, and **Log out** buttons are activated and the **Log in** button is deactivated, as shown in Figure 1-4.

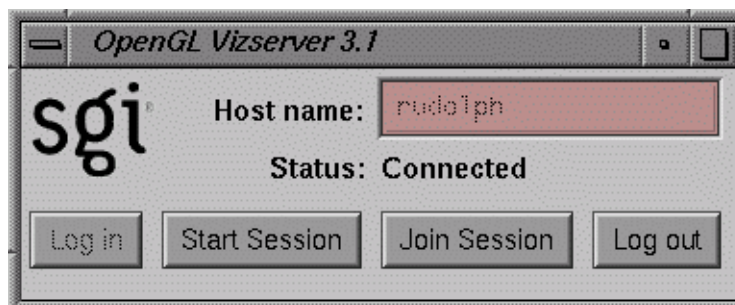


Figure 1-4 OpenGL Vizserver Main Window (after Logging In)

Alternatively, you can enter the OpenGL Vizserver server's hostname in a command-line window as follows:

```
% vizserver -h serverhost
```

If *serverhost*, is a valid server hostname and an OpenGL Vizserver server process is running, this command will take you automatically through steps 1, 2, and 3 (that is, connect to the OpenGL Vizserver server until the login stage, where the required login information is needed).

Running a Single-User Session

A single-user session is a session in which a remote client can view and interact with an application that runs and renders on the OpenGL Vizserver server. Only one client can view and interact with the applications in a single-user session. A graphics pipe on the server is allocated for the session's exclusive use.

The following sections describe how to run a single-user session:

- "Starting a Session" on page 5
- "Configuring a Session" on page 9
- "Starting an Application" on page 11
- "Stopping an Active Session" on page 14

Starting a Session

To start a single-user session, perform the following steps.

1. Connect to an OpenGL Vizserver server.

For instructions on how to connect, see “Connecting to a Server” on page 1.

2. Click the **Start Session** button in the **OpenGL Vizserver** main window.

The **Status** field indicates that the session is starting, as shown in Figure 1-5, and the **OpenGL Vizserver Session Start** window is opened, as shown in Figure 1-6.

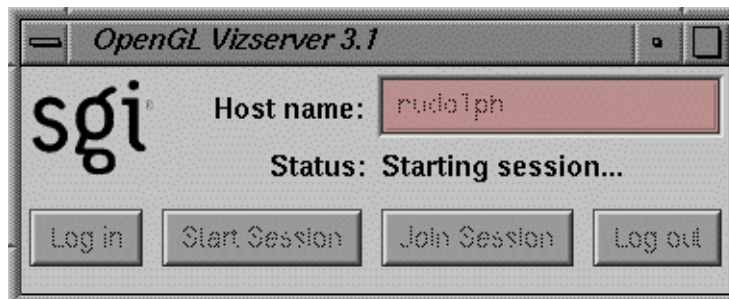


Figure 1-5 OpenGL Vizserver Main Window (Session Starting)

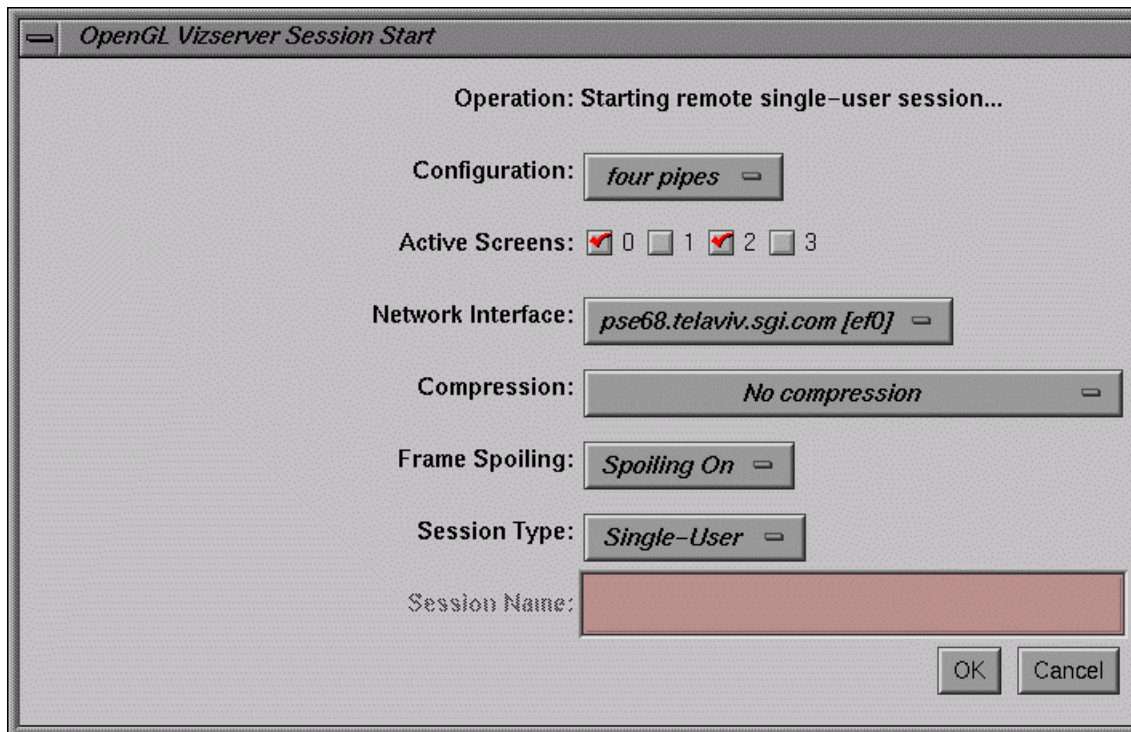


Figure 1-6 OpenGL Vizserver Session Start Window (Single-User Session)

3. In the **OpenGL Vizserver Session Start** window, set the option menu **Session Type** to **Single-User**. It is set to **Collaboration** by default. The **Session Name** field is deactivated because the session name is not needed.

In the **OpenGL Vizserver Session Start** window, you can also choose the number of graphics pipes, the screens to serve, network interface, initial compression rate, and a frame-spoiling option. For more detailed descriptions about these options, see "Configuring a Session" on page 9.

4. In the **OpenGL Vizserver Session Start** window, click the **OK** button.

The **OpenGL Vizserver Session Start** window closes. If the session starts successfully, the **OpenGL Vizserver Session Control** window and the **OpenGL Vizserver Console** window appear, as shown in Figure 1-7 and Figure 1-8, respectively. The **Status** field in the **OpenGL Vizserver** main window displays **Active**, confirming that the session is active, as shown in Figure 1-9.



Figure 1-7 OpenGL Vizserver Session Control Window (Single-User Session)

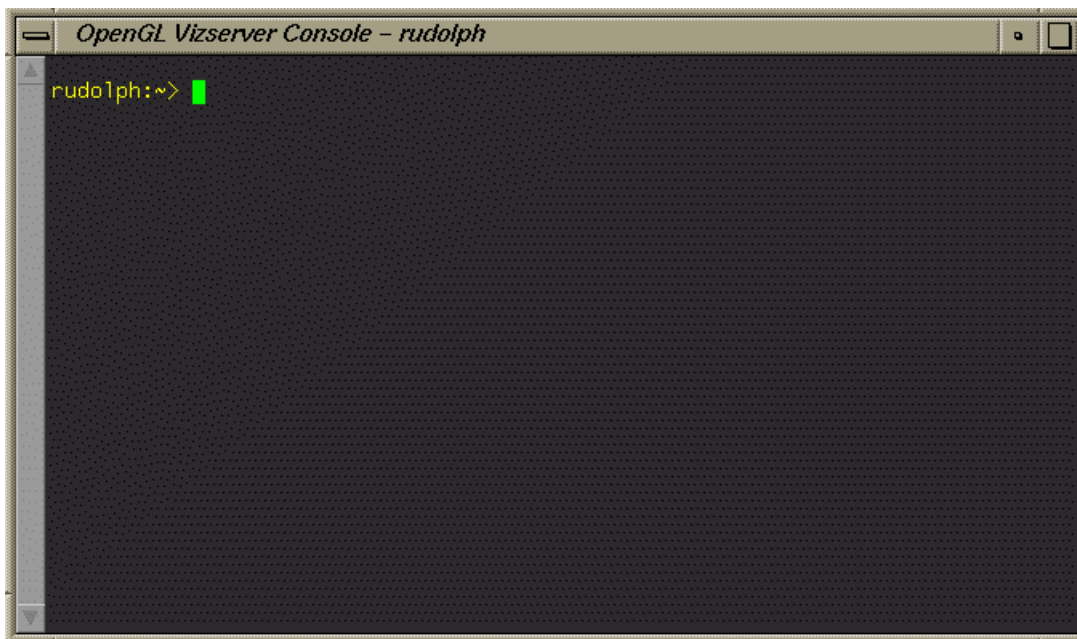


Figure 1-8 OpenGL Vizserver Console Window

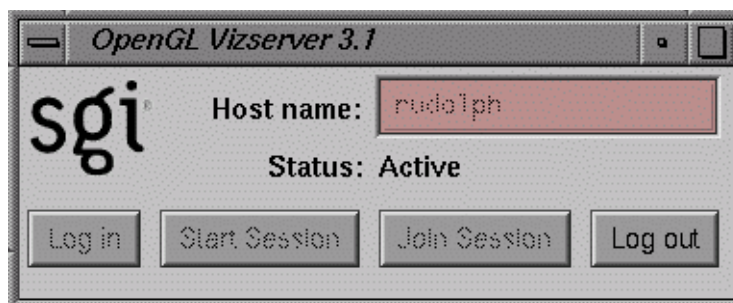


Figure 1-9 OpenGL Vizserver Main Window (Session Active)

The **OpenGL Vizserver Console** window also identifies the current OpenGL Vizserver server that you are using by displaying the server's hostname in the window's title.

Now the session is ready to accept your command to launch an application under OpenGL Vizserver.

Configuring a Session

After you press the **Start Session** button in the **OpenGL Vizserver** main window, the **OpenGL Vizserver Session Start** window appears. In this window, you can choose a configuration and a network interface, set the initial compression rate, or choose a frame-spoiling option. When this window appears, the displayed compression and spoiling values default to the last selections you made during your previous session.

The **Operation** label on the top of the **OpenGL Vizserver Session Start** window indicates the type of session that is to be started.

There is currently no multiplexing on the use of graphics pipes. Therefore, for a single-user session, only one user can use exclusively the graphics pipes configured for the session.

Note: Currently, OpenGL Vizserver does not support multiple, active, single-user sessions for the same user.

To configure your session, perform the following steps.

1. Choose a configuration from the **Configuration** option menu.

This specifies how many pipes will be used for the single-user session. The maximum number of available pipes for a user depends on the server's configuration.

2. Select which screens to serve using the **Active Screens** check boxes.

This specifies which of the screens in the session's X server will be actually served to the client. The number of available screens is the value selected in the **Configuration** option menu in step 1.

Selecting a subset of the screens for serving is useful when running software composition applications where some of the graphics pipes are used only for rendering and you want to serve only the composition target pipes. Another case is when you start a local collaborative session where the number of session pipes is determined by the already running X server and you want to serve only a subset of these pipes.

3. Choose a network interface from the **Network Interface** option menu.

This specifies which network interface the client uses to receive window updates. For example, users on multi-homed systems can specify the fast network interface.

4. Choose a compression rate from the **Compression** option menu.

When you set the compression rate, you reduce the amount of data that is required to transmit your image, trading off some image quality for greater interaction capability. Choose a compression rate that is consistent with the frame rate you want to achieve. OpenGL Vizserver compression runs globally; therefore, all applications running under OpenGL use the same selected compression scheme. If you run two applications simultaneously and you change the compression setting, both applications switch to the selected compression scheme.

You can choose one of six compression options:

Compression Option	Description
No compression	Turns off compression. In this mode, the server transmits an uncompressed image.
Lossless Delta	For each frame, the server transmits only the difference from the previous frame. This is a non-lossy compression.
Interpolated Color Cell Delta	For each frame, the server transmits only the difference from the previous frame, using Interpolated Cell Compression. The compression ratio is better than 4:1.
Color Cell Delta	For each frame, the server transmits only the difference from the previous frame, using Color Cell Compression. The compression ratio is better than 8:1.
Scaled Interpolated Color Cell Delta	For each frame, the server transmits only the difference from the previous frame, using Scaled Interpolated Cell Compression. The compression ratio is better than 16:1.
Scaled Color Cell Delta	For each frame, the server transmits only the difference from the previous frame, using Scaled Color Cell Compression. The compression ratio is better than 32:1.

5. Choose a frame spoiling option from the **Frame Spoiling** option menu.

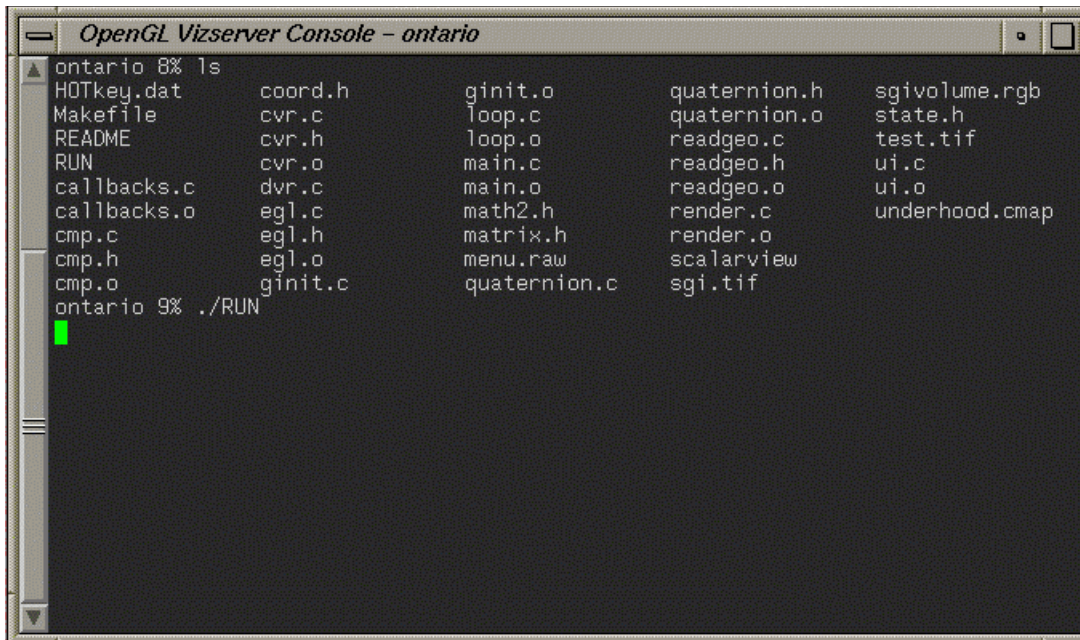
The following options are available:

Spoiling On	<p>The server generates new frames at the maximum application rate. An application frame is considered ready for readback by OpenGL Vizserver every time the application calls one of the <i>readback trigger functions</i>—glFlush(), glFinish(), and glXSwapBuffers().</p> <p>This option should be used in cases where OpenGL Vizserver is used over a low-bandwidth network link, where dropped frames are acceptable.</p>
Spoiling Off	<p>The server does not generate a new frame until the current frame is displayed on the client side. As in the <code>Spoiling On</code> mode, an application frame is read back by OpenGL Vizserver after the application calls one of the readback trigger functions.</p>
Time-Based	<p>The server sends a new frame every time the client is ready to handle it, regardless of the application's behavior. Use this option in special cases where the application being served is incompatible with the OpenGL Vizserver paradigm, which uses readback trigger functions. Typically, such applications do not call a readback trigger function at the end of each frame or draw to the same window with both OpenGL and X drawing functions.</p>

6. Click the **OK** button in the **OpenGL Vizserver Session Start** window to start the session. This brings you to the state described in the last step of the previous section and you are all set to launch an application under OpenGL Vizserver.

Starting an Application

Start the desired application by typing commands in the **OpenGL Vizserver Console** window, as shown in the example in Figure 1-10.



```
OpenGL Vizserver Console - ontario
ontario 8% ls
HOTkey.dat      coord.h        ginit.o        quaternion.h    sgivolume.rgb
Makefile        cvr.c          loop.c          quaternion.o    state.h
README          cvr.h          loop.o          readgeo.c      test.tif
RUN             cvr.o          main.c          readgeo.h      ui.c
callbacks.c     dvr.c          main.o          readgeo.o      ui.o
callbacks.o     egl.c          math2.h         render.c        underhood.cmap
cmp.c           egl.h          matrix.h        render.o
cmp.h           egl.o          menu.raw        scalarview
cmp.o           ginit.c        quaternion.c    sgi.tif
ontario 9% ./RUN
```

Figure 1-10 Starting an Application in the OpenGL Vizserver Console Window

Alternatively, you can start the desired application as follows:

1. Use the `telnet` or `rlogin` command to access the OpenGL Vizserver server machine.
2. Start the desired application using the `vsx(1)` command.

The application renders the image. Then the OpenGL Vizserver server transmits it to your local workstation and displays it on your screen in the application window, as shown in the example in Figure 1-11.

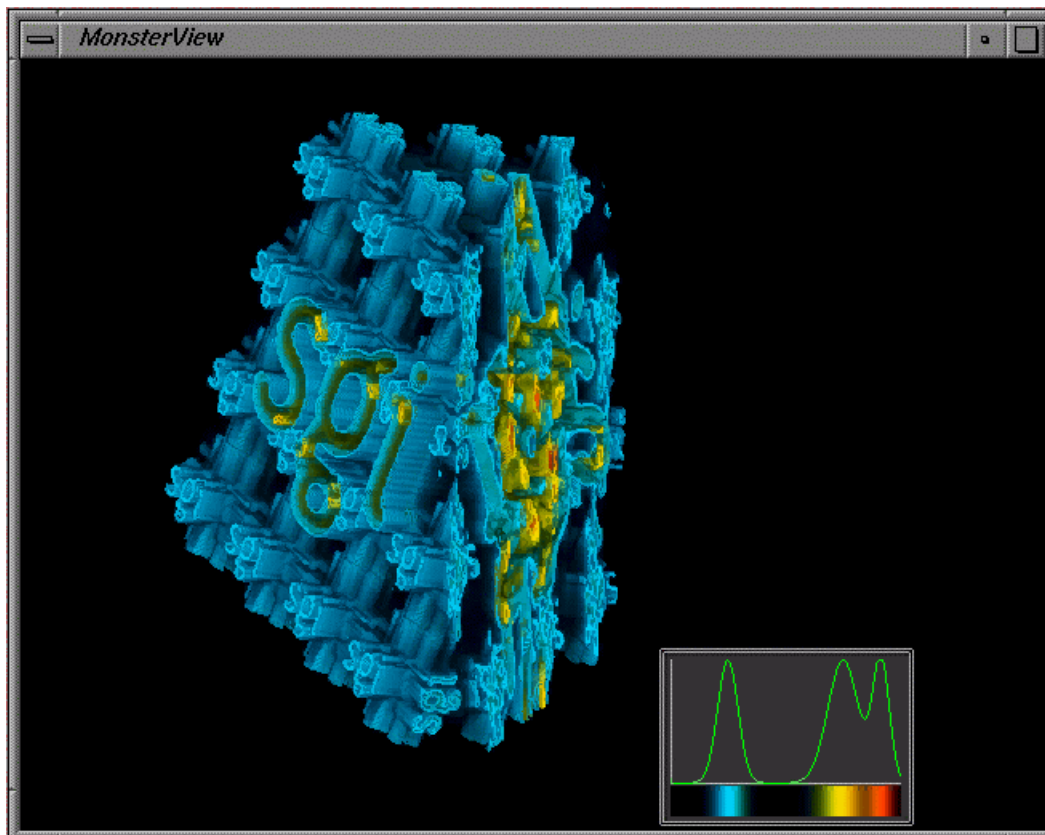


Figure 1-11 Displaying a Rendered Image in an Application Window

After starting an application, your image appears in an application window. Then use the **OpenGL Vizserver Session Control** window, shown in Figure 1-7 on page 7, to reset compression rate, change the spoiling option, redraw your image, or stop the active session.

Clicking the **Redraw Windows** button in the **OpenGL Vizserver Session Control** window redraws the image at the selected compression rate. You can reset the compression and redraw the image at any time.

Stopping an Active Session

To stop an active session, click the **Stop session** button in the **OpenGL Vizserver Session Control** window, shown in Figure 1-7 on page 7.

You can stop a session at any time. If you stop a session while your application is being run, the application is stopped and the session is stopped. If you close the **OpenGL Vizserver Session Control** window, it is the same as clicking the **Stop session** button.

After stopping the session, the **Start Session** and **Join Session** buttons in the **OpenGL Vizserver** main window become activated to allow you to start a session again or join in a collaborative session that is running on the same server that you are logging in to.

Running a Collaborative Session

A collaborative session is a session in which multiple clients can view and interact with the same application to work together in real time.

There are two types of collaborative sessions: local and remote. A local collaborative session is a session that involves a local client on the server starting a session on its local display, and multiple remote clients joining in the session later. A remote collaborative session is a session that involves a remote client starting a session on a newly created X server, and multiple clients joining in the session later.

Whoever starts a session becomes the master of the session. But only a local client can start a local collaborative session.

The following sections describe how to run a collaborative session:

- “Starting a Local Collaborative Session” on page 15
- “Starting a Remote Collaborative Session” on page 18
- “Joining in a Collaborative Session” on page 21

- “Input Control” on page 27
- “Leaving a Collaborative Session” on page 28
- “Stopping a Collaborative Session” on page 28

Starting a Local Collaborative Session

To start a local collaborative session, perform the following three steps.

1. Connect to an OpenGL Vizserver server before starting the session or joining the session.

For instructions about how to connect, see “Connecting to a Server” on page 1.

2. Click the **Start Session** button in the **OpenGL Vizserver** main window.

The **Status** field in the **OpenGL Vizserver** main window indicates that the session is starting, as shown in Figure 1-12, and the **OpenGL Vizserver Session Start** window is opened, as shown in Figure 1-13.

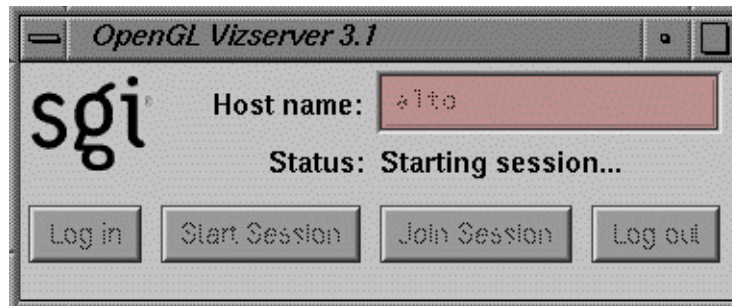


Figure 1-12 OpenGL Vizserver Main Window (Session Starting)

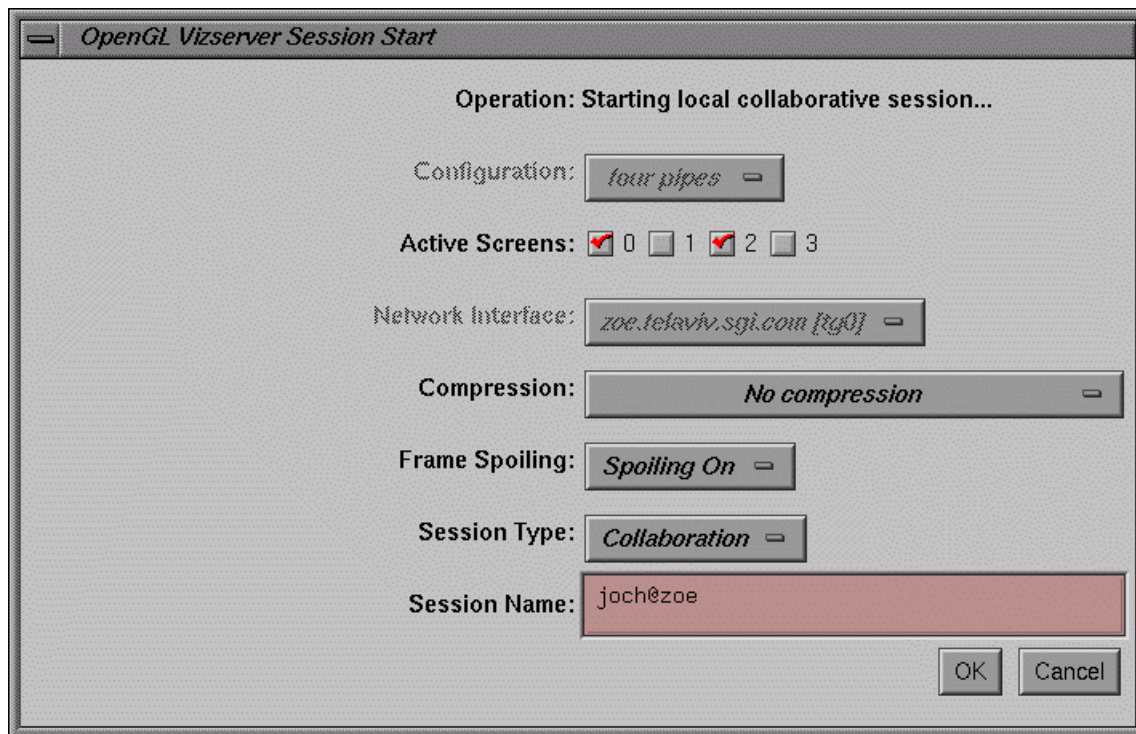


Figure 1-13 OpenGL Vizserver Session Start Window (Local Collaborative Session)

The OpenGL Vizserver server understands internally whether the client is connected to the local server or a remote server.

Note: When starting a local collaborative session, all fields except **Active Screens**, **Session Type**, and **Session Name** are disabled in the **OpenGL Vizserver Session Start** window because the display will be local and no network or bandwidth settings are needed.

3. Enter a session name in the **Session Name** text field for the collaborative session.
A default session name is given as *username@serverhost*, but the master of the session can specify any string.

Note: A session name is required for a collaborative session. Later, other users must use the session name to join the session.

In the example in Figure 1-13, you can see that it is starting a local collaborative session because *serverhost* from the default session name and the machine name from the **Network Interface** option menu are the same.

4. Click the **OK** button in the **OpenGL Vizserver Session Start** window.

The **OpenGL Vizserver Session Start** window closes. If starting the session is successful, the **Status** field in the **OpenGL Vizserver** main window indicates that the session is **Active**. The **OpenGL Vizserver Session Control** window and the **OpenGL Vizserver Console** window will also appear, as shown in Figure 1-14 and Figure 1-15, respectively.

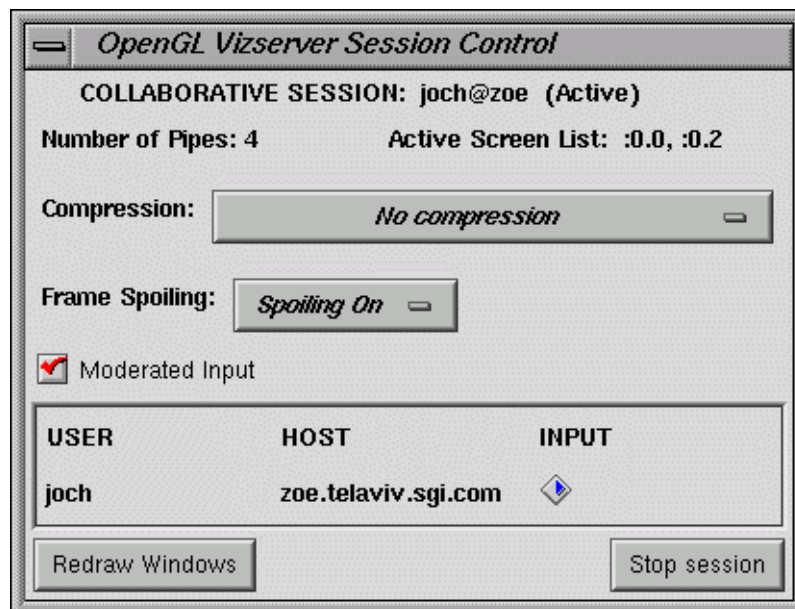


Figure 1-14 OpenGL Vizserver Session Control Window (Collaborative Session)

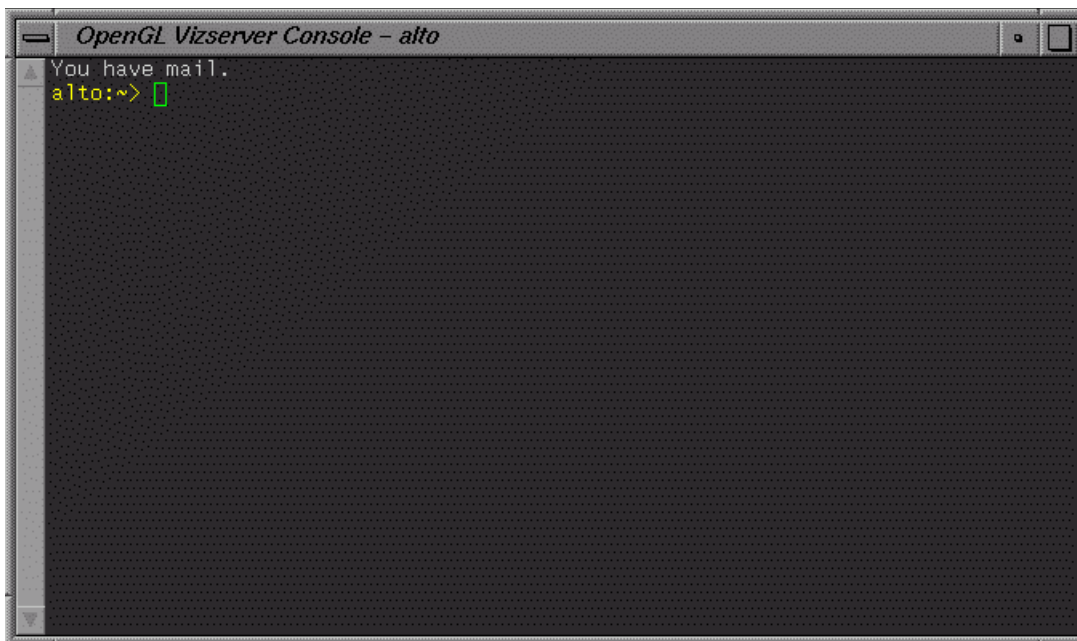


Figure 1-15 OpenGL Vizserver Console Window

Note: The **OpenGL Vizserver Session Control** window for a collaborative session is different from one for a single-user session; it shows the list of clients participating in the session and has a selection box for the input moderation between multiple clients.

Now the session is ready to accept your command to launch an application under OpenGL Vizserver.

Starting a Remote Collaborative Session

To start a remote collaborative session, perform the following four steps.

1. Click the **Start Session** button in the **OpenGL Vizserver** main window.

The **Status** field in the **OpenGL Vizserver** main window indicates that the session is starting, as shown in Figure 1-5 on page 5. Then the **OpenGL Vizserver Session Start** window is opened, as shown in Figure 1-16.

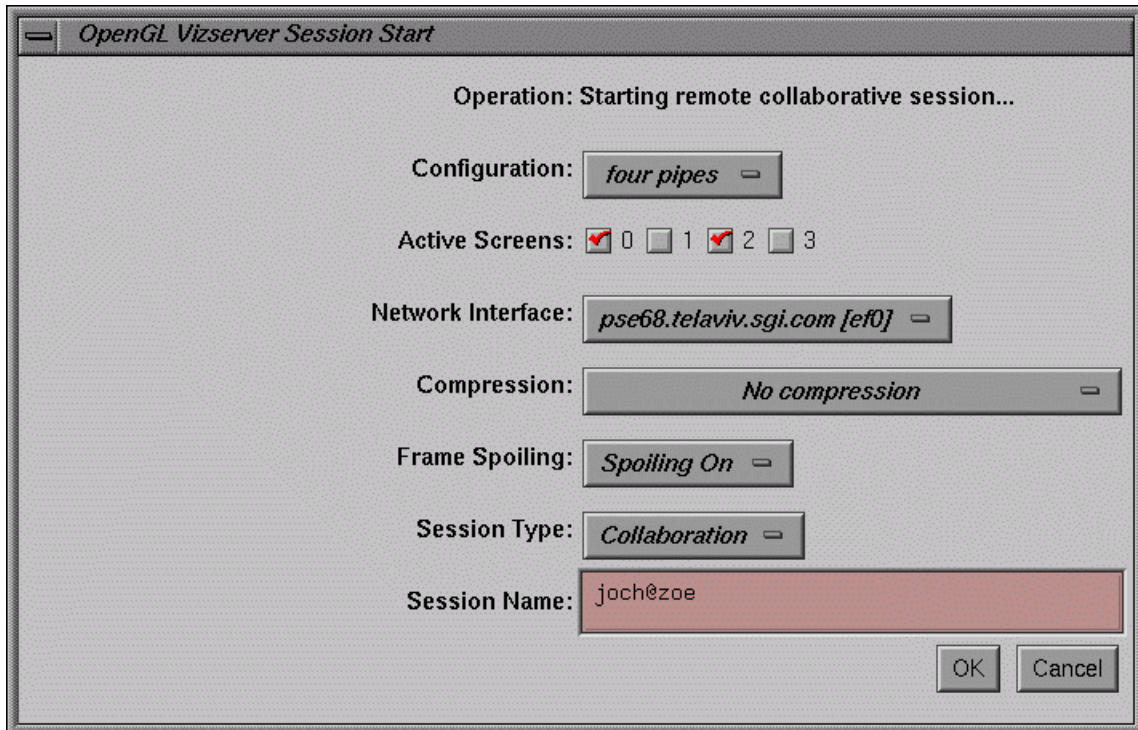


Figure 1-16 OpenGL Vizserver Session Start Window (Remote Collaborative Session).

Note: When starting a remote collaborative session, as contrasted to starting a local collaborative session, all the options in the **OpenGL Vizserver Session Start** window are enabled.

2. Enter a session name for the collaborative session in the **Session Name** text field.

A default session name is given as *username@serverhost*, but the master of the session can select any string.

Note: A session name is required for a collaborative session. Later, other users must use the session name to join the session.

In the example in Figure 1-16, you can see that it is starting a remote collaborative session because *serverhost* from the default session name and the machine name from the **Network Interface** option menu are different.

3. Select different number of graphics pipes, network interface, compression type, and so on, if necessary.

For a detailed description of each option, see “Configuring a Session” on page 9.

Here, select the **Color Cell Delta(8:1)** in the **Compression** option menu as an example.

4. Click the **OK** button in the **OpenGL Vizserver Session Start** window.

The **OpenGL Vizserver Session Start** window closes and, if starting the session is successful, the **Status** field in the **OpenGL Vizserver** main window indicates that the session is **Active**. The **OpenGL Vizserver Console** window and the **OpenGL Vizserver Session Control** window will also appear, as shown in Figure 1-8 on page 8 and Figure 1-17, respectively.

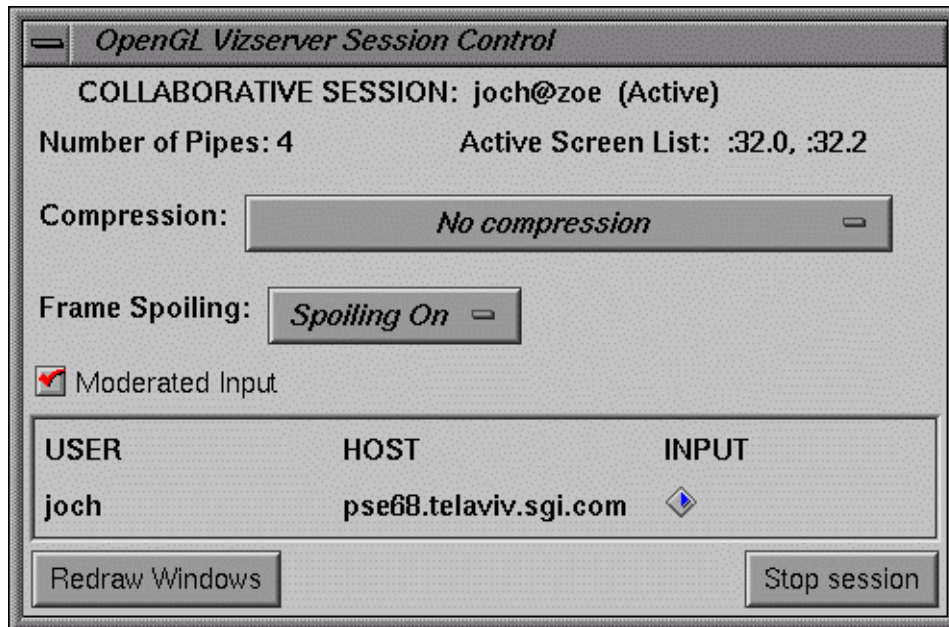


Figure 1-17 OpenGL Vizserver Session Control Window (Collaborative Session)

The OpenGL Vizserver Session Control window has the Color Cell Delta(8:1) compression type, as shown in Figure 1-17, because the user selected it in the previous step.

Now the session is ready to accept your command to launch an application under OpenGL Vizserver.

Joining in a Collaborative Session

Joining in a collaborative session requires that the session is already started and active by performing the steps in previous sections. Also, a joining client needs to get the approval of the master of the session.

To join in a collaborative session requires actions from both the joining client and the master client.

Joining Client

You must perform the following actions if you are the joining client:

1. Click the **Join Session** button in the **OpenGL Vizserver** main window.

The **Status** field in the main window indicates that the session is joining, as shown in Figure 1-18, and the **OpenGL Vizserver Session Join** window opens, as shown in Figure 1-19.



Figure 1-18 OpenGL Vizserver Main Window (Session Joining)

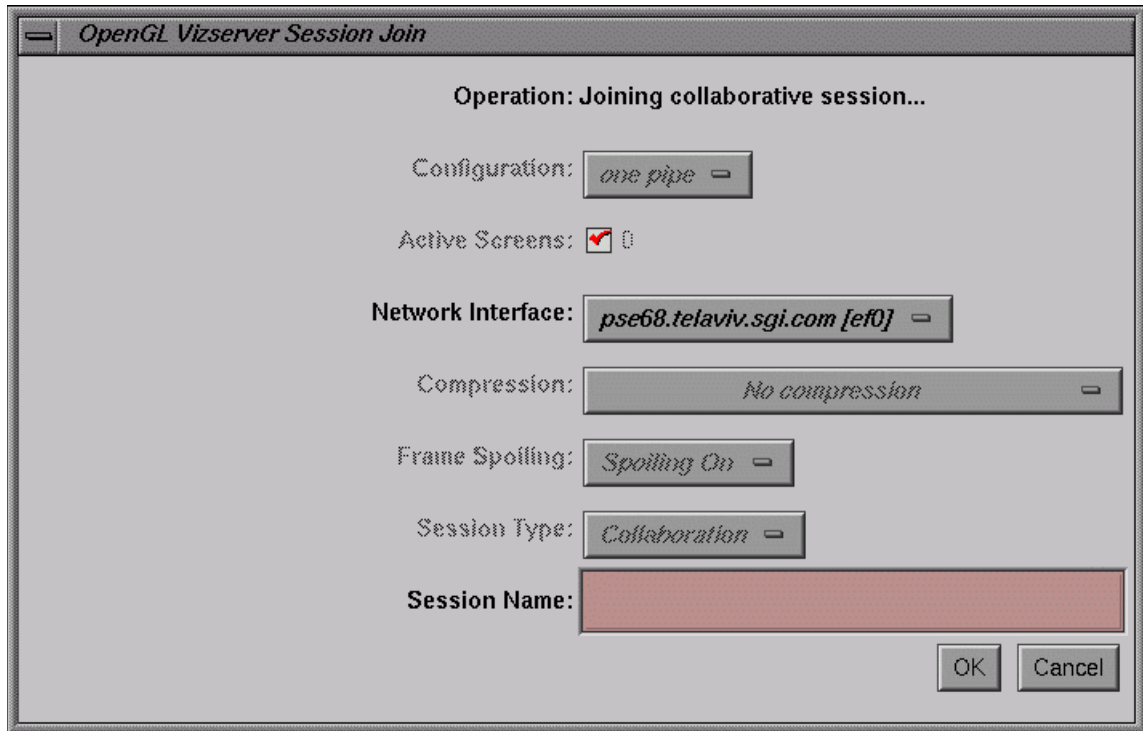


Figure 1-19 OpenGL Vizserver Session Join Window

2. Enter the name of the collaborative session to join in the **Session Name** text field of the **OpenGL Vizserver Session Join** Window.

This session name must match exactly the session name chosen by the master who started the session.

3. Select a different network interface, if necessary.

Note: When joining in a collaborative session, all the fields are disabled in the **OpenGL Vizserver Session Join** window (except the **Network Interface** and **Session Name** fields) because the number of graphics pipes being used by the session is determined by the master when the session is started, the compression and frame spolling options are the ones being used in the session, and the session is obviously the collaborative one.

4. Click the **OK** button in the **OpenGL Vizserver Session Join** window and wait for the master's approval.

While waiting for the master's acceptance, the **OpenGL Vizserver** main window shows the status of waiting, as shown in Figure 1-20.

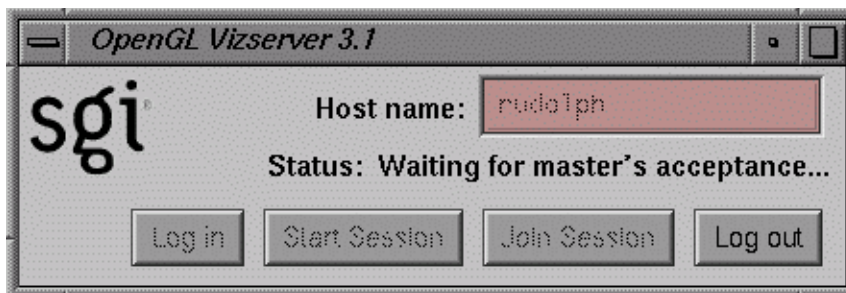


Figure 1-20 OpenGL Vizserver Main Window (Waiting for Master's Acceptance)

5. If the master accepts the request, the newly joined client and other nonmaster clients, if any, have the updated client list in their **OpenGL Vizserver Session Control** window, as shown in Figure 1-21.

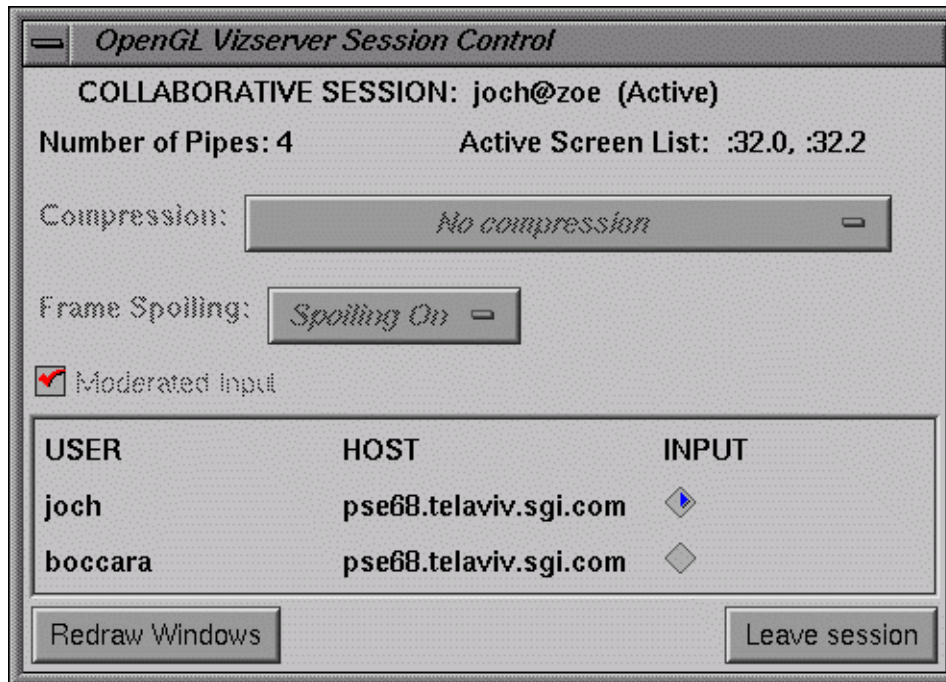


Figure 1-21 OpenGL Vizserver Session Control Window (Nonmaster: After Joining)

If the master refuses the request, the client will get a refusal message, as shown in Figure 1-22.

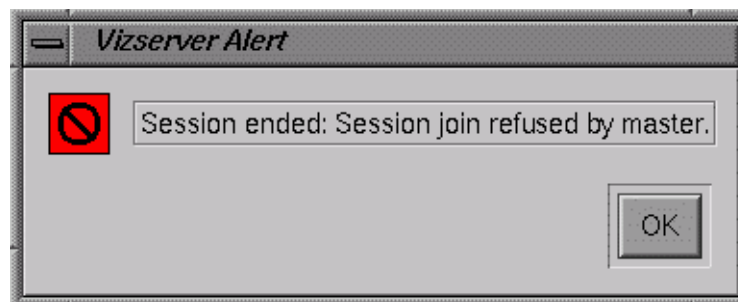


Figure 1-22 Refusal Message

Master Client

The OpenGL Vizserver server sends a question message to the master of the session that a client wants to join in. You must perform the following actions if you are the master client:

1. Click the **OK** button for approval or the **Cancel** button for denial in the question dialog box, as shown in Figure 1-23.

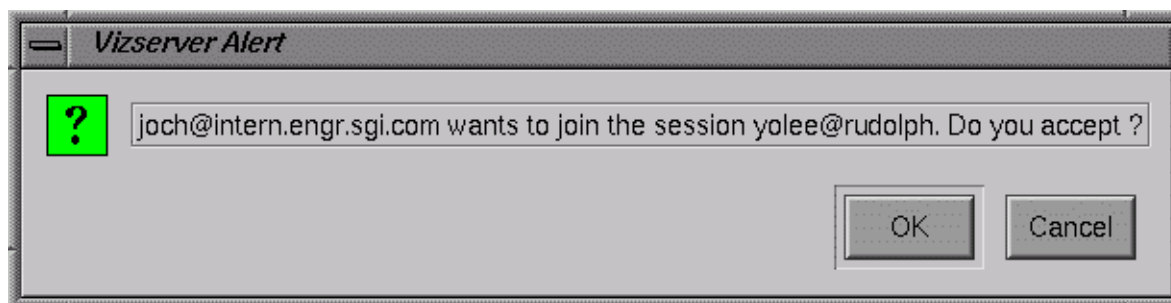


Figure 1-23 Asking Acceptance

2. If the master accepts a new joining request in the session, all of the nonmaster clients, including the newly joined client, get the **OpenGL Vizserver Session Control** window, as shown in Figure 1-21. The master gets the **OpenGL Vizserver Session Control** window, as shown in Figure 1-24.

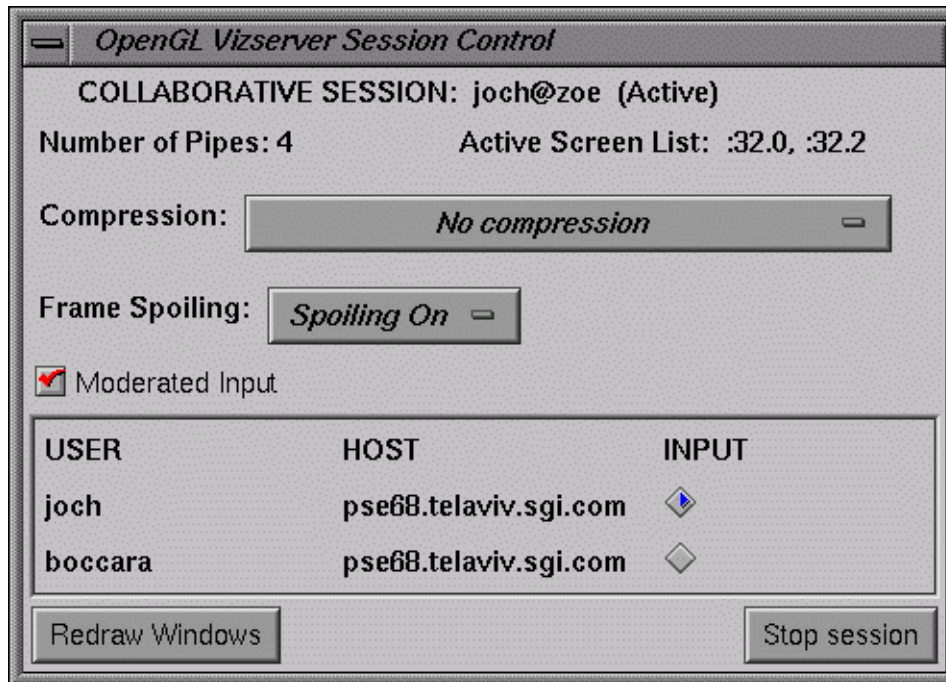


Figure 1-24 OpenGL Vizserver Session Control Window (Master: After Joining)

If an application has already been running, the newly joined client will see the application as soon as the joining process is done.

Input Control

A method is provided to control inputs between multiple clients in a collaborative session. The master of the session has the privilege to turn the input control on or off.

In a collaborative session, clients have two additional elements in the **OpenGL Vizserver Session Control** window: a **Moderated Input** check box and a list of clients in the session.

The client list consists of three columns: **USER**, **HOST**, and **INPUT**. The **USER** is the user's login name of the client. The **HOST** is the hostname of the client machine. The

INPUT is a radio button indicating which client owns the input in the session at the moment.

Note: Only the client that has its **INPUT** radio button turned on has the input control in the session and can change **Compression** and **Frame Spoiling** options.

The **Moderated Input** check box displays whether nonmasters can take their own input control. This check box is deactivated in nonmasters' **OpenGL Vizserver Session Control** windows.

Note: Only master clients can turn the **Moderated Input** check box on or off.

If the **Moderated Input** check box is on, only the master can give input control to the clients in the session.

If the **Moderated Input** check box is off, nonmaster clients can get their input control by turning on the radio button. They cannot give input control to others. Only master clients can give others input control.

Leaving a Collaborative Session

Nonmaster clients have the **Leave session** button in the **OpenGL Vizserver Session Control** window. To leave a session, press the **Leave session** button, as shown in Figure 1-21 on page 25.

When a nonmaster client leaves the session, the session continues and other clients in the session will get the updated list of clients in the session from the OpenGL Vizserver server.

Stopping a Collaborative Session

Only a master client has the **Stop session** button in the **OpenGL Vizserver Session Control** window. To stop a session, press the **Stop session** button, as shown in Figure 1-24 on page 27.

Only the master of a session can stop the session. In other words, the client that starts a session is the only one that can stop the session.

Note: Nonmaster clients cannot stop the session.

When a master client stops the session, the session ends and all the clients, including the master in the session, will have their **OpenGL Vizserver Session Control** and **OpenGL Vizserver Console** windows closed. In addition, nonmaster clients in the session get the message that the session is terminated by the master.

If an application is being run when the session is stopped, the application will be stopped too. The master of the session should be aware of this and how it may affect the clients participating in the session.

Using Command-Line Options

The preceding sections describe how you can run OpenGL Vizserver using its graphical interfaces. You can also use the command-line options of the `vizserver` command to perform all of the tasks, from connecting to the server to starting a session or joining a session.

Table 1-1 briefly describes the `vizserver` command-line options.

Table 1-1 Command-Line Options of the Command `vizserver`

Option	Action
<code>-h serverhost</code>	Connects to the OpenGL Vizserver server with the hostname <i>serverhost</i> .
<code>-a auth_fields</code>	Uses the values of <i>auth_fields</i> for the login process. The value <i>auth_fields</i> is a comma-separated list of authentication values—for example, <code>-a username, password</code>
<code>-S session_name</code>	Starts a session automatically. If you do not specify <i>session_name</i> , OpenGL Vizserver starts a single-user session. If you specify <i>session_name</i> , OpenGL Vizserver starts a collaborative session with the name <i>session_name</i> .

Table 1-1 Command-Line Options of the Command `vizserver` (continued)

Option	Action
<code>-J session_name</code>	Joins the collaborative session <i>session_name</i> . The value <i>session_name</i> is required. This option cannot be used with the <code>-S</code> option.
<code>-p npipe</code>	Specifies the number of graphics pipes (<i>npipe</i>) when starting a session. If you do not specify this option, the default number of graphics pipes is one. This option is meaningless when used with the <code>-J</code> option.
<code>-c compressor</code>	Uses the specified value <i>compressor</i> when starting a session. The value <i>compressor</i> is one of the following: <code>vsIFCCC</code> , <code>vsIFICC</code> , <code>vsIFSCCC</code> , <code>vsIFSICC</code> , or <code>vsIFLLC</code> . This option is meaningless when used with the <code>-J</code> option.
<code>-s mode</code>	Sets the spoiling mode when starting a session. The <code>-s</code> and <code>-s 0</code> options turn spoiling off. The <code>-s 1</code> option turns spoiling on (this is the default value). The <code>-s 2</code> option turns on the Time-Based readback mode.
<code>-x</code>	Enables exiting upon a program failure.
<code>-A active_screens</code>	Turns on OpenGL Vizserver serving for the screens specified by the values of <i>active_screens</i> . The value <i>active_screens</i> is a comma-delimited list of screen numbers—for example, <code>-A 0, 2</code> . If you do not specify the <code>-A</code> option, the default behavior is to serve all screens.

The options `-p`, `-c`, and `-s` are meaningless when used with the `-J` option because they can be set when starting a session or changed when you have an input control after joining a collaborative session.

For more details, see the `vizserver(1)` man page.

Logging Out

If you want to log out, click the **Log out** button in the **OpenGL Vizserver** main window.

If a client logs out while the single-user session is active, the session is stopped and the client is logged out from the server.

If the master logs out while the collaborative session is active, the session stops and the master is logged out from the server. However, other clients (nonmasters) in the collaborative session are not logged out. They will get the message that the session is terminated by the master. At this stage, they can start a new session or join in another session that is already active in the server, if any, without logging out.

If a client (nonmaster) logs out while the collaborative session is active, the client is logged out from the server but the collaborative session continues without any hinderance and the rest of the clients in the collaborative session will receive the updated version of the users list in their **OpenGL Vizserver Session Control** window.

Reserving Graphics Pipes

Note: This does not apply to starting a local collaborative session.

Depending on the server's configuration, you might need to reserve one or more graphics pipes on the server before you start a session. Otherwise, the start session operation will fail, with the following error message:

```
Pipe not reserved.
```

A pipe reservation is defined by a time slot (start time and end time), a user name, and the number of pipes to be reserved. In order to add a reservation, you need to use a reservation client. A sample reservation web interface can be started from the following URL:

```
http://serverhost/vizserver/reservation
```

The path element *serverhost* is the full name of the host machine where the OpenGL Vizserver server is running. For more information about this web interface, see Appendix A, "Reservation Web Interface".

Once you have started a session for which you have reserved enough pipes, this session will be available for use until the end of the time slot. A warning message will be sent to you before the end of the reservation time slot.

Interacting with a Rendered Image

Because OpenGL Vizserver provides an application-transparent service, your applications perform as if they are using the indirect rendering features of X11 and GLX. Interacting with one of your applications is the same as interacting with any other X-based application. The application is not aware that the OpenGL Vizserver system is rendering the image.

In a collaborative session, multiple clients participating in the session can interact with the application. However, only the client that has the input control radio box turned on can interact with the application. For a description of how to get the input control in a collaborative session, see “Input Control” on page 27.

Note: The local client’s pointer is displayed on the remote client in a collaborative session only.

In order to overcome incompatibilities between different window managers, a special meaning was assigned to the key combination `CTRL-F3`. Pressing this combination on application windows on the client’s side will minimize all OpenGL Vizserver application windows under any local windows.

Using an Application with Stereo

Stereoscopic, or stereo, viewing is accomplished by presenting different views of the same scene to each eye. In a traditional OpenGL quad-buffered stereo application, the following steps occur:

1. Set the draw buffer to the back left buffer.
2. Draw the OpenGL scene.
3. Set the draw buffer to the back right buffer.
4. Set up the right eye projection (`glTranslate()`).
5. Draw the OpenGL scene.
6. Swap buffers.

Stereo support in OpenGL Vizserver means simply delivering multiple images (left and right eye) to the client.

In order to support stereo, the hardware must support the two following conditions:

Condition 1 A visual mode that has both left and right eye color buffers

Condition 2 A video mode that refreshes multiple fields (left and right eye fields)

If both conditions are not satisfied, OpenGL Vizserver does not allow stereo image delivery to function.

On the server, OpenGL Vizserver stereo support is limited to only SGI IRIX machines. On the client, even though there are various assortment of OpenGL Vizserver clients, stereo support is currently limited only on SGI IRIX machines.

You can verify that Condition 1 is met on both client and server sides by checking to see if a visual on the platform supports stereo. This visual checking can be done by either the `glxinfo` or `findvis` command.

For example, a `glxinfo` command entered on an IRIX platform will show you something similar to below. Here, the `stro` field shows that one (0x2d) supports stereo:

```

visual  x  bf lv rg d st  r  g  b  a  ax dp st accum buffs  ms
id dep cl sp sz l  ci b ro sz sz sz sz bf th cl  r  g  b  a ns b
-----
0x2c 12 tc . 16 . r y . 4 4 4 4 . . . 16 16 16 16 . .
0x2d 12 tc . 16 . r y y 4 4 4 4 . 24 8 16 16 16 16 . .

```

You can verify that Condition 2 is met by checking the video format files on that IRIX system. These files are located in the `/usr/gfx/ucode/PLATFORM/dg4/vof` directory, where `PLATFORM` specifies the graphics board family for that machine. If there is a mode that has `s` in the name, this usually refers to a stereo capable format. See the `setmon(1G)` and `ircombine(1G)` man pages for more information regarding these formats.

The current application and the video mode between both client and server must simultaneously support stereo. Not having the server in a stereo video mode will thus cause stereo image delivery not to work. If the client is not in a stereo video format, stereo image delivery may work, but stereo effects will not be seen.

Since not every application running under OpenGL Vizserver will be stereo based, there is no need to capture and deliver stereo images all the time. OpenGL Vizserver can be

configured to understand the changes transparently when an application uses a stereo visual or the video mode on the server changes. For details on how to do this, see the `vizserver(1)` man page.

In a collaborative session, various clients with different platform capabilities can mean a heterogenous mix of both nonstereo- and stereo- enabled clients. The basic rule for this scenario is as follows. If one remote client that supports stereo is currently joined in a collaborative session, then a stereo support on the OpenGL Vizserver server is enabled. This allows stereo image delivery and operation to work on those clients only. Clients that do not meet the stereo requirements will not see video mode changes nor will they receive extra bandwidth related to the right eye image. If the stereo enabled client leaves the session, then stereo delivery will be disabled. To be even more simple, stereo will work just as it would in collaborative sessions as it would in single-user sessions.

For stereo image delivery between two machines, you can expect about half the performance for the application through OpenGL Vizserver as compared to nonstereo performance. This is simply due to OpenGL Vizserver doing twice the work for both left and right eye images in one application frame update. You can also expect about double the network traffic.

Reservation Web Interface

The reservation web interface is a client of the reservation database on the remote server. It can be accessed through a URL that resides on the remote host. The URL is `http://remotehost/vizserver/reservation`, where *remotehost* is the full name of the remote host where the server is running.

To log onto the reservation system, enter your authentication data (if needed) in the fields of the welcome screen, and press the **Login** button.

Common Header

All the following pages contain a header section, as shown in Figure A-1. This header shows—in addition to the SGI and OpenGL Vizserver logos—the name of the currently logged in user, the name of the remote host, and the system time on the remote host. On the left side is a **LOGOUT** link, that allows you to log out from the reservation server at any time and go back to the login page for the reservation system.



Figure A-1 Header

Login Screen

If requested, fill in your authentication data in the login page. The name of the OpenGL Vizserver remote host is displayed at the top of the login page. Figure A-2 shows a typical UNIX login screen, with a user name and a password field. If login parameters are correct, you enter the daily view of the reservation database, as described in the following section.

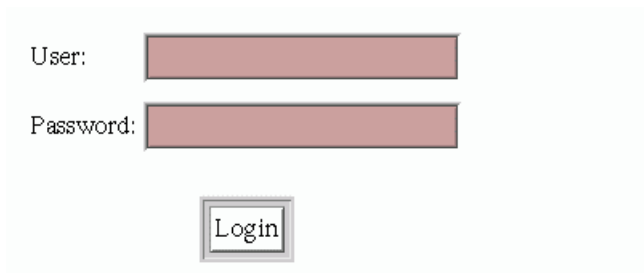


Figure A-2 Login Screen

Daily View

Once logged in, the first and main page of the reservation web interface is the daily view. The daily view shows a table of the reservations for a given day (the default is today), assigned to each pipe allocated for OpenGL Vizserver on the remote host. From this page you can select a reservation time slot to get its details (see, “Reservation View” on page 40); select a time on the table’s vertical time scale to go to the Add Reservation page; or jump to another day through one of the day navigation links.

Day Table

The day table, as shown in Figure A-3, represents reservations as rectangles; the time scale is on the vertical axis and the associated pipe are on the horizontal axis. There are as many pipe columns as there are pipes on the remote server allocated for OpenGL Vizserver. The user name of the owner is written on the time slot. Depending on the browser’s functionality, a yellow hint note is displayed when the mouse moves on the reservation time slot, showing details (start and end time) of the reservation. As color codes are used, one color is assigned to the reservations of the currently logged in user, and another color is assigned to all other users’ reservations. Each user name in a time slot is a link to a more detailed view of this reservation’s data (see “Reservation View” on page 40). Each hour item on the vertical time axis is a link to a form for adding a new reservation (see “Add Reservation Form” on page 39), starting by default at the selected time. All these links open a separate browser window.

Thursday, February 21, 2002

Time	pipe 1	pipe 2
00:00		
01:00		
02:00		
03:00		
04:00		
05:00		
06:00		
07:00		
08:00		
09:00		
10:00	guest	guest
11:00		
12:00	boccaro	
13:00		joch
14:00		
15:00		
16:00		
17:00		
18:00		
19:00		
20:00		
21:00		
22:00		
23:00		

Legend: Mine: Others':

Figure A-3 Day Table

Day Navigation Links

Links are provided to jump to the previous, next, or current day, as shown in Figure A-4. Another form is provided to directly select the day you want to view, this day being defined by its year, month, and day of the month.

The image shows a web form for day navigation. It starts with the text "Go to:" followed by four input fields: a text box containing "17", a dropdown menu showing "September", a text box containing "2001", and a "Go" button. To the right of these fields are three blue underlined links: "Previous Day", "Next Day", and "Today".

Go to: [Previous Day](#) [Next Day](#) [Today](#)

Figure A-4 Day Navigation

Add Reservation Form

The Add Reservation form, as shown in Figure A-5, contains fields to enter the day, start time, and duration of the reservation you want to add, as well as the number of pipes to be reserved. The owner is always the user who is logged in. A reservation will then be added after checking that this does not create a pipe overallocation for the time slot. In case of such a conflict, the operation will fail. On success, the daily view will be updated with the new reservation.

Add Reservation for **guest**:

Day:

Start Time: :

Duration: minutes

Note: Duration must be at least 1 minutes.

Number of reserved pipes:

Figure A-5 Add Reservation Form

Reservation View

The reservation view, as shown in Figure A-6, provides the following details about a reservation: number of reserved pipes, start time, end time, and owner user. If you are a privileged user, or the owner of the displayed reservation, you can delete this reservation by pressing the **Delete** button. If not, the deletion operation will fail and an error message will be displayed.

Reservation Details:

Owner: **guest**

Start Time:	10:00	Thursday, February 21, 2002
End Time:	11:00	Thursday, February 21, 2002

Number of reserved pipes: **2**

[Delete Reservation](#)

Figure A-6 Reservation View

Login Timeout

A timeout is defined in a configuration file and is reserved for an administrator's use. After a timeout has passed since your last operation on the reservation web interface, the login page will be displayed and you must log in again to the reservation system.

Index

A

active screens, 9, 30
applications
 interacting with, 32
 starting, 11-13

C

collaborative sessions
 definition, xii, 1, 14
 joining, 21
 starting, 14
Color Cell Compression (CCC), 10
Color Cell Delta, 10
commands
 rlogin, 12
 telnet, 12
 vizserver, 2, 29
 vsx, 12
compression options, 10
configuration options
 compression rate, 10
 frame spoiling, 11
 network interface, 10
 pipes, 9
configuration procedure, 9
console window
 starting application, 11

E

errors
 pipe not reserved, 31

F

frame spoiling, 11, 30

G

glFinish() function, 11
glFlush() function, 11
glXSwapBuffers() function, 11

H

host name, 2

I

images
 displaying, 13
 interacting with, 32
Interpolated Cell Compression, 10
Interpolated Cell Compression (ICC), 10
IRIX platforms, xii

L

Linux platforms, xii
login, 3

M

main window
 connected status, 3
 display, 2
 session active, 8
 session starting, 5

N

network interface options, 10

O

OpenGL Vizserver icon, 2

P

pipes
 configuring, 9
 reserving, 31
platforms
 IRIX, xii
 Red Hat Linux, xii
 SGI ProPack, xii
 SGI servers, xi
 Solaris, xii
 Windows 2000, xii
 Windows NT, xii
 Windows XP, xii

R

readback trigger functions, 11
Red Hat platforms, xii
rlogin command, 12

S

Scaled Color Cell Compression (SCCC), 10
Scaled Color Cell Delta, 10
Scaled Interpolated Cell Compression (SICC), 10
Scaled Interpolated Color Cell Delta, 10
server
 connecting to, 1-3
session
 stopping, 14
session name, 23
sessions
 collaborative, xii, 1, 14
 single-user, xii, 1, 5
 stopping, 14
SGI ProPack platforms, xii
SGI servers, xi
single-user sessions, xii, 1, 5
software platforms (See also platforms.), xii
Solaris platforms, xii
spoiling, 11, 30
Stop Session, 14

T

telnet command, 12

V

vizserver command, 2, 29
vsx command, 12

W

Windows platforms, xii

