

White Paper

SGL<sup>®</sup> OpenGL Multipipe<sup>™</sup>



# SGI® OpenGL Multipipe™

*This document describes SGI OpenGL Multipipe and explains the uses for this application and its general architecture.*

## Overview

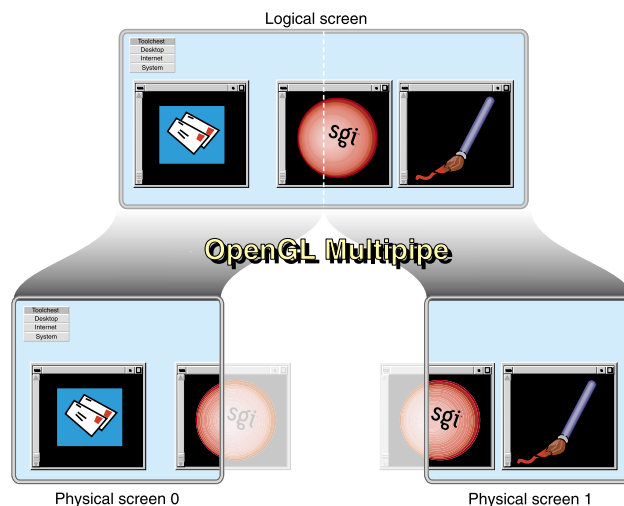
SGI has always been focused on high-end graphics solutions. The SGI® Onyx® family of scalable visualization supercomputers allows you to associate multiple graphics pipes on one single-system-image machine in order to reach new visualization performances. These multipipe systems are commonly used to drive expanded visualization systems such as SGI Reality Center™ facilities. OpenGL Multipipe extends the use of these powerful supercomputers to a broad spectrum of graphics applications without the requirement to modify the applications.

Many existing graphics applications—like Netscape™ or Open Inventor™ applications, for example—are constrained to run on one single pipe. On these single-pipe applications, you can choose the pipe on which to open the application's windows, but the windows cannot be dragged from one pipe to another. The main reason is that the graphics pipes are separate logical units and are handled by an X™ server as

different, unconnected screens. This means that the X server does not provide any functionality to group multiple screens into a single logical display. A second reason is that you can directly connect OpenGL® to a specified graphics pipe and bypass the X protocol layer.

In the past, displaying an application on multiple screens required you to explicitly write the application for that purpose. You had to use tools like the OpenGL Performer™ or OpenGL Multipipe SDK™ libraries to help you create these multipipe applications. These tools allow you to explicitly open windows on different screens and to draw into them using OpenGL. However, this solution lacks consistency. In fact, all of the windows on the different pipes are independent; hence, moving or iconifying one window on one screen will not handle the other windows accordingly.

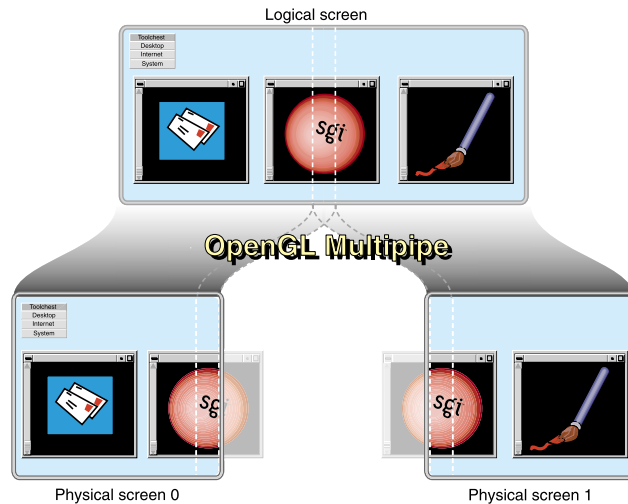
OpenGL Multipipe has been designed to overcome these difficulties. The goal is to group all pipes managed by the X server in order to create a consistent, single virtual screen as shown in Figure 1. This means that the application clients are not aware of the underlying hardware configuration. Rather, they only know about a single display and behave accordingly.



**Figure 1** OpenGL Multipipe with Non-Overlapping Screens

In contrast to Figure 1, if you have screens that overlap each other (like in an SGI Reality Center wall display with edge blending),

OpenGL Multipipe allows you to specify the amount of this overlap. Figure 2 shows the image blended on overlapping screens.



**Figure 2 OpenGL Multipipe with Overlapping Screens**

## Architecture

OpenGL Multipipe consists of two components:

- X server with the Xinerama™ extension
- Transparent OpenGL layer

### X Server with the Xinerama Extension

The Xinerama extension to the X server, introduced by X11R6.4, displays multiple physical screens of an X server as a single logical screen. The underlying physical screens are all managed by the same X server. The X server multicasts client requests to all of the underlying screens.

The Xinerama extension allows a multiheaded system (a system with multiple graphics pipes) to operate all managed screens as a single large screen. To enable Xinerama, the X server must be restarted with a special command-line argument.

If needed, multipipe-aware applications can bypass the Xinerama layer if the user sets an environment variable.

The logical position of each underlying physical screen is configurable.

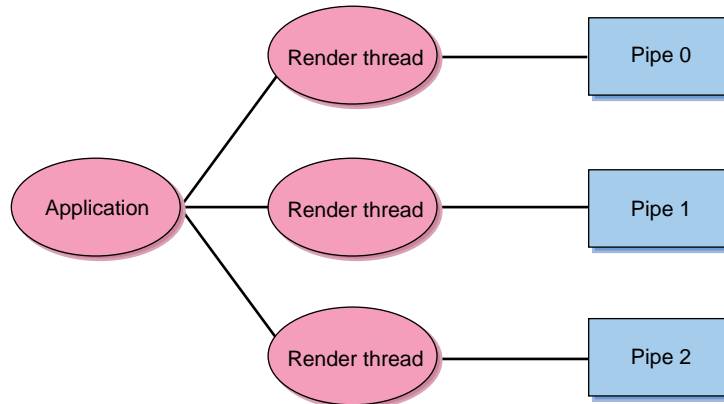
The extension is implemented in the device-independent level of the X Window System™ server. The Xinerama extension substitutes the X server dispatch calls with its own dispatch routines. These routines in turn call the original routine for each screen, substituting the per-screen resources as required.

Applications can also use the Xinerama extension protocol to query details about the underlying screens.

### Transparent OpenGL Layer

Xinerama allows only pure X applications to run transparently. The Transparent OpenGL layer allows OpenGL applications to take advantage of multiple pipes transparently. This layer ships as a DSO [dynamic shared object] and intercepts all OpenGL and GLX™ commands to the system. The transparent OpenGL layer creates underlying multiple threads, one per pipe, as shown in

Figure 3. It also synchronizes these threads when required.



**Figure 3** Running Single-Pipe Applications with OpenGL Multipipe

## Summary

OpenGL Multipipe allows single-pipe applications to run in a multipipe environment without modification. The OpenGL Multipipe layer results in a larger “total” framebuffer size. However, it adds some overhead to these applications. Some X extensions and resources are inherently attached to a single screen. In such cases, the X server may not allow these resources to run transparently over multiple pipes.

For maximum performance, users may want to run their own custom applications. Such multipipe-aware custom applications can run concurrently with transparent single-pipe applications.

For more information, interested readers are welcome to subscribe to the OpenGL Multipipe mailing list:

*[ogl\\_multipipe@els.sgi.com](mailto:ogl_multipipe@els.sgi.com)*.



**Corporate Office**  
1600 Amphitheatre Pkwy.  
Mountain View, CA 94043  
[650] 960-1980  
[www.sgi.com](http://www.sgi.com)

North America [800] 800-7441  
Latin America [650] 933-4637  
Europe [44] 118.925.75.00  
Japan [81] 3.5488.1811  
Asia Pacific [65] 771.0290

©2000–2002 Silicon Graphics, Inc. All rights reserved. Specifications subject to change without notice. Silicon Graphics, SGI, the SGI logo, Onyx, and OpenGL are registered trademarks, and GLX, Open Inventor, OpenGL Multipipe, OpenGL Multipipe SDK, Performer, and Reality Center are trademarks of Silicon Graphics, Inc. Netscape is a trademark of Netscape Communications Corporation. X, X Window System, and Xinerama are trademarks of The Open Group.

Document Number 007-4416-001