



Defence Research and
Development Canada

Recherche et développement
pour la défense Canada



Specialty configurations for the SGI Onyx4

Configuring graphics and applications for VR

*R. Carbone
DRDC Valcartier*

Defence R&D Canada – Valcartier

Technical Note

DRDC Valcartier TN 2006-730

January 2007

Canada

Specialty configurations for the SGI Onyx4

Configuring graphics and applications for VR

R. Carbone
DRDC Valcartier

Defence R&D Canada – Valcartier

Technical Note

DRDC Valcartier TN 2006-730

January 2007

Principal Author

Richard Carbone
Programmer/Analyst

Approved by

Yves van Chestein
Head/Knowledge Information Management Section

© Her Majesty the Queen as represented by the Minister of National Defence, 2007

© Sa Majesté la Reine, représentée par le ministre de la Défense nationale, 2007

Abstract

During the construction of the new virtual reality (VR) facility at DRDC Valcartier, it was determined early on that this facility would require one key component to drive it, the SGI Onyx4 visualization server. However, this computer system required certain specialized configurations in order for it to function correctly within this environment. Consequently, this required the reconfiguration of certain graphics-based configuration files that controlled the system's high-performance graphics subsystem. The SGI's role in the VR environment is to provide near real-time simulation-based capabilities for various high-end 3D models by leveraging the system's integrated hardware and software subsystems. Furthermore, the main applications and toolkits provided with the system had to be fine-tuned to allow for the high level of performance required. This document, therefore, describes all the necessary system configuration and administration-related information required for configuring the system's hardware and software subsystems that will be required for powering the VR environment.

This page intentionally left blank.

Executive summary

Specialty configurations for the SGI Onyx4: Configuring graphics and applications for VR

Carbone, R.; DRDC Valcartier TN 2006-730; Defence R&D Canada – Valcartier; January 2007.

The purpose of this document is to provide an all-in-one reference for system administration-related information required for configuring the SGI Onyx4 visualization server's hardware and software subsystems. Due to the system's complexity, it was considered imperative that this information be made available to others who at some point may be required to carry out the system's maintenance and administration-related tasks.

The virtual reality (VR) facility is, to date, only partially finished and is not scheduled for full completion until mid-2007. However, before construction of the facility is finalized the Onyx4 must be brought online in order to begin interoperability and interconnectivity tests between itself and its surrounding environment. This environment includes equipment such as projectors and displays, audio systems, 3D Stereo, synchronization devices, and motion trackers. The computer system has been ready for interoperability tests since Spring 2006, when the various toolkits and libraries were installed and configured. The computer system's functionality, driven by its hardware, is controlled by its various graphics settings managed through its configuration files. The system's capabilities will be used to provide near real-time simulations of high-end 3D models and scenegraphs that can work with a variety of display configurations. This document, therefore, provides an in-depth examination of the various configurations and background information necessary for exploiting the Onyx4 visualization server in a VR-based environment.

The Onyx4 visualization server, via its hardware and software, offers many powerful capabilities and features such as OpenGL Multipipe and OpenGL Performer, a hardware graphics compositor, four high-powered dual-channel high-refresh rate graphics pipes, and a high-memory bandwidth parallel computing architecture. The results are a computer system capable of delivering the necessary computational power required to drive a VR environment.

This page intentionally left blank.

Table of contents

Abstract	i
Executive summary	iii
Table of contents	v
List of figures	vii
List of tables	viii
1. Introduction.....	1
1.1 Objective	1
1.2 Introduction	1
1.3 Caveats and basic assumptions.....	1
2. System basics.....	3
2.1 Understanding XF86Config-4 configuration file basics.....	3
2.2 Connecting and understanding the graphics subsystems.....	4
2.3 Stopping and restarting graphics	5
2.4 Loading alternate configuration files.....	6
2.5 Troubleshooting the changing of configuration files	7
3. Modifying various configuration file settings.....	8
3.1 Introduction	8
3.2 Configuring pipes and channels	8
3.2.1 Channel cloning and desktop extension	10
3.2.2 Configuring various display types	12
3.3 Regenerating the XF86Config-4 configuration file.....	14
3.4 Configuring 3D Stereo and FSAA.....	14
3.5 Using the built-in resolutions	16
3.6 Understanding how to use and create custom Modelines.....	18
3.7 Configuring various monitor layouts.....	22
3.7.1 Examples	23
3.7.1.1 Example 1: The horizontal line display configuration.....	23
3.7.1.2 Example 2: The square display configuration	24
3.7.1.3 Example 3: The CAVE display configuration.....	26
3.7.2 Using coordinates.....	27
3.7.3 Final notes.....	28
3.8 Filenames and storage locations	28
3.9 Resolutions and refresh rates for currently configured display devices	30
4. Configuring applications, scripts, and system parameters	32
4.1 Introduction	32
4.2 Required system variables.....	32

4.3	Multipipe and DMX	33
4.3.1	Multipipe, DMX, Performer, and X specifics	33
4.3.2	Understanding DMX configuration files.....	34
4.3.3	Configuring various DMX configuration files.....	36
4.3.3.1	Example 1: The horizontal line display configuration.....	36
4.3.3.2	Example 2: The square display configuration	37
4.3.3.3	Example 3: The CAVE display configuration.....	37
4.3.4	Starting the virtual desktop	38
4.3.5	Running other DWM's.....	38
4.3.6	Running applications under Multipipe and DMX.....	39
4.4	Performer based applications	40
4.4.1	About Performer and Perfly	40
4.4.2	Configuring Perfly and the visualization scripts	40
4.4.2.1	Example 1: Perfly and the horizontal line display configuration.....	41
4.4.2.2	Example 2: Perfly and the square display configuration	41
4.4.2.3	Example 3: Perfly and the CAVE display configuration.....	42
4.4.3	Using the hardware graphics compositor with Perfly	42
5.	Conclusion	44
	References	45
	Annex A Example XF86Config-4 configuration files.....	47
A.1	The system default XF86Config-4.....	47
A.2	XF86Config-4 CAVE configuration file	65
A.3	XF86Config-4 2x2 ViewSonic VP2290b configuration file	84
A.4	XF86Config-4 hardware graphics compositor configuration file	101
	List of symbols/abbreviations/acronyms/initialisms	120
	Glossary.....	121

List of figures

Figure 1. Displays arranged in a straight line	23
Figure 2. Displays arranged as a square	25
Figure 3. Displays arranged in a CAVE configuration.....	26

List of tables

Table 1. Description of the various XF86Config-4 user-configurable sections	3
Table 2. DesktopSetup values and their significance	11
Table 3. Value breakdown for MonitorLayout.....	12
Table 4. Pre-calculated Modelines for various resolution/refresh rate combination	21
Table 5. Display positions in a straight line	24
Table 6. Display positions in a square configuration	25
Table 7. Display positions in CAVE configuration.....	26
Table 8. Directory locations and their purpose.....	28
Table 9. Display resolution and refresh rate combinations vs. display type (primary and secondary) for existing configurations according to signal type.....	30

1. Introduction

1.1 Objective

The objective of this document is to serve both as a reference to system administrators and users alike. Due to the complexity of the configurations made to the [Onyx4](#) visualization server, it was deemed important to document them so that others, if it ever became necessary, have the required information readily available in order to implement a variety of graphics-based changes to the system allowing for many different configurations. Throughout this document, it is therefore possible to learn and understand how to configure the system's graphics hardware and software related subsystems all from one useful reference.

1.2 Introduction

It is because the [SGI Onyx4](#) is such a complex system to configure and administrate (as most commercial [UNIX](#) systems are), that this document has been written. Although the system has already been preconfigured by its system administrator, it is important to document what has been done for the benefit of others, so that when (and if) the need arises where this information becomes necessary to implement new changes, the information is already available. Although there are many high quality documents and references available from [SGI's Techpubs Library](#), they are not always sufficient due to their vagueness or excessively technical language.

The [Onyx4](#) visualization server offers many powerful and complex hardware and software features. The various components examined in this document are [X Windows](#) and its configuration, [OpenGL Multipipe](#), [OpenGL Performer](#), the hardware graphics [compositor](#), virtual desktops, and extended and cloned displays.

[X Windows](#) provides the necessary graphical environment and framework necessary for providing basic graphics while the [OpenGL Multipipe](#) and [OpenGL Performer](#) provide [X Windows](#) with newer and extended capabilities. More specifically, [OpenGL Multipipe](#) is used to manage both the system's hardware graphics [compositor](#) and virtual desktops. Subsequently, [OpenGL Performer](#) supports an advanced visualization tool, [Perfly](#), which is used to power the system's high-end 3D models and scenegraphs. Furthermore, the configuration of 3D Stereo, Full Scene Anti-Aliasing (FSAA), graphics pipe and channel configurations are also discussed in subsequent sections of this report and are managed from one configuration file. Finally, an examination of the various displays and projectors that are configured for use with the system, including the VR facility's projectors and the [ViewSonic VP2290b](#) LCD are also examined.

1.3 Caveats and basic assumptions

This document is by no means an extensive tutorial. It is expected that the reader already has some basic understanding of computer graphics, [OpenGL](#), and the [UNIX](#) operating system. The user(s) or system administrator should also be familiar with the [X Windows](#) environment, and its familiarity is assumed throughout this text. A background in system administration is preferable,

as this will aid in comprehending the material presented throughout this document. This document's task is not to prepare users for general system administration.

This text should not be construed as a tutorial for interconnecting displays, cables, or other devices, including the addition or removal of subsequent synchronization signals. This text also cannot serve as an all-encompassing guide to the deployment and exploitation of the hardware graphics [compositor](#) as it far too complex a device and requires a very in-depth coverage for fully exploiting its capabilities. Instead, direct exploitation of this device using [OpenGL Performer](#) (using [Perfly](#)) is examined.

There are many complex and highly technical reference and guides available from [SGI](#), including Man files. All of these sources of information are considered important reading material to fill in any gaps not directly addressed by this document. This document's [References](#) section is an excellent starting point to learning more about these various additional sources of information.

2. System basics

2.1 Understanding XF86Config-4 configuration file basics

The file *XF86Config-4* is the [X Windows](#)-specific configuration file found for all systems running older versions of the [XFree86 X Windows](#) graphical system. Very recent versions of [XFree86](#) use the *xorg.conf* configuration file instead. The storage locations of either of these configuration files may vary from operating system to operating system and distribution to distribution. [XFree86](#) is commonly found in [Linux](#) and [BSD](#) operating systems. However, with the [Onyx4](#) system, [SGI](#) engineers chose to take advantage of a high quality [X](#) windowing system that was both freely available and completely compatible with the COTS graphics hardware implemented throughout the system, leveraging on its multiplatform capabilities. The [Onyx4](#) was a dramatic shift away from [SGI](#)'s proprietary graphics technology, replacing it with consumer driven graphics components offering powerful and high quality graphics capabilities at only a fraction of the regular cost.

Thus, the *XF86Config-4* configuration file provides a means for defining the various runtime and environmental parameters used by [X Windows](#) during its start-up that will determine its behaviour throughout the session. This configuration file defines the locations of the various video drivers, display and video resolution, I/O devices, colour depth, hardware acceleration, etc. In short, the overall behaviour and appearance of the [X Windows](#) system, with the exception of the desktop manager are loaded from this file.

Due to the importance of the configuration file, it is vital to understand its various sections. The file contains multiple sections that are examined below. Only modifiable sections that are of used for the VR facility are examined in-depth in Section [3](#). The following are the various sections that can be found with [SGI](#)'s *XF86Config-4* configuration file [\[5\]](#), and are as follows:

Table 1. Description of the various XF86Config-4 user-configurable sections

<u>Section Name</u>	<u>Description</u>
Files	This section defines the pathnames to various files such as fonts and RGB files.
ServerFlags	This section contains X server-specific run-time parameters.
Module	This section specifies runtime modules such as OpenGL , fonts, video acceleration options and other drivers that are to be loaded when X starts up.
InputDevice	This section defines the keyboard and mouse I/O devices.

Device	This section defines the attributes of a specific graphics device including multi-head, 3D Stereo, and FSAA.
Monitor	This section is used to define monitor resolutions and refresh rates.
Screen	This section defines screen-specific options such as resolution and colour depth.
ServerLayout	This section defines how the displays are to be positioned with respect to the other displays.
DRI	This section is used to define information necessary for Direct Rendering Infrastructure (DRI). This section is not normally modified.

Other possible sections such as *Modes*, *Vendor*, and *VideoAdapter* are not used by the [Onyx4](#)'s *XF86Config-4* configuration file, and thus are not included in the above table. However, their usage and implementation can vary from operating system to operating system.

Sections normally start with a *Section* tag (i.e. Section "Screen") and end with an *EndSection* tag found on a different line further below. Sections can have embedded subsections and they are denoted by a *SubSection* tag that ends with an *EndSubSection* tag.

Certain sections will require modification while others do not. The modifications implemented will be dependent on the functionality desired. Certain changes in functionality may require minor modifications to one section, while other functionality changes may require extensive modifications to multiple sections of the configuration file.

In this document, most of the modifications are rather simple to implement as the default *XF86Config-4* configuration file that ships with the [Onyx4](#) is suitable for a variety of uses. Thus, modifications examined throughout this document are considered as tweaks (or improvements) on the default settings.

2.2 Connecting and understanding the graphics subsystems

Before attempting to connect or change the graphics subsystem, all possible requirements should be determined prior to implementing any modifications whatsoever. In this way, any changes to be made can be done more effectively and logically, thus reducing the need to fix potential errors. Furthermore, once certain changes are implemented they can be difficult to remove afterwards. For example, prior to any modification, it is important to determine the number of displays that will be required in order to define the graphics pipes and channels that will be needed.

The system is capable of driving up to eight simultaneous displays. However, there are some caveats with the graphics subsystem. Before examining this in more detail, it is important to understand some of the basics about the system's graphics subsystem. Each graphics adapter (or graphics device or video card), is commonly referred to in the high-end visualization domain as a graphics pipe (or just "pipe"). Each of the pipes has two heads, making each device dual-head, and each head is commonly referred to as a channel. Thus, 2 channels per pipe x 4 pipes yields 8 possible displays. Furthermore, each pipe, including its channels, can use either a digital or an analog signal. Each channel's signal type can be configured independently of the other. Thus, it is important to remember when connecting any graphics device to any of the various channels that the connecting device be capable of supporting the required signal type, otherwise damage could result to either the display, channel, or pipe.

Analog video signals are most commonly used by CRT's and many low-cost LCD's. However, many of today's high performance displays will only accept digital signals. The main implementation issue surrounding the use of digital video signals is that the length of the video cable is usually limited to 10 feet or less. It is possible to go beyond this limit; however, this requires the implementation of custom made fibre optic cables. This also requires specialty adapters, and the entire solution is very expensive, approximately several thousand dollars per cable and adapter. Conversely, an analog signal can travel for up to several hundred feet without the need for any signal repeaters or specialized fibre optic cable technology. However, the cable length will depend largely on the quality of the cable. Generally, the higher the quality of the cable, the thicker both it and its shielding must be.

Connecting a given cable to a display is not generally a complex or difficult task. However, each display must be connected to an appropriate signal type. All of the system's channels are DVI-I, thus connecting an analog display will require either a DVI-I-to-VGA adapter or a DVI-I-to-13W3 adapter, depending on the display type. Furthermore, it is imperative that high quality adapters are used in order to obtain the best possible image.

It is important to understand that the graphics configuration file defines the system in terms of left and right channels. The left-handed channel of any pipe is channel 0 and the right-handed one is channel 1. Pipes are commonly referred to as pipes 0, 1, 2, and 3, depending on their positions vis-à-vis the PCI bus. Normally, the bottom-most pipe is pipe 0, and it follows sequentially from there moving bottom-up.

There are caveats with the system's graphics performance with respect to the number of channels used. Using both channels on any given pipe will affect that pipe's overall performance; therefore, dual-channel usage should be limited to specific needs and requirements (see Section [3.2.1](#)). When it is not needed, it should not be enabled in order to increase performance.

2.3 Stopping and restarting graphics

The graphics subsystem is a combination of graphics hardware and software. The software driving the system's graphics is [X Windows](#), an intuitive windowing desktop environment suitable for high-performance visualization. The graphics system is normally started at system boot-up during the final phases of system initialization. However, if it needs to be shutdown or

started-up again at any other time, it can only be done by the root user. The commands to start and stop graphics [1] are as follows:

```
$ /usr/gfx/startgfx
```

```
$ /usr/gfx/stopgfx
```

These commands should never be executed individually from within [X Windows](#). Running these commands, as they are within an [Xterm](#) will cause the windowing system to crash and could result in a kernel panic and coredump. Instead, these commands should only be executed from a command line shell as provided by a TTY-connected console or from a remote *Telnet* session. However, these commands can be combined into a shell script that can be run under [X Windows](#). This file, called *start-stop.sh* should be placed in the system directory */usr/sbin*. Its contents are as follows:

```
#!/usr/bin/sh
/usr/gfx/stopgfx
/usr/gfx/startgfx
```

Using this script, it is now possible to restart graphics from within the windowing system. However, in order to do so, it must be run as a background task in order to pass control back to [X](#). Commands are placed in the background as follows:

```
$ /usr/sbin/startstop.sh &
```

The symbol **&** places the command in the background and enables the command to stop and restart graphics without losing control of it. However, it is never suggested to stop and restart graphics while other users are logged in to [X Windows](#) as they will be immediately and without warning be logged off and all unsaved work will be lost.

2.4 Loading alternate configuration files

It is always considered a best practice that multiple *XF86Config-4* configuration files be used to store various configurations that may be used at any given time. It is not reasonable to assume that the system administrator will be able to properly document or remember all of the various parameters that can be used from one requirement to the next. Therefore, as requirements change over time, more configuration files should be created, each defining a certain set of displays and display-based options such as 3D Stereo, FSAA, resolutions, and refresh rates. The system is currently configured with many such files, and when one specific file is needed, the [X](#) server is stopped, the required configuration file is copied over the existing configuration file, and the [X](#) server is restarted.

For example, consider configuration file */etc/X11/new/new-config-4*. This file stores a specific graphics configuration that is required for a particular graphics setup. This can be done using the following commands (as the root user):

```
$ /usr/gfx/stopgfx
```

```
$ cp /etc/X11/new/new-config-4 /etc/X11/XF86Config-4
```

```
$ /usr/gfx/gfxinit
```

```
$ /usr/gfx/startgfx
```

In so doing, it possible to replace the configuration files as many times as necessary without ever having to reboot the system. However, these commands must always be used, only replacing the above filenames with the appropriate references.

2.5 Troubleshooting the changing of configuration files

On occasion, even after a valid configuration file has been replaced by another one, the old graphics and configuration states remains active despite having been replaced and a stopping/restarting of the graphics. This can occur because the system did not correctly resynchronize graphics state-based information according to the new configuration file. This can be fixed manually, as the root user, by deleting several files and then stopping, reinitializing, and restarting graphics, as shown below using the following commands:

```
$ cd /var/X11/Xvc
```

```
$ rm SG2_0_TimingTable
```

```
$ rm SG2_1_TimingTable
```

```
$ rm SG2_2_TimingTable
```

```
$ rm SG2_3_TimingTable
```

```
$ cd /usr/gfx/
```

```
$ ./stopgfx
```

```
$ ./gfxinit
```

```
$ ./startgfx
```

After several seconds, the [X](#) server should have picked up the new configuration and create the necessary new state files in `/var/X11/Xvc/`. If this does not work, then re-delete the aforementioned files and reboot the system. This will flush all dynamic system states and buffers relating to graphics that may be residing in kernel space memory. The aforementioned commands can also be accomplished using various scripts rather than performed manually.

3. Modifying various configuration file settings

3.1 Introduction

In this section, various issues directly relating to the proper configuration of the *XF86Config-4* configuration file are explored. The objective is to demonstrate through basic theory and simple examples how various components of the configuration file are correctly configured. Although it is always possible to use GUI-based tools to make the necessary changes to graphics subsystem, little understanding is gained in this manner, and when these tools fail to work, the configuration file must be edited manually.

The first item to be examined is the configuration of various pipes and channels. Specifically, the enabling and disabling of pipes and channels are examined in some detail. This also takes into account the concepts of cloning and desktop extension, as well as examining the various display-based configurations.

Once these basics are understood, enabling 3D Stereo graphics and FSAA are reviewed. Then the information on modifying standard display resolutions and Modelines are examined. The creation and implementation of customized Modelines are explored, as are the physical layout and implementation of the various displays.

In general, creating custom a graphics configuration is a relatively simple operation, once the necessary background information is correctly understood. Furthermore, while many configurations have already been created for ease of use, many more useful and workable combinations have yet to be created. Enhancements and modifications to the current configurations can also be made.

3.2 Configuring pipes and channels

Once the number of required displays has been determined, it is important to decide which pipes and channels to use. To maximize graphics throughput and performance, it is suggested that only one channel per pipe be used. To further increase performance, the hardware graphics [compositor](#) can also be used; however, its use will limit the overall usable desktop size to 1920x1200 pixels, and it will not allow for multiple concurrent desktops, extended desktops, or cloned desktops (the latter two are examined in more detail in sections [3.2.1](#) and [3.2.2](#)). Therefore, based on the physical layout of the pipes and channels, it is possible to define the actual configuration of the *XF86Config-4* configuration file. The information presented in this section is based on information found in [\[1\]](#) and Annex [A.1](#).

The example shown below presents two different sections from a valid configuration file. Here, two pipes are used, pipes 0 and 1. These sections are the [Device](#) and [Screen](#) sections. They both contain the same basic information; however, the former is specific to the hardware devices and the latter is specific to the displays that are connected to those devices. These sections are as follows:

Device section:

```
Section "Device"
  Identifier    "SGI SG-0"
  Driver       "fglrx"
  BusId        "PCI:2:0:0"
# === Screen Management ===
  Option "DesktopSetup"    "0x00000000"
  Option "MonitorLayout"   "AUTO, AUTO"
  . . .
EndSection

Section "Device"
  Identifier    "SGI SG-1"
  Driver       "fglrx"
  BusId        "PCI:2:1:0"
# === Screen Management ===
  Option "DesktopSetup"    "0x00000000"
  Option "MonitorLayout"   "AUTO, AUTO"
  . . .
EndSection
```

Screen section:

```
Section "Screen"
  Identifier    "Screen SG-0"
  Device       "SGI SG-0"
  Monitor      "Generic Monitor"
  DefaultDepth 24
  SubSection   "Display"
    Depth      24
    Modes      "1280x1024_60"
  EndSubSection
EndSection

Section "Screen"
  Identifier    "Screen SG-1"
  Device       "SGI SG-1"
  Monitor      "Generic Monitor"
  DefaultDepth 24
  SubSection   "Display"
    Depth      24
    Modes      "1280x1024_60"
  EndSubSection
EndSection
```

It is important to understand the function of the various lines presented above. Only the portions of immediate interest are actually shown. The irrelevant portions are represented by “. . .” and

can be looked in Annex [A.1](#). Starting with the *Device* section, the beginning of each physical device's defining section is denoted by the tag *Section "Device"*. The physical device itself is denoted using a virtual name with tag *Identifier*. The virtual name of the first physical device is *SGL SG-0*. The [X Windows](#) system communicates with the physical device through this *Identifier* tag. The actual graphics device driver, *fglrx*, is identified by the tag *Driver*. The actual physical location of the physical graphics pipe is described in relation to its actual position on system's PCI bus. Each device's physical location is identified by the tag *BusId*. Therefore, if tag *BusId* has a value of *PCI:2:0:0*, this indicates that graphics *pipe 0* is on PCI bus 2. The graphics device itself is represented by *0* (for *pipe 0*), and the primary channel of the device is represented by the second value *0*. For example, *pipe 1/channel 0* is denoted by *PCI:2:1:0*, as is seen in the second *Device* section above.

The tags *DesktopSetup* and *MonitorLayout* are examined in more detail in sections [3.2.1](#) and [3.2.2](#), respectively. They describe display cloning and desktop extension.

The next part of the configuration to examine is the *Screen* section. Each physical graphics pipe is associated with a *Screen* section that can be identified with a tag *Section "Screen"*. These sections state the basic display configuration that will be used for each corresponding graphics pipe. Furthermore, these sections are not designed to specify the number of channels that will be used per pipe; instead, this is configured in a screen's corresponding *Device* section (see sections [3.2.1](#) and [3.2.2](#)). A given *Screen* section assumes that both channels, whether they are used or not, must be both uniform in terms of their resolution and refresh rates.

The *Identifier* value for the *Screen* sections are not the same values as those found in the *Device* sections. Here, the *Identifier* value corresponds to the virtual names of the displays and not that of the physical graphics devices. However, the communication link between the display and the physical device are made by using the values *SGL SG-0* and *SGL SG-1*. For the two *Device* sections, the communication link is provided by its *Identifier* tags. For the *Screen* sections, this link is provided by the *Device* tags.

The *Monitor* tag is used only for generic purposes; there is no real need to change its value. The *DefaultDepth* tag specifies the default colour depth, which is 24-bits. The *"Display"* subsection is denoted by the tag *SubSection*. Its tags *Depth* and *Modes* are self-explanatory. The former specifies the colour depth and the latter specifies the resolution and refresh rate. Incidentally, the default system resolution and refresh rate is 1280x1020 @ 60 Hz.

As a final note for this section, the number of *Device* sections must be in accord with the number of *Screen* sections. Doing otherwise will cause [X Windows](#) to fail during start-up. It is also important to understand that not all the pipes must be enabled. Any unused pipes can simply have their corresponding *Device* and *Screen* sections deleted from the configuration file.

3.2.1 Channel cloning and desktop extension

Channel cloning and desktop extension are configured in the *XF86Config-4* configuration file. The material presented in this section can be found in [\[1\]](#) and [\[4\]](#).

The values for the tag *DesktopSetup* found in the *Device* section are used to enable channel cloning and desktop extension. The cloning action occurs from the left channel to the right

channel. Furthermore, whatever occurs on the left-handed channel is copied exactly “as is” onto the right-handed channel. Desktop extension can also occur from left to right or top to bottom. The difference between cloning and desktop extension is that the latter actually extends the usable size of a pipe’s desktop rather than just redisplay the same image. However, there are caveats in using cloning and desktop extension. The caveats are as follows: if any pipe under the management of a given¹ X server has its second channel enabled then all the pipes also under that X server’s management have their second channels enabled. Furthermore, both channels for each corresponding pipe must be use the same resolution and refresh rate. However, each individual pipe can use a different resolution and refresh rate than the other pipes. There is also the performance-related aspect of cloning and desktop extension that is examined later on in this section. Finally, cloning and desktop extension cannot be both be used at the same time; for example, cloning is used on one pipe and the desktop extension is used on another pipe; they mutually exclusive.

The available values for the *DesktopSetup* tag are *0x00000000*, *0x00000100*, *0x00000200* and *0x00000300*. Their purposes are as follows: 1) to disabled dual-channel use (*0x00000000*); 2) to create an exact copy of the left-handed channel to the right-handed channel (*0x00000100*); 3) to create an extended or managed desktop (*0x00000200* and *0x00000300*). Desktop extension value *0x00000200* creates an extended desktop of two displays in the horizontal direction, from left to right, while the value *0x00000300* creates an extended desktop of two displays in the vertical direction, from top to bottom. These values can also be seen in the table below.

Table 2. DesktopSetup values and their significance

<u><i>DesktopSetup</i> value</u>	<u>Significance</u>
0x00000000	Cloning and desktop extension disabled.
0x00000100	Channel cloning from left to right.
0x00000200	Extend desktop horizontally from left to right across both channels.
0x00000300	Extend desktop vertically from top to bottom across both channels.

There are also performance considerations to take into account. Channel cloning and desktop extension will consume more graphics memory bandwidth as well as increased GPU processing. Therefore, when neither is required, it is best left deactivated.

There are many reasons for using cloning and desktop extension. Cloning is useful so that the left-handed channels can be piped out to one set of displays in one room, and the right-handed

¹ There can be multiple X servers running concurrently on the system, each managing, at a minimum, just one pipe. The X server manages physical devices (i.e. pipes) and not logical devices (i.e. channels).

channels piped out to a different set of displays in the same or different room. Cloning, however, does not give eight useful displays; there are always only four “real” displays. Desktop extension, on the other hand, can yield up to eight useful displays; however, the total number of independently managed desktops remains at four. Only the amount of useable workspace has increased, which now scales up to eight displays.

Consider the following example: using eight 24" [Sony/SGI](#) FD Trinitron Flat CRT’s, the total workspace can be very large when using one virtual desktop (see Section [4.3.2](#)). The CRT’s maximum usable resolution is 1920x1440 @ 75 Hz. This provides 2,764,800 pixels per display, and using eight displays yields a total workspace of 22,118,400 pixels. However, performance will be very significantly affected. No real-time visualization will be possible. Furthermore, combining these resolutions with 3D and FSAA will not work due to bandwidth limitations on the graphics pipes. Therefore, when choosing to work with large workspaces the refresh rates must go down to support the required increase in memory bandwidth. While the system has not been configured with such a configuration, it is easy to do once all of Section [3](#) has been read.

3.2.2 Configuring various display types

The types of displays that can be used on the system are configured from the *XF86Config-4* configuration file. Several options can be configured for the various types of displays. Information in this section has been obtained from [\[1\]](#) and [\[4\]](#). The possible values for a given display are expressed using the *MonitorLayout* tag found in the *Device* section. Valid *MonitorLayout* values are *NONE*, *TMDS*, *AUTO*, and *CRT*. It is important that each *MonitorLayout*’s values be defined for each channel, even if cloning or desktop extension is not used; doing otherwise could cause the [X](#) server to hang.

These four values are all interchangeable and their purpose is easy to understand. The value *NONE* indicates that no display is present and instructs the system to disable a particular channel. Even if a display were present, nothing would be displayed to it. Value *TMDS* instructs the system to enable the channel using a digital single for use with digital-only displays. Value *CRT* instructs the system to enable the channel using an analog signal for use with a variety of display types. Value *AUTO* instructs the system to auto-detect the display type first and then enable the appropriate channel type, or if no device were present, the channel would then be disabled. However, if the value *AUTO* is used then the device must be connected to the system prior to starting up the [X](#) server otherwise it will not be detected and left non-configured.

The following table shows all possible valid combinations for the four values that can be used, including a brief explanation:

Table 3. Value breakdown for *MonitorLayout*

<u>Values (channel 0, channel 1)</u>	<u>Explanation (channel 0, channel 1)</u>
“AUTO”, “AUTO”	Automatic detection, automatic detection

“AUTO”, “CRT”	Automatic detection, analog display
“AUTO”, “NONE”	Automatic detection, no display
“AUTO”, “TMDS”	Automatic detection, digital display
“CRT”, “AUTO”	Analog display, automatic detection
“CRT”, “CRT”	Analog display, analog display
“CRT”, “NONE”	Analog display, no display
“CRT”, “TMDS”	Analog display, digital display
“NONE”, “AUTO”	No display, automatic detection
“NONE”, “CRT”	No display, analog display
“NONE”, “NONE”	No display, no display
“NONE”, “TMDS”	No display, digital display
“TMDS”, “AUTO”	Digital display, automatic detection
“TMDS”, “CRT”	Digital display, analog display
“TMDS”, “NONE”	Digital display, no display
“TMDS”, “TMDS”	Digital display, digital display

The default values for the system’s default *XF86Config-4* configuration file for *MonitorLayout* is *AUTO*. However, if the system’s hardware graphics [compositor](#) is to be used, the channels connected to it must be using a digital signal (*TMDS or AUTO*); it will not accept an analog signal. Again, it is must be understood that for each physical pip that is used, each *Device* section in the configuration file must have both its channels defined, even if one of them is left deactivated.

3.3 Regenerating the XF86Config-4 configuration file

When a configuration no longer works or is corrupt due to poor configuration and a backup copy of the original configuration file is not readily available, it is possible to regenerate a **new** default *XF86Config-4* configuration file. The system supports regeneration of this file and provides a tool specifically for this purpose.

The tool is a shell script that is written using the [Perl](#) scripting language. It is found under */etc/X11* and its filename is *gen-XF86Config*. In order to use the script, all running [X Windows](#) desktops must be shut down and the offending *XF86Config-4* file must be deleted. The execution of the script must be performed by the root user and is as follows:

```
$ /usr/gfx/stopgfx          (if X Windows is not already shut down)
$ cd /etc/X11
$ rm XF86Config-4
# ./gen-XF6Config
```

Upon execution of these commands, a new configuration file will be generated. This file will be the same as the one found with the system when it was first shipped from the factory.

3.4 Configuring 3D Stereo and FSAA

Based on the information provided in [1], configuring 3D Stereo and FSAA is a relatively simple operation.

3D Stereo is configured in the *Device* section of the *XF86Config-4* configuration file. The required lines appear in the *QBS Management* portion of this section. By default, 3D Stereo is disabled. To enable it, the following lines must appear as follows:

```
# === QBS Management ===
Option "Stereo"          "1"
Option "StereoSyncEnable" "1"
```

Most standard user applications will not use or display in 3D. However, once the appropriate rendering or visualization application is used along with the required 3D Stereo goggles, the display image can be seen in full 3D Stereo. The use of 3D Stereo may require the additional use of an infrared synchronization signal if the goggles are not directly connected to the computer system. However, this is not examined in this document.

It is not required that all the pipes under a given [X](#) server's management are configured for 3D Stereo. In fact, this option can be applied to any or all pipes. However, it cannot be applied on a channel-by-channel basis. To disable a pipe from using 3D Stereo, the above lines can be set with values of *0* or the lines can be altogether deleted from the configuration file.

Using 3D Stereo requires increased memory bandwidth from a given graphics pipe, including increased GPU processing. Therefore, there will be a performance penalty for enabling it; how significant the penalty is depends on many factors such as resolution and refresh rate, cloning or desktop extension, and the use of FSAA. Depending on the type of application and textures used, experimentation will most likely be required to find an acceptable compromise between resolutions, refresh rates, and other potential factors.

Full Scene Anti-Aliasing, or FSAA as it sometimes known, is configured by changing or adding the following lines to the *FSAA Management* portion of the *Device* section of the configuration file. This is as follows:

```
# === FSAA Management ===
Option "FSAAScale"          "1"
Option "FSAADisableGamma"  "no"
Option "FSAACustomizeMSPos" "yes"
Option "FSAAMSPosX0"       "0.250000"
Option "FSAAMSPosY0"       "0.416666"
Option "FSAAMSPosX1"       "0.083333"
Option "FSAAMSPosY1"       "0.083333"
Option "FSAAMSPosX2"       "0.416666"
Option "FSAAMSPosY2"       "0.750000"
Option "FSAAMSPosX3"       "0.750000"
Option "FSAAMSPosY3"       "0.916666"
Option "FSAAMSPosX4"       "0.583333"
Option "FSAAMSPosY4"       "0.250000"
Option "FSAAMSPosX5"       "0.916666"
Option "FSAAMSPosY5"       "0.583333"
```

Generally, for FSAA, the only value that requires modification is the first line, *FSAAScale*. There are different valid values for this tag; they are *1*, *2*, *4*, and *6*. *1* is used only when FSAA is to be applied on a per-window basis. This means that when a windowed application needs to use FSAA, it can, but otherwise the rest of the desktop and open windows are not affected. However, when values *2*, *4*, and *6* are used, then “global” anti-aliasing is applied to all windows under a given desktop. The other values for this portion are best left unchanged, as they are “factory-set.”

As with 3D Stereo, FSAA can be applied to individual pipes and it need not be applied to all pipes. However, once applied, it applies to both channels. It cannot be enabled or disabled for a given channel. To disable FSAA, all the lines can be deleted in the *FSAA Management* portion of the *Device* section, or the tag *FSAAScale* can be set to *0*.

As with 3D Stereo, FSAA requires increased memory bandwidth and GPU processing. Thus, it is important to find a compromise between screen resolution and its refresh rate. Furthermore, if both FSAA and 3D Stereo are used, then even more calculations are required by the GPU and performance will degrade further. Moreover, if cloning or desktop extension are enabled (Section [3.2.1](#)), then this will result in further degradation. If all of these options are required, then it is important to find a workable set of parameters for resolution and refresh rate. Performance can be somewhat improved by deleting the line *FSAACustomizeMSPos* and all of the lines containing *FSAAMSPos*. This will result in a slightly degraded image, but with slightly more responsive

graphics. For reasons of performance, FSAA should only be enabled when very high-quality images are required, otherwise, from a reasonable distance, the human eye cannot easily distinguish between the two². It is further suggested to leave [X Windows](#) running until either 3D Stereo or FSAA is actually required. When needed, the appropriate *XF86Config-4* configuration file can then be loaded and applied.

3.5 Using the built-in resolutions

Configuring a display for running at a specific resolution and refresh rate is generally not very difficult to do. However, it does entail that the user or system administrator has determined which resolutions and refresh rates will be required. The default system setting is 1280x1024 @ 60 Hz; this is the factory and operating system default; however, this can be changed at any time. The standard *XF86Config-4* configuration file (including regenerated ones (see Section [3.3](#))) comes complete with many additional built-in resolutions and refresh rates. These built-in settings are examined in this section.

Resolutions and refresh rates are configured for a given pipe in its corresponding *Screen* section. This is the only location that directly specifies a display's configuration. More information can be obtained from [\[1\]](#) and [\[5\]](#), although information must be extrapolated from these two sources. An example *Screen* section is shown below:

```
Section "Screen"
    Identifier      "Screen SG-2"
    Device          "SGI SG-2"
    Monitor         "Generic Monitor"
    DefaultDepth    24
    SubSection      "Display"
        Depth      24
        Modes       "1280x1024_60"
    EndSubSection
EndSection
```

The *Modes* tag describes the actual resolution and refresh rate of the display that will be used across a pipe's two channels. The *Modes* value as seen above is 1280x1024 @ 60 Hz. It is not necessary to incorporate the refresh rate into the *Modes* value if it is specifically defined by a Modeline elsewhere in the configuration. However, if there are other similar Modeline entries in the configuration file, such as 1280x1024_75, then the specific refresh rate should be included in the *Modes* value.

Modelines are defined in the *Monitor* section of the configuration file. Unlike for *Device* and *Screen* sections, there can be only one *Monitor* section. This is because Modelines are grouped together because they are limited in their scope, restricted to specifying specific resolutions, refresh rates, and timings. *Device* and *Screen* sections, on the other hand, are device specific and thus correspond to a particular system device.

² An individual's ability to discern the difference between images displayed using FSAA and without it vary greatly based on distance, luminosity, and a variety of human-related factors.

Regardless of how a *Modes* value is written, the information for any specific *Modes* value must be supported by an existing *Modeline* within the configuration file. If a specific *Modes* value was set but there was no supporting *Modeline* then the [X Windows](#) start-up and initialization would fail. An example *Modeline* that would adequately support the *Modes* value in the example above is the following:

```
Modeline "1280x1024_60" 121.5 1280 1396 1540 1800 1024 1027 1030 1125
```

The exact meaning of this *Modeline* will be further examined in Section 3.6. For the mean time, it is sufficient to understand that each pipe can be configured independently of all the other pipes; no two pipes have to use the same *Modeline*, although they can. However, each channel must use the same *Modeline* as its parent pipe. As such, the pipe dictates the resolution and refresh rate for its graphics channels.

In the example below, assuming that two different *Modelines* exist in a given configuration file in the *Monitor* section and that the user or system administrator wishes to a one *Modeline* for one pipe and another *Modeline* for a different pipe, then this can be expressed as:

```
Section "Monitor"
```

```
...
```

```
Modeline "1280x1024_60" 121.5 1280 1396 1540 1800 1024 1027 1030 1125
```

```
Modeline "1600x1200_60" 162 1600 1664 1856 2160 1200 1201 1204 1250
```

```
...
```

```
Section "Screen"
```

```
Identifier "Screen SG-0"
```

```
Device "SGI SG-0"
```

```
Monitor "Generic Monitor"
```

```
DefaultDepth 24
```

```
SubSection "Display"
```

```
Depth 24
```

```
Modes "1280x1024_60"
```

```
EndSubSection
```

```
EndSection
```

```
Section "Screen"
```

```
Identifier "Screen SG-1"
```

```
Device "SGI SG-1"
```

```
Monitor "Generic Monitor"
```

```
DefaultDepth 24
```

```
SubSection "Display"
```

```
Depth 24
```

```
Modes "1600x1200_60"
```

```
EndSubSection
```

```
EndSection
```

Thus, pipe 0 (the first *Screen* section) will display at 1280x1024 @ 60 Hz while pipe 1 (the second *Screen* section) will display at 1600x1200 @ 60 Hz.

Configuring resolutions and refresh rates is relatively simple when the necessary Modelines already exist. However, the graphics subsystem can become unstable if unsupported resolution/refresh rate combinations are used, or if a given combination requires too much bandwidth and “chokes” the graphics pipe.

3.6 Understanding how to use and create custom Modelines

While configuring the *XF86Config-4* configuration file for use with an existing Modeline is a relatively simple operation, as seen in the previous section, it is not so simple if the Modeline must be created manually. In this section, the creation of customized Modelines is examined.

It is important to understand the system’s default Modelines before attempting to build newer or customized Modelines. Modelines are found in the *Monitor* section of the *XF86config-4* configuration file. Below is the system’s default Modelines that have been extracted from the configuration file. They have been sorted in order to improve readability. They are as follows:

1280x1024:

```
Modeline "1280x1024_50f" 104 1280 1336 1472 1664 1024 1027 1030 1250 -hsync -vsync
Modeline "1280x1024_5994f" 115.2 1280 1296 1440 1560 1024 1027 1030 1232 -hsync -vsync
Modeline "1280x1024_60f" 121.5 1280 1396 1540 1800 1024 1027 1030 1125 -hsync -vsync
ModeLine "1280x1024_85n" 157 1280 1344 1504 1728 1024 1025 1028 1072 -hsync -vsync
ModeLine "1280x1024_96" 164.966 1280 1300 1460 1600 1024 1026 1029 1074
ModeLine "1280x1024_120" 233.793 1280 1384 1528 1776 1024 1025 1028 1097
```

1600x1200:

```
ModeLine "1600x1200_60n" 162 1600 1664 1856 2160 1200 1201 1204 1250 -hsync -vsync
ModeLine "1600x1200_75n" 202 1600 1664 1856 2160 1200 1201 1204 1250 -hsync -vsync
ModeLine "1600x1200_85n" 229 1600 1664 1856 2160 1200 1201 1204 1250 -hsync -vsync
```

1920x1154:

```
Modeline "1920x1154_48f" 144 1920 1968 2160 2400 1154 1157 1160 1250 -hsync -vsync
Modeline "1920x1154_50f" 144 1920 1968 2160 2400 1154 1157 1160 1200 -hsync -vsync
```

```

Modeline "1920x1154_5994f" 192 1920 1992 2192 2464 1154 1157 1160 1300 -hsync -
vsync
Modeline "1920x1154_5994f_p" 156 1920 1968 2000 2200 1154 1157 1160 1283 -
hsync -vsync
Modeline "1920x1154_60f" 172.8 1920 1968 2160 2400 1154 1157 1160 1200 -hsync
-vsync
Modeline "1920x1154_60f_p" 162 1920 1968 2000 2160 1154 1157 1160 1250 -hsync -
vsync

```

1920x1200:

```

Modeline "1920x1200_50f" 150 1920 1968 2160 2400 1200 1203 1206 1250 -hsync -
vsync
Modeline "1920x1200_60f" 180 1920 1968 2160 2400 1200 1203 1206 1250 -hsync -
vsync
Modeline "1920x1200_60f_p" 162 1920 1968 2000 2160 1200 1203 1206 1250 -hsync -
vsync
Modeline "1920x1200_60n" 193 1920 2048 2256 2592 1200 1201 1203 1242 -
hsync -vsync
Modeline "1920x1200_60p" 157.39 1920 1950 2072 2112 1200 1203 1206 1242
Modeline "1920x1080_72n" 211 1920 2056 2264 2608 1080 1081 1084 1126 -
hsync -vsync
Modeline "1920x1200_75n" 246 1920 2064 2272 2624 1200 1201 1203 1253 -
hsync +vsync

```

The first item that should be noticed is the different letters associated with a resolution/refresh rate combination found with various Modelines. For example, consider 1920x1200_75n. The letter “n” signifies a special video output format. Not all of the letters are found in the default configuration file. According to [8], there are seven possible video output formats. Their significance is for the most part irrelevant since using a standard Modeline configuration and then applying the necessary modifications to the appropriate *Device*, *Screen*, and *ServerLayout* sections will provide the bulk of the functionality that will ever be required. In this way, when creating custom Modelines it is not necessary to worry about special video output formats. Furthermore, all of the system’s equipment, including all of the display devices outlined in Section 3.9, including the hardware graphics [compositor](#), genlocking, FSAA and 3D Stereo will all work correctly with any valid Modeline, regardless of any video output format information included in a given Modeline. [SGI](#) provides very limited documentation on these formats, and as such, they should be used sparingly, if at all.

Each Modeline is made up of several components strung together that form a descriptive line detailing a display’s capabilities [6] [7]. Each Modeline consists of a resolution, refresh rate, clock timing, and horizontal and vertical timings. An example will help to clarify this. Take for example the default resolution/refresh rate of 1280x1024 @ 60 Hz. Its *Modeline* is as follows (after removing the unnecessary video format type):

```

Modeline "1280x1024_60" 121.5 1280 1396 1540 1800 1024 1027 1030 1125 -hsync
-vsync

```

Although the following is not a valid Modeline, breaking it down into its constituent components aids in better understanding it:

<u>Resolution</u>	<u>Clock</u>	<u>Horizontal timing</u>	<u>Vertical timing</u>	<u>Flags</u>
1280x1024_60	121.5	1280 1396 1540 1800	1024 1027 1030 1125	-hsync -vsync

Another valid manner that Modelines can be written is as follows:

```
Mode "1600x1200"  
DotClock 121.5  
HTimings 1280 1396 1540 1800  
VTimings 1024 1027 1030 1125  
Hsync low  
Vsync high  
EndMode
```

Either *Modeline* expression is equally valid. However, the notation taken used [SGI](#) is the less descriptive one-lined Modeline.

Modelines can be calculated by hand. These techniques are explained in-depth in [7]; however, they can also be calculated using a variety of web-based tools that can perform all of the complex calculations automatically. All that is required as input in order to obtain a valid Modeline is a starting resolution and refresh rate. From this information, all of the other values can be calculated. Once the calculations are complete, it suffices simply to copy the Modeline into the *Monitor* section of the configuration file, and make any required changes to the appropriate *Screen* sections of the configuration file. Some of the various web-based Modeline calculators are:

<http://zaph.com/Modeline/Calculate.py?sec=804838>

<http://www.tkk.fi/Misc/Electronics/faq/vga2rgb/calc.html>

<http://www.sh.nu/nvidia/gtf.php>

<http://osdn.dl.sourceforge.net/sourceforge/gtf/gtf.c>³

Furthermore, when using custom Modelines, it is best to create them using one of the aforementioned web-based tools, as these tools will not include any video output formats unless if supported the appropriate option is specified. Moreover, when an existing Modeline uses a video output format, the format can either be directly stripped from the Modeline or from the *Modes* value found in a corresponding *Screen* section.

Provided below is a table that lists all of the pre-calculated Modelines supporting the various display devices already in use with the [Onyx4](#) visualization server (see Section 3.9). It is

³ This tool is an open source C command line-based Modeline calculator. The source code can be compiled on multiple platforms.

important to remember that not all Modelines can be used with any given display device. Many of these Modelines were compiled by the system administrator.

Table 4. Pre-calculated Modelines for various resolution/refresh rate combination

<u>Resolution @ Refresh rate</u>	<u>Modeline</u>
1280x1024@60	Modeline "1280x1024_60.00" 108.88 1280 1360 1496 1712 1024 1025 1028 1060 -HSync +Vsync
1280x1024@75	Modeline "1280x1024_75.00" 138.54 1280 1368 1504 1728 1024 1025 1028 1069 -HSync +Vsync
1280x1024@85	Modeline "1280x1024_85.00" 159.36 1280 1376 1512 1744 1024 1025 1028 1075 -HSync +Vsync
1280x1024@96	Modeline "1280x1024_96.00" 182.81 1280 1376 1520 1760 1024 1025 1028 1082 -HSync +Vsync
1400x1050@60	Modeline "1400x1050_60.00" 122.61 1400 1488 1640 1880 1050 1051 1054 1087 -HSync +Vsync
1400x1050@75	Modeline "1400x1050_75.00" 155.85 1400 1496 1648 1896 1050 1051 1054 1096 -HSync +Vsync
1400x1050@96	Modeline "1400x1050_96.00" 205.45 1400 1512 1664 1928 1050 1051 1054 1110 -HSync +Vsync
1400x1050@100	Modeline "1400x1050_100.00" 214.39 1400 1512 1664 1928 1050 1051 1054 1112 -HSync +Vsync
1600x1200@60	Modeline "1600x1200_60.00" 160.96 1600 1704 1880 2160 1200 1201 1204 1242 -HSync +Vsync
1600x1200@75	Modeline "1600x1200_75.00" 205.99 1600 1720 1896 2192 1200 1201 1204 1253 -HSync +Vsync
1600x1200@85	Modeline "1600x1200_85.00" 234.76 1600 1720 1896 2192 1200 1201 1204 1260 -HSync +Vsync

1920x1200@60	Modeline "1920x1200_65.00" 211.22 1920 2056 2264 2608 1200 1201 1204 1246 -HSync +Vsync
1920x1200@75	Modeline "1920x1200_75.00" 246.59 1920 2064 2272 2624 1200 1201 1204 1253 -HSync +Vsync
1920x1440@60	Modeline "1920x1440_60.00" 234.59 1920 2064 2272 2624 1440 1441 1444 1490 -HSync +Vsync
1920x1440@75	Modeline "1920x1440_75.00" 297.59 1920 2072 2280 2640 1440 1441 1444 1503 -HSync +Vsync
960x2400@30	Modeline "960x2400_30.00" 94.91 960 1024 1128 1296 2400 2401 2404 2441 -HSync +Vsync
960x2400@41	Modeline "960x2400_41.00" 132.11 960 1032 1136 1312 2400 2401 2404 2456 -HSync +Vsync
1920x1200@30	Modeline "1920x1200_30.00" 89.67 1920 1992 2184 2448 1200 1201 1204 1221 -HSync +Vsync
1920x1200@41	Modeline "1920x1200_41.00" 127.38 1920 2024 2224 2528 1200 1201 1204 1229 -HSync +Vsync
3840x2400@25	Modeline "3840x2400_25.00" 311.55 3840 4072 4480 5120 2400 2401 2404 2434 -HSync +Vsync

When creating customized *XF86Config-4* configuration files, it is suggested that each file should be specific to just one Modeline. This will help to simplify the creation of multiple customized Modelines. It is also suggested that configuration files be given names reflective of their contents. For example, consider the filename *XF86Config-4-1920x1440x75*. Furthermore, using multiple configuration files makes it easier to switch from one configuration to another. All that needs to be done is to copy a different configuration over the current one and restart [X Windows](#).

3.7 Configuring various monitor layouts

It is expected that any advanced windowing environment will offer the ability to reposition displays into various configurations to allow for varying visualization-based requirements. [X Windows](#) fully supports the logical and physical rearrangement of displays and provides a simple facility for configuring display layouts. Many common configurations are possible and include a horizontal arrangement of displays (a straight line); a vertical arrangement of displays (a vertical line); a square, as well as many other possibilities. One specialty configuration has also been

created; this configuration is referred throughout the rest of this text as the *CAVE* configuration; it mimics the actual layout of the projectors used within the VR facility for creating a CAVE-based environment.

Creating new display layout configurations is a straightforward process. The layout information is entered into the *XF86Config-4* configuration file's *ServerLayout* section. This section defines the juxtaposition of each display with relation to all of the other displays around it. Each display is configured in terms of where the other displays are positioned relative to it; other displays can be to the top, bottom, left, and right of a given display.

After reviewing the following examples, much will become clear. Each of the examples below is realistic and reflects commonly used display layouts. Furthermore, each example will include an illustrated diagram, a table, and line listing from each example's configuration file *ServerLayout* section. The diagrams are denoted by *Identifier* tags from their respective *Screen* sections found in their respective *XF86Config-4* configuration files.

3.7.1 Examples

Below are various examples that will help to better understand display layouts and can they can be arranged, created, and configured.

3.7.1.1 Example 1: The horizontal line display configuration

Looking at Figure 1, Table 5, and then the actual listing of lines from the configuration file for the example below, the implementation of a horizontal line of displays is created. This configuration is referred to as *4x1-display* configuration⁴. Its configuration file is found in Annex [A.1](#) and is as follows:

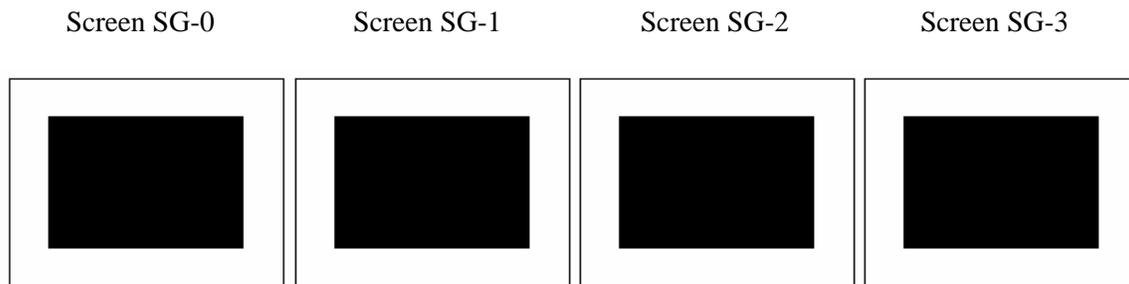


Figure 1. Displays arranged in a straight line

⁴ A *1x4-display* configuration would be a vertical line of displays.

Table 5. Display positions in a straight line

<u>Screen Name</u>	<u>Top</u>	<u>Bottom</u>	<u>Left</u>	<u>Right</u>
Screen SG-0	""	""	""	"Screen SG-1"
Screen SG-1	""	""	"Screen SG-0"	"Screen SG-2"
Screen SG-2	""	""	"Screen SG-1"	"Screen SG-3"
Screen SG-3	""	""	"Screen SG-2"	""

ServerLayout lines from *XF86Config-4*:

```

Screen "Screen SG-0"    ""    ""    ""    ""
Screen "Screen SG-1"    ""    ""    "Screen SG-0"    "Screen SG-2"
Screen "Screen SG-2"    ""    ""    "Screen SG-1"    "Screen SG-3"
Screen "Screen SG-3"    ""    ""    "Screen SG-2"    ""

```

3.7.1.2 Example 2: The square display configuration

Looking at Figure 2, Table 6, and then the actual listing of lines from the configuration file for the example below, a square configuration is created. This configuration is referred to as a *2x2-display* configuration. Its configuration file is found in Annex [A.2](#) and is as follows:

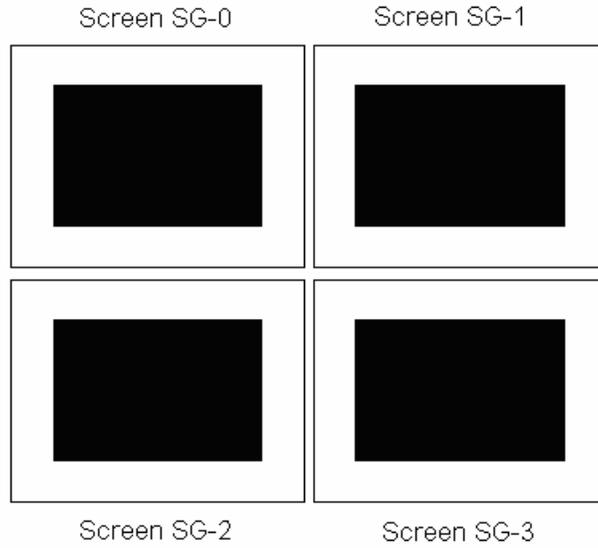


Figure 2. Displays arranged as a square

Table 6. Display positions in a square configuration

<u>Screen Name</u>	<u>Top</u>	<u>Bottom</u>	<u>Left</u>	<u>Right</u>
Screen SG-0	""	"Screen SG-2"	""	"Screen SG-1"
Screen SG-1	""	"Screen SG-3"	"Screen SG-0"	""
Screen SG-2	"Screen SG-0"	""	""	"Screen SG-3"
Screen SG-3	"Screen SG-1"	""	"Screen SG-2"	""

ServerLayout lines from *XF86Config-4*:

```
Screen "Screen SG-0" "" "" "Screen SG-2" "" "Screen SG-1"
Screen "Screen SG-1" "" "" "Screen SG-3" "Screen SG-0" ""
Screen "Screen SG-2" "Screen SG-0" "" "" "Screen SG-3"
Screen "Screen SG-3" "Screen SG-1" "" "" "Screen SG-2" ""
```

3.7.1.3 Example 3: The CAVE display configuration

Looking at Figure 3, Table 7, and then the actual listing of lines from the configuration file for the example below, a *CAVE* configuration is created. This configuration is referred to as a *3x1-display* configuration. Its configuration file is found in Annex [A.3](#) and is as follows:

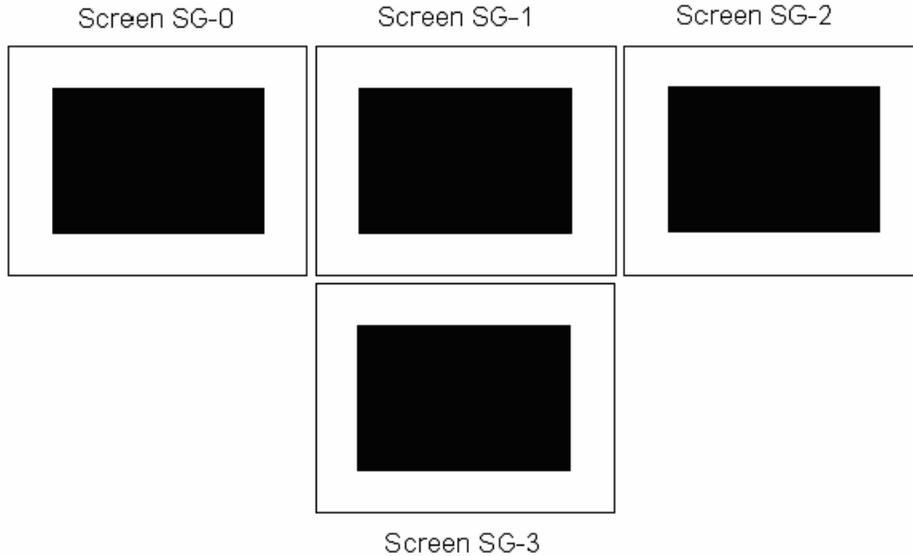


Figure 3. Displays arranged in a CAVE configuration

Table 7. Display positions in CAVE configuration

<u>Screen Name</u>	<u>Top</u>	<u>Bottom</u>	<u>Left</u>	<u>Right</u>
Screen SG-0	""	""	""	"Screen SG-1"
Screen SG-1	""	"Screen SG-3"	"Screen SG-0"	"Screen SG-2"
Screen SG-2	""	""	"Screen SG-1"	""
Screen SG-3	"Screen SG-1"	""	""	""

ServerLayout lines from *XF86Config-4*:

```
Screen "Screen SG-0" "" "" "" "Screen SG-1"
Screen "Screen SG-1" "" "Screen SG-3" "Screen SG-0" "Screen SG-2"
Screen "Screen SG-2" "" "" "Screen SG-1" ""
```

Screen "Screen SG-3" "Screen SG-1" "" "" ""

3.7.1.3.1 Examining the CAVE configuration

The first two examples are rather obvious. The third one is more complex; however, it too will appear simple once it is explained. In Example 3, looking at Figure 3 and Table 7, it is not so easy to envision the actual layout. The VR facility uses projection-based screens rather than physical displays. The screens are about 9 ft wide and 7 ft tall. Three screens are upright and the fourth screen is floor-mounted. When the system drives a VR simulation and all four screens are needed, the application used must understand this odd arrangement so that it can correctly display across all four screens.

According to Table 7, the leftmost screen is *Screen SG-0*. It has no screens to its left, top, or bottom, only *Screen SG-1* to its right. *Screen SG-1*, the middle-upright screen has *Screen SG-0* to its left, *Screen SG-2* to its right, and *Screen SG-3* to its bottom. *Screen SG-2* is the rightmost screen; it has no screens to its top, bottom, or right, only *Screen SG-1* to its left. Floor-mounted *Screen SG-3* has *Screen SG-1* to its top, none to its bottom, and none to its left or right.

3.7.1.3.2 Caveats

However, when running simulations it may be necessary to provide control information to the program understands the actual layout of the upright screens with respect to the floor-mounted screen. Depending on the visualization program utilized, the floor-mounted screen may not be supported. The problem is that most programs cannot handle non-standard display-based configurations such as the CAVE configuration because it does not understand how to adjust its rendering taking into account the floor-mounted screen. Generally, most application will simply treat the floor-mounted screen as the fourth screen in a 4x1-display configuration. In comparison, most applications can easily handle standard configurations such as a 2x2, 4x1, or 1x4-display based configurations.

There are advanced simulation programs, both commercial and open source in nature that do support [Performer](#)-based API's that will allow a developer to add certain extensions to a program so that it can correctly support any type of CAVE configuration. There are also commercially available applications that fully support CAVE configurations "out-of-the-box." However, these applications have not been purchased for the current VR facility and are not explored any further in this document.

3.7.2 Using coordinates

However, there may be instances where specific display positions need to be described using a more specific form of positioning using coordinate-based referencing [via x, y^5 coordinates]. There are two are types of coordinates, absolute and relative. Both types are specified in the same manner, differing only in their use of the tags *Relative* and *Absolute* in the *ServerLayout* Section. Their implementation is not examined in this document; however, it is important to know that it is

⁵ Cartesian coordinates where the upper left hand corner of *Screen SG-0* marks $0, 0$ on the coordinate grid.

possible to use coordinate-based referencing such that if the need arises the possibility exists. More information can be found in [5].

3.7.3 Final notes

Under other circumstances where more natural display configurations such as the horizontal line (4x1-display configuration), square (2x2-display configuration), or vertical line (1x4-display configuration) are used, simulation applications generally behave without the need for any modification. In building an appropriate display configuration, it is important to decide where one display will be in relation to the other displays. By using this approach to building display layouts, it is possible to support a large number of displays since each display's position is defined in terms of the other displays around it.

3.8 Filenames and storage locations

This section provides an overview of the storage locations where the various pre-configured *XF86Config-4* configuration files, scripts, and [Multipipe](#) configuration files (see Section 4.3) are located for the system. For users and system administrator, the following table will be of help in locating files more easily. The table provides the directory location where a set of related files can be found. It also provides a description detailing the contents of these directories to enable a faster looking up of files, scripts, and configurations.

Table 8. Directory locations and their purpose

<u>Directory location</u>	<u>Purpose</u>
/etc/X11/1280x1024x60	Stores file related 1280x1024 @ 60 Hz, including script.
/etc/X11/1280x1024x75	Stores file related 1280x1024 @ 75 Hz, including script.
/etc/X11/1280x1024x85	Stores file related 1280x1024 @ 85 Hz, including script.
/etc/X11/1280x1024x96	Stores file related 1280x1024 @ 96 Hz, including script.
/etc/X11/1400x1050x60	Stores file related 1400x1050 @ 60 Hz, including script.
/etc/X11/1400x1050x75	Stores file related 1400x1050 @ 75 Hz, including script.
/etc/X11/1400x1050x96	Stores file related 1400x1050 @ 96 Hz, including script.
/etc/X11/1400x1050x100	Stores file related 1400x1050 @ 100 Hz, including script.

/etc/X11/1600x1200x60	Stores file related 1600x1200 @ 60 Hz, including scripts.
/etc/X11/1600x1200x75	Stores file related 1600x1200 @ 75 Hz, including scripts.
/etc/X11/1600x1200x85	Stores file related 1600x1200 @ 85 Hz, including scripts.
/etc/X11/1920x1200x60	Stores file related 1920x1200 @ 60 Hz, including scripts.
/etc/X11/1920x1200x75	Stores file related 1920x1200 @ 75 Hz, including scripts.
/etc/X11/1920x1440x60	Stores file related 1920x1440 @ 60 Hz, including scripts.
/etc/X11/1920x1440x75	Stores file related 1920x1440 @ 75 Hz, including scripts.
/etc/X11/digital-compositor- XF86Config	Stores file related to the hardware graphics compositor <i>XF86Config-4</i> files with and without 3D Stereo and FSAA, running at: 1280x1024 @ 60 Hz; 1280x1024 @ 75 Hz; 1600x1200 @ 60 Hz, and 1920x1200 @ 60 Hz. This directory goes with directory <i>/etc/X11/set-compositor-scripts</i>).
/etc/X11/dmx	Stores DMX (Multipipe) configuration files for CRT, ViewSonic , CAVE, and hardware graphics compositor .
/etc/X11/docs	Stores simple text-based reference files for troubleshooting purposes.
/etc/X11/openflight	Stores OpenFlight scripts for running 3D demos, models, and scenegraphs.
/etc/X11/set-compositor- scripts	Stores scripts required for setting the hardware graphics compositor using various resolutions (goes together with directory <i>/etc/X11/digital-compositor-XF86Config</i>)
/etc/X11/VP2290b	Stores the files relating to the ViewSonic VP2290b LCD running at different resolutions and refresh rates with a variety of different pipe configurations.

3.9 Resolutions and refresh rates for currently configured display devices

This section provides a breakdown of currently available resolution/refresh rate configurations for the various displays and projection equipment that is currently configured for the system. This information is broken down in the table below. The displays currently configured and that can be readily used on the system are:

- 1) One [SGI](#) 18.1" F180 digital/analog LCD [2],
- 2) Four 24" [Sony/SGI](#) FD Trinitron Flat CRT GDM-FW900,
- 3) One [ViewSonic](#) VP2290b 22.2" LCD [3], and
- 4) Four [FakeSpace](#) projectors for the VR facility.

The table below lists the aforementioned devices and presents them down according to signal type and resolution/refresh rate combination. However, each display can run using multiple resolution/refresh rate combinations and the presents this as well. In so doing, sometimes two different displays will support both the same signal type and resolution/refresh rate combination. Primary displays take into account the main display that will be used given a resolution/refresh rate combination and signal type. However, sometimes, a second display is equally supported; hence, it is then listed as a secondary display.

Table 9. Display resolution and refresh rate combinations vs. display type (primary and secondary) for existing configurations according to signal type

<u>Primary Display</u>	<u>Secondary Display</u>	<u>Signal type</u>	<u>Resolution@ Refresh rate (Hz)</u>
SGI F180 18.1" LCD	24" Trinitron CRT	Analog	1280x1024@60
SGI F180 18.1" LCD	None	Digital	1280x1024@60
24" Trinitron CRT	SGI F180 18.1" LCD	Analog	1280x1024@60
24" Trinitron CRT	SGI F180 18.1" LCD	Analog	1280x1024@75
24" Trinitron CRT	None	Analog	1280x1024@85
24" Trinitron CRT	None	Analog	1280x1024@96
24" Trinitron CRT	None	Analog	1400x1050@60

24" Trinitron CRT	None	Analog	1400x1050@75
24" Trinitron CRT	FakeSpace projector	Analog	1400x1050@96
24" Trinitron CRT	FakeSpace projector	Analog	1400x1050@100
24" Trinitron CRT	None	Analog	1600x1200@60
24" Trinitron CRT	None	Analog	1600x1200@75
24" Trinitron CRT	None	Analog	1600x1200@85
24" Trinitron CRT	None	Analog	1920x1200@60
24" Trinitron CRT	None	Analog	1920x1200@75
24" Trinitron CRT	None	Analog	1920x1440@60
24" Trinitron CRT	None	Analog	1920x1440@75
FakeSpace projector	24" Trinitron CRT	Analog	1400x1050@96
FakeSpace projector	24" Trinitron CRT	Analog	1400x1050@100
ViewSonic VP2290b	None	Digital	960x2400@30
ViewSonic VP2290b	None	Digital	960x2400@41
ViewSonic VP2290b	None	Digital	1920x1200@30
ViewSonic VP2290b	None	Digital	1920x1200@41
ViewSonic VP2290b	None	Digital	1920x2400@25

The 24" [Sony/SGI](#) FD Trinitron Flat CRT GDM-FW900 and the [FakeSpace](#) projectors are analog-only devices. Display device [SGI](#) F180 18.1" LCD can accept both analog and digital signals, while display device [ViewSonic](#) VP2290b can only accept digital signals.

4. Configuring applications, scripts, and system parameters

4.1 Introduction

This section will concentrate on presenting relevant information about: 1) the configuration of global system parameters for use with [OpenGL Performer](#); 2) deploying and using virtual desktops; 3) providing multipipe functionality to single-pipe applications; 4) exploiting the 3D models and scenegraphs using [OpenGL Performer](#)'s visualization tool [Perfly](#), and 5) using the hardware graphics [compositor](#) with [Perfly](#).

The installation of [OpenGL Performer](#) and [Multipipe](#) are not explored in this document, nor are the installations of the system's various 3D models and scenegraphs. Installation and development-related information can be found in the system's documentation library that is available running the command *infosearch*.

4.2 Required system variables

All the system's 3D models and scenegraphs can be displayed using the [Perfly](#) visualization tool. Because [Perfly](#) is used, these simulations can run in single-pipe or multipipe mode (see Section 4.4). They can also run on the hardware graphics [compositor](#) using [Perfly](#). However, in order for [Perfly](#) and the hardware graphics [compositor](#) to work correctly together, several system variables must be set, including one general use memory-based variable that optimizes performance.

The majority of the issues between [Perfly](#) and the hardware graphics [compositor](#) are due to how [Perfly](#) attempts to controls the hardware graphics [compositor](#). By setting certain system variables, it becomes possible to ensure compatibility between them. These variables should be set in the global initialization file, */etc/cshrc*, so that the changes are available across the system.

The first variable that must be changed allocates physical memory to [Perfly](#), optimizing its performance. The more memory that is available for it to store textures and other data the better visualization performance will be. Through some experimentation, it has been determined that with [Perfly](#), more than 1 GB of memory per instance of the application will not help to increase performance; instead, it may actually have the contrary effect. Therefore, the following variable must be set in */etc/cshrc*:

```
PFSHAREDSIZE=1000000000
```

After reboot, every instance of [Perfly](#) that will require 1 GB RAM. Therefore, no more than three concurrent [Perfly](#) simulations should be run unless the amount of RAM dedicated to [Perfly](#) is reduced. The system currently sports 4 GB RAM and purchasing more is too expensive.

The variables listed below resolve the issues with the hardware graphics [compositor](#). The hardware graphics [compositor](#) is designed to take signals from each of its connected graphics pipes and create an enhanced image where each pipe handles only a part of the rendering load.

This load is distributed across all of the connected pipes and the hardware graphics [compositor](#) “rebuilds” the split images and recombines them into one full image that can be redisplayed.

However, [Perfly](#) can encounter certain incompatibilities, depending on how it is used. While the full [OpenGL Performer](#) SDK is fully capable of working with the hardware graphics [compositor](#), [Perfly](#) does not necessarily take full advantage of those API’s. At times, the incompatibilities could be so severe that the kernel would panic and perform a core dump. Fixes were made known through discussions with [SGI](#)’s senior graphics engineers. The variables and parameters they proposed to fix the problem (which worked very well) are as follows:

```
PF_SKIPSWAPBARRIERS=1

PFCOMP_SPATIAL_PADDING=1

PF_SEMA_BEFORE_SWAP=1

PFU_DONT_LOAD_WIN_CURSORS=1
```

The five variables examined in this section will not in any way affect other [OpenGL](#)-based applications. They are also completely compatible with channel cloning and desktop extension (see Section [3.2.1](#)).

4.3 Multipipe and DMX

4.3.1 Multipipe, DMX, Performer, and X specifics

[OpenGL Multipipe](#) [9] [10] is another method for harnessing multiple graphics pipes. [OpenGL Multipipe](#) consists of two parts: the API provided by the [OpenGL Multipipe](#) SDK, and [DMX](#), a command line tool. The SDK provides all the API’s that a developer needs to write a multipipe-aware [OpenGL](#)-based program. [DMX](#), the Distributed Multihead X proxy is an [X](#)-based proxy service that runs on top of [X Windows](#), intercepting all [OpenGL](#) calls and translating them so that single-pipe [OpenGL](#) programs can be run in a multipipe environment, making them behave as though they actually were multipipe-based. [DMX](#) accomplishes this by providing a hardware abstraction layer so that the physical underlying hardware is completely hidden from the single-pipe [OpenGL](#) program. This fools the program into thinking that it is displaying using just one pipe; however, in reality, it is displaying to multiple pipes.

However, [Multipipe](#) is not very useful for [Performer](#)-based applications (such as [Perfly](#)). This is because most [Performer](#)-based applications can take full advantage of multiple graphics pipe by using various API’s provided by the [OpenGL Performer](#) SDK. Through its very extensive set of API’s, multipipe-aware applications can easily be written. Furthermore, running [Performer](#)-based application under a [DMX](#) environment will result in a significant slowdown because [Performer](#)-based applications communicate directly with the system’s hardware rather than through the proxy service provided by [DMX](#). The [Performer](#) SDK is a very high-performance real-time application development library used to provide and create realistic 3D simulations. As such, [DMX](#) should only be considered in the event that a [Performer](#)-based application is not multipipe-aware. In the situation occurs, then [Multipipe](#) SDK can be used to add multipipe functionality to

the program, or using the [DMX](#) command line launcher the application can be made to run as though it were multiple-enabled.

The [X](#) server, compared to [Multipipe](#) (and [Performer](#)) is a strange creature by comparison. Under [X Windows](#), each pipe, by default, has its own desktop. Channels on a given pipe share the same desktop. However, on the [Onyx4](#), dual-channel capability is enabled only if channel cloning or desktop extension is enabled (see Section [3.2.1](#)). One of the major limitations of having one desktop per pipe is that they are all independent and data and programs cannot be enlarged or shared across multiple desktops. However, the mouse can move seamlessly between desktops.

However, through [Multipipe](#) SDK and [DMX](#), these limitations can be surpassed. If during development the [Multipipe](#) or [Performer](#) SDK is used, then the application is already multipipe-aware. However, if this is not the case, then [DMX](#) can be used to create an enlarged virtual desktop where applications and data can be shared, moved, and resized across multiple pipes. Once a [Multipipe](#) virtual desktop is created, it is no longer single-pipe in nature, and neither are the programs and applications running under [X Windows](#). Now any application can be scaled to any size that is supported by the system. Programs such as web browsers, [Xterm](#)'s, and configuration utilities can now be rescaled and data can be moved anywhere across the workspace. The maximum desktop size supported by [Multipipe](#) and by [X Windows](#) is a managed workspace of up to 32 K x 32 K pixels in size.

[Performer](#), [Multipipe](#), and [DMX](#) will work on any array of displays: square, CAVE, vertical or horizontal arrangements (see Section [3.7](#)). They are also fully compatible with channel cloning and desktop extension (see Section [3.2.1](#)). Combining desktop extension and [DMX](#) can be used to form very large desktops that can be combined using any of the various display configurations examined in Section [3.7](#). However, using desktop extension with [DMX](#) will result in a drop of overall visualization performance due to the overhead of the proxy service and the additional bandwidth and GPU processing requirements placed on the graphics pipes. While [Multipipe](#) and [DMX](#) are powerful and versatile, they should not be overused. The system as whole may experience anything from a slight drop to a severe degradation in performance, depending on how [Multipipe](#) or [DMX](#) is implemented. In addition, the same holds true for [Performer](#); although [Perfly](#) can display to both channels if desktop extension is enabled, a significant degradation of performance will ensue due to graphics pipe bandwidth limitations.

4.3.2 Understanding DMX configuration files

Based on information from [\[9\]](#), [DMX](#) can be used to create extended virtual desktops. This can be done through the creation of configuration files that describe the absolute location of all [Multipipe](#)-enabled displays in a desktop. These text-based files are known as [DMX](#) configuration files. They are simple to create once their basic structure is understood.

It is considered a best practice if the [DMX](#) configuration file follows the same basic physical display layout as set forth in the *ServerLayout* section of a corresponding *XF86Config-4* configuration file. This will greatly ease the design of the [DMX](#) configuration file. However, [DMX](#) configuration files can also be used to rearrange the order and appearance of the physical displays and their relation to one another, potentially counteracting any *XF86Config-4* configurations.

Creating a virtual desktop is also known as creating an [OpenGL X](#) Proxy-based service layer. This [X](#) proxy layer (or service) is a generic and abstract layer for general multipipe computing. However, [DMX](#) use should be avoided when running [Performer](#)-based multipipe-aware applications such as [Perfly](#).

The structure of a [DMX](#) configuration file is always the same, and it is easy to read and understand. The first line of every [DMX](#) file starts with a *virtual* tag followed by a *virtual name* tag (i.e. “*virtual*” and “*cave*”, respectively). This is then normally followed by a *size* tag where the actual size of the combined displays is described. However, this can be skipped if for some reason the calculation of the total size is either uncertain or difficult to calculate. The *virtual name* tag should be descriptive and use names such as *square*, *cave*, *vertical*, *horizontal*, etc. Then the first pipe is described using the logical pipe name. All other pipes’ positions depend on the position of the first pipe.

The [DMX](#) configuration file must be set up in terms of graphics pipes and not channels; [Multipipe](#) has no direct method for addressing specific channels. However, [X](#) can directly address them using channel cloning or desktop extension (see Section [3.2.1](#)). If desktop extension is to be used under [DMX](#), it is important to state the true size of each pipe, taking into account both channels. Because each pipe’s desktop will be twice its original size, either in the horizontal or vertical direction (depending on the extension type used), an appropriate adjustment must be made to the configuration file. For example, if all the pipes before desktop extension were 1,280 pixels in width, then enabling horizontal desktop extension, each pipe would be 2,560 pixels wide in the [DMX](#) configuration file. The same rule applies if the pipes are extended in the vertical direction. The size of standard geometric shapes such as squares and rectangles are simple to calculate. However, complex geometric shapes may not be so simple to calculate, and in such a case, it is best not apply any global size and allow [DMX](#) to manage the size automatically.

Pipes are addressed by their logical device numbers such as pipes *:0.0*, *:0.1*, *:0.2*, and *:0.3*. The referencing positions used in a [DMX](#) file are always relative to the first pipe specified, which is normally *pipe :0.0*, although it does not have to be. Consider the following example:

```
Virtual example_config 3840x2400 {  
    display :0.3 1920x1200;  
    display :0.1 1920x1200 @1920x0;  
    display :0.2 1920x1200 @0x1200;  
    display :0.0 1920x1200 @1920x1200  
}
```

In such a scenario, *pipe :0.3* is the point of reference for all other pipes. The position of the first pipe described in a [DMX](#) configuration file is absolute; all other positions are relative with respect to it. Either all other pipes are vertically at the same level as the first pipe, or below it; they are also horizontally to its right. The first pipe is considered to have position *0, 0*. Virtual displays are created using an inverted first quadrant Cartesian coordinate system, where moving to the right and to the bottom are considered as positive. In such a system, the natural tendency is for displays to be written to the right of and below a given point of reference (i.e. the first pipe). All other pipes must have a non-zero position. With this knowledge, it is now possible to build [DMX](#) configurations files.

For example, if all the pipes in a given configuration are 1920x1200 in size, then they can be expressed as being x pixels to the right and y pixels below the first pipe. Examining *pipe :0.1* from the above example, it is easy to see that it is 1,920 pixels to right of and 0 pixels below *pipe :0.3*. *Pipe :0.2* is directly 1,200 pixels below *pipe :0.3*. Moreover, *pipe :0.0* is 1,920 pixels to the right and 1,200 pixels below *pipe :0.3*. In this manner [DMX](#), configuration files are created.

The size 3840x2400 is simple to understand. The example above is using a square configuration with two pipes on the top and two on the bottom. Adding up the desktop area of each pipe yields 3840x2400 pixels. However, if desktop extension were enabled, horizontally, then it would have read 7680x2400 pixels.

There are of course certain exceptions to the aforementioned rules. It may occur when there are cases that a display must be configured to the left or even above the first pipe. These situations are rare and generally can be avoided if the pipe(s) can be physically rewired at the rear of the computer system. However, if rewiring cannot be done, then these rules can be violated. However, these scenarios are not be explored in this document. Furthermore, it is altogether possible to leave out or skip over certain displays from the virtual desktop. This is not normally useful; however, sometimes a user may wish to have one desktop that is independent of the others. This type of configuration is simpler to put into use than implementing multiple [X](#) servers to manage multiple desktops. These configurations will not be explored in this document.

Finally, it is also possible to define overlapping displays using [DMX](#) configuration files. This option is useful when using the hardware graphics [compositor](#) or when adjusting for screen overlap caused by adjacent displays. This option is not examined in this document.

4.3.3 Configuring various DMX configuration files

In keeping with the examples provided in Section [3.7.1](#), the following [DMX](#) configuration files correspond to the examples provided from that section. Short explanations will be provided below, following each of the example DMX configuration files.

4.3.3.1 Example 1: The horizontal line display configuration

This example corresponds to Section [3.7.1.1](#). It examines a horizontal line (4x1-display configuration) arrangement of displays, running at 1400x1050. It can be written as follows:

```
virtual horizontal 5600x1050 {
    display :0.0 1400x1050;
    display :0.1 1400x1050 @1400x0;
    display :0.2 1400x1050 @2800x1050;
    display :0.3 1400x1050 @4200x0;
}
```

This example is not particularly difficult to understand. The pipes, in terms of the logical order in which they will be physically displayed are arranged in a horizontally continuous configuration. Each pipe follows the next pipe, with each position firmly fixed. The first pipe, *pipe 0*, does not require any position since it sits at position 0, 0. The sizes of the pipes used indicate that they

correspond to the same resolution used by the VR facility's projectors. The pipes go from left to right in terms of visual output.

4.3.3.2 Example 2: The square display configuration

This example corresponds to Section [3.7.1.2](#). It shows a square-based configuration (2x2-display configuration) which is very interesting because it represents one of the more useful screen configurations for the [ViewSonic](#) VP2290b LCD. This device can be displayed at resolutions up to 3840x2400 (9.2 mega-pixels) and can be driven using 1, 2, or 4 channels. Using four channels will prevent the pipes from overloading so that more realistic-looking visualization can occur using higher refresh rates. Therefore, using one channel per pipe (for a total of four pipes) yields the following [DMX](#) file:

```
virtual vp2290b 3840x2400 {
  display :0.0 1920x1200 @0x0;
  display :0.1 1920x1200 @1920x0;
  display :0.2 1920x1200 @0x1200;
  display :0.3 1920x1200 @1920x1200;
}
```

The configuration defines the displays starting from top-left to top-right and then bottom-left to bottom-right. Notice that *pipe 0*'s position is explicitly stated; this could just as easily have been left out. However, sometimes it is useful to include the position, especially for readability.

4.3.3.3 Example 3: The CAVE display configuration

The CAVE (3x1-display configuration) configuration example found in Section [3.7.1.3](#) is a more complex [DMX](#) configuration to create. This configuration is particularly useful for the VR facility's projectors and CRT displays. For example, assuming that the projectors and CRT's can be driven at resolutions up to 1920x1440, the following [DMX](#) configuration could be used for driving a CAVE:

```
virtual cave {
  display :0.0 1920x1440 @0x0;
  display :0.1 1920x1440 @1920x0;
  display :0.2 1920x1440 @3840x0;
  display :0.3 1920x1440 @1920x1440;
}
```

This configuration is a little more difficult to visualize, yet it is still comprehensible. It merely describes, in terms of absolute location, what the *ServerLayout* section of the corresponding configuration file describes using relative terms such as top, bottom, left, and right (see Section [3.7.1.3](#)). An interesting observation from this configuration is that *pipe :0.3*, although physically and logically below *pipe :0.1*, it is using a positive number that describes a bottom-based display.

4.3.4 Starting the virtual desktop

Before using a virtual desktop, a [DMX](#) configuration file should have been created. Once this file has been created, using a command line program from within an [Xterm](#), the virtual desktop can be started. Virtual desktops can be created by any user currently logged in to the [X Windows](#) system; it is not a root-only task. For example, using a 2x2-display configuration as shown in Section [4.3.3.2](#), the [DMX](#) configuration would have descriptive name, *square.dmx*, and could be found under */etc/dmx*. The following command would be used to start up the virtual desktop:

```
$ ompstartdmx -cfgfile /etc/dmx/square.dmx
```

This will start up and create a virtual desktop as defined by the *square.dmx* [DMX](#) configuration file. The only time where it may not be necessary to create a [DMX](#) configuration file is when a horizontally based display configuration is used, as shown in Section [4.3.3.1](#). Instead of using a configuration file, simply running the following command without any optional parameters will create the virtual desktop based on [DMX](#) configuration defaults, as follows:

```
$ ompstartdmx
```

Once the virtual desktop is running, it will eventually become necessary to logout in order to return to the standard system desktop. To shut off any virtual desktop that is currently running, it can be logged out from as done from any other windowing desktop environment. However, if for some reason this is not possible due to an unresponsive desktop, then a forced exit is required. This can be accomplished using the hotkey “Alt-Ctrl-Q.” The hotkey will immediately kill the virtual desktop and restore the original system desktop.

4.3.5 Running other DWM's

The default DWM (Desktop Window Manager) for [IRIX](#) is [4DWM](#). On the current system, [GNOME](#) and [KDE](#) are available for use as optional DWM's. They were obtained from [The Written Word](#), a for-profit open source company that recompiles popular open source products for the [IRIX](#) operating system (and many other commercial [UNIX](#) platforms as well). These packages are already installed on the system and can be run by any user. A user does not need to have root access to run another DWM from within a virtual desktop or from within the system's default DWM. However, if the [GNOME](#) or [KDE](#) DWM is launched before a virtual desktop is created, the DWM will only run under the specific desktop from where it was launched; all other desktops will remain independent and continue running their default DWM.

However, if the virtual desktop is enabled, then any subsequent DWM that is launched will span across the entire virtual desktop. To accomplish this, it suffices to start the virtual desktop using the *ompstartdmx* command (see Section [4.3.4](#)). Once it is running, [GNOME](#) or [KDE](#) can be launched using the following two commands, respectively:

```
$ gnome-session &
```

Or

```
$ startkde &
```

Each desktop has its own hotkey or procedure for shutting down and it should be used whenever possible. Once the DWM is shutdown, the user is automatically returned back to the virtual desktop or default DWM desktop, of which either can also be logged out from, if necessary. Each desktop has its own emergency hotkey for immediate shutdown. However, if a DWM is running under a virtual desktop, then it can be killed at any time using the “Alt-Ctrl-Q” hotkey. This will immediately shutdown the DWM and return the user back to the system’s original DWM desktop.

4.3.6 Running applications under Multipipe and DMX

While [DMX](#) can be used to create enlarged virtual desktops, up to 32K x 32 K pixels in size, it can also be used to enable single-pipe applications to run as though they were multipipe-enabled [9]. However, multipipe-enabled and multipipe-aware applications are not the same thing. They differ primarily in that multipipe-aware applications have been programmed using either the [Performer](#) or [Multipipe](#) SDK and utilize multipipe-aware API’s. Multipipe-enabled applications, on the other hand, are single-pipe programs that have been programmed without the use of any multipipe-aware API. Furthermore, these programs may possibly not be using [OpenGL](#) graphics-based API’s. Either way, the application does not have the ability to use multiple pipes and must rely on the [DMX X](#)-based proxy service for this capability.

Multipipe-aware applications should not be run in a [DMX](#) environment due to the additional overhead caused by the interception and retranslation of existing multipipe API’s used by the program. Instead, these programs should be run under a normal [X Windows](#) environment so that the application can see the actual physical hardware and communicate directly with it rather than through a proxy service. However, if for some reason a [DMX](#) environment is necessary for other reasons, then the application can be run under [DMX](#) using the following command:

```
$ omprun -aware application
```

The *-aware* flag alerts the [DMX](#) proxy service not to perform any [OpenGL](#) interception or call translation on the running program. This will enable the application to run at its native speed and communicate directly with the hardware.

Standard [X Windows](#) applications are run as they normally are since they do not use any [OpenGL](#) calls; all graphics API’s are [X](#)-based. Therefore, pure [X](#) programs do not need to be run using any command line launcher. Running these programs as they are under a [DMX](#) environment is sufficient to allow them to behave as though they were multipipe-enabled, as the proxy service will take care of everything. However, there are certain compatibility issues between older [X](#)-based programs and the [DMX](#) proxy service. This is, however, rare. This tends to occur with very old [X](#)-based programs. The problem can be fixed by exiting a [DMX](#) environment and running the application under a standard [X](#) environment; however, the program will not be able to benefit from multiple pipes.

The final type of application to run is [OpenGL](#)-based; that does not use any multipipe-based API’s. These types of programs can be multipipe-enabled by running them as follows:

```
$ omprun application
```

This will cause the program's [OpenGL](#) calls to be intercepted and translated by the [DMX](#) proxy service, empowering it with multipipe-based capabilities. The program can then be moved and rescaled across multiple pipes as though it were multipipe-aware. However, these programs will run slightly slower than they normally would under a non-[DMX](#) environment due to the added proxy service overhead.

4.4 Performer based applications

4.4.1 About Performer and Perfly

According to [11], [Performer](#) can be defined as “*an extensible software toolkit for creating real-time 3D graphics. Typical applications are in the fields of visual simulation, entertainment, virtual reality, broadcast video, and computer aided design. OpenGL Performer provides a flexible, intuitive, toolkit-based solution for developers who want to optimize application performance.*”

[Performer](#) is most commonly found on high-end graphics workstations and supercomputers that drive near real-time simulations that use heavily textured 3D models and scenegraphs commonly found in today's complex and demanding VR environments. [Performer](#) is the most high-performance; multi-platform, multipipe-aware, multi-capable, multi-threaded [OpenGL](#) programming toolkit that currently exists on the market. Almost all of the high-end commercially available simulation and visualization-based applications are based on [Performer](#).

[Perfly](#) is a sample visualization exploration tool that is programmed completely using the [OpenGL Performer](#) SDK. Although it is only a sample program, it is nonetheless very capable and powerful. It can be used to visualize and render very complex 3D models and scenegraphs and it can work in a variety of physical environments and on any platform supported by [Performer](#)⁶. It is also capable of recognizing the majority of today's scenegraphs, 3D model formats, and databases, including [OpenFlight](#) formats versions 11 and 14. It can be used to drive very demanding simulations using the hardware graphics [compositor](#) where improved rendering performance is required.

4.4.2 Configuring Perfly and the visualization scripts

The system currently has many different high quality and heavily textured scenegraphs ready for use. Examples are of villages, cities (including Quebec City), desert terrains, military base installations and airports, and many others. These scenegraphs are very heavy and at the time of purchase in 2004, no other COTS PC-based system could compete with the rendering capabilities of the [Onyx4](#) visualization server.

It was decided early on that [Perfly](#) had most of the necessary functionality required for displaying large and complex scenegraphs. These scenegraphs were created using the [OpenFlight](#) format, ensuring compatibility between various platforms since the [OpenFlight](#) format is considered an industry standard. However, [Perfly](#) is lacking in its ability to display in 3D Stereo, and by default, it does not support a CAVE configuration. However, with the purchase of [OpenGL](#)

⁶ It currently runs on [IRIX](#), [Linux](#), and [Windows](#).

[Performer](#) comes the source code to [Perfly](#), and therefore, modifications can be made to it to incorporate these requirements. This may be done at a later time.

Each scenegraph or model has its specific scripts supporting the various display configurations examined in Section [3.7.1](#), allowing for a variety of ways in which the data can be visualized. In total, there more 51 [Perfly](#) simulation-based scripts. However, due to the number of scripts, rather than review them one at a time, the scripts for the City of Quebec will be used. The examples shown below will be representative of the examples presented in Section [3.7.1](#). It is important to note that [DMX](#) is not necessary for running any of the scripts shown below. Furthermore, the locations of the [Perfly](#) simulation scripts are outlined in Section [3.8](#) and can be found in */etc/X11/openflight*.

4.4.2.1 Example 1: Perfly and the horizontal line display configuration

This section's script, shown below, is based on the horizontal line found in Section [3.7.1.1](#). The memory buffer for [Performer](#) is set, and then the script changes directory to the scenegraph's storage location and loads the scenegraph without any additional command line options. It is important that the appropriate horizontal line display configuration file be loaded prior to running the script. The script is as follows:

```
#!/bin/csh
setenv PFSHAREDSIZE 1000000000
cd /home3/openflight/Quebec-cmartel/modele/flt
/usr/sbin/perfly quebec.flt
```

4.4.2.2 Example 2: Perfly and the square display configuration

The example below is based on the square configuration as shown in Section [3.7.1.2](#). However, this script differs in many regards from the script shown above in Example 1. The variables shown in Section [4.2](#) are re-set in this script in case the user's environment may no longer have these settings.

To display using a square configuration, [Perfly](#) relies on three special [compositor](#)-based environmental settings. However, instead of using the hardware graphics [compositor](#), software compositing will be used instead. Nevertheless, [Performer](#) requires these variables to be set in order to work correctly. They are:

```
PF_SEMA_BEFORE_SWAP
PFU_DONT_LOAD_WIN_CURSORS, and
PF_SKIPSWAPBARRIERS
```

Furthermore, [Perfly](#) must be explicitly instructed how to setup the displays. When compositing, the bottom displays are set before the top displays. However, the displays are similar to the configuration found in a [Powerwall](#). Therefore, a special configuration exists within [Perfly](#) to deal with software-based compositing for a [Powerwall](#)-like device. Before showing the command line option for this, it is important to understand the physical layout of the displays. This is shown below using a simple diagram:

```
-----  
|:0.0 |:0.1 |  
-----  
|:0.2 |:0.3 |  
-----
```

The command line option that supports this setup is:

```
- M5,2,2,0,0,p=:0.2,p=:0.3,p=:0.0,p=:0.1
```

More information about the command line options used, the Man file [12] may be consulted. Before running the following script, the appropriate square display configuration file must be loaded. The script is as follows:

```
#!/bin/csh  
setenv PFSHAREDSIZE 1000000000  
setenv PF_SEMA_BEFORE_SWAP 1  
setenv PFU_DONT_LOAD_WIN_CURSORS 1  
setenv PF_SKIPSWAPBARRIERS 1  
cd /home3/openflight/Quebec-cmartel/modele/flt  
/usr/sbin/perfly -M5,2,2,0,0,p=:0.2,p=:0.3,p=:0.0,p=:0.1 quebec.flt
```

4.4.2.3 Example 3: Perfly and the CAVE display configuration

Unfortunately, [Perfly](#) does not support the CAVE display configuration as found in Section [3.7.1.3](#). However, according to SGI's technical support staff the issue can be fixed by making several modifications to the [Perfly](#)'s source code. Performing these modifications is outside the scope of this document and requires an in-depth knowledge of the [Performer](#) library. Therefore, while the CAVE display configuration file can be used, a horizontal line [Perfly](#) script must be used, as shown in Section [4.4.2.1](#). Before running this script, however, the CAVE display configuration file must be loaded. The script to run would be:

```
#!/bin/csh  
setenv PFSHAREDSIZE 1000000000  
cd /home3/openflight/Quebec-cmartel/modele/flt  
/usr/sbin/perfly quebec.flt
```

4.4.3 Using the hardware graphics compositor with Perfly

[Perfly](#)'s [Performer](#)-based API's are fully multipipe-aware and capable of driving and controlling the hardware graphics [compositor](#). No background theory about the hardware graphics [compositor](#) will be provided here due to the device's complexity. The system comes with special timing-table files that are used with the hardware graphics [compositor](#). These files are found */usr/gfx/ucode/STINGRAY/cmb* and their names are self-explanatory. The following formats are supported by the hardware graphics [compositor](#):

- 1280x1024 running at 60, 75, and 96 Hz
- 1600x1200 running at 60 Hz
- 1920x1200 running at 60 Hz

In order to use the hardware graphics [compositor](#), a timing-table and its corresponding *XF86Config-4* configuration file must be loaded. In this case, the timing-table is loaded from */usr/gfx/ucode/STINGRAY/cmb* and the *XF86Config-4* configuration file is loaded from */etc/X11/digital-compositor-XF86Config*. For example, assuming that the hardware graphics [compositor](#) will be running at 1280x1024@60 Hz, then the *XF86Config-4* configuration file as shown in Annex [A.4](#) will be used. To prepare the hardware graphics [compositor](#) for use, the script */etc/X11/set-compositor-scripts/1280x1024_60.sh* can be used. However, this script must be from either a TTY or a Telnet session command line; doing otherwise will undoubtedly crash [X Windows](#), and possibly upload corrupted data to the hardware graphics [compositor](#). Its contents are as follows:

```
#!/sbin/csh

rm /var/X11/Xvc/*TimingTable
setenv DISPLAY :0.0
/usr/gfx/setmon -xp0 1280x1024_60
/usr/gfx/setmon -xp1 1280x1024_60
/usr/gfx/setmon -xp2 1280x1024_60
/usr/gfx/setmon -xp3 1280x1024_60
/usr/gfx/setmon -xSp0 1280_1024_60_no_tiles

/usr/gfx/stopgfx
/usr/gfx/gfxinit
/usr/gfx/startgfx
```

This script, once run, flushes are state files and forcibly uploads the appropriate timing-table file to the hardware graphics [compositor](#) and sets all graphic pipes to the same uniform resolution. Before running this script, the corresponding [compositor](#)-based *XF86Config-4* must replace the file */etc/X11/XF86Config-4*.

To run a [Perfly](#) simulation, the Quebec City simulation script *realdb.sh* can be run from */etc/X11/openflight/compositor*. The script's contents are as follows:

```
#!/bin/csh
setenv PFSHAREDSIZE=1000000000
cd /home3/openflight/Quebec-realdb/QUEBEC_DATABASE
/usr/sbin/perfly -M6,c=0 QUEBEC_FINAL.flt
```

Now the Quebec City [OpenFlight](#) scenegraph is running on the hardware graphics [compositor](#) and runs with more than triple the performance of running at the same resolution from just a single graphics pipe. To learn more about the command line option used in this script, the Man file [\[12\]](#) may be consulted. Furthermore, this reference will also provide information on toggling between static and dynamic compositing.

5. Conclusion

It has been shown using relatively simple terms and easy to understand examples how to work not only with the *XF86Config-4* configuration file, but also how to modify it and add additional capabilities to it such as 3D Stereo, FSAA, channel cloning and desktop extension. Using information from this report it is now possible to create new Modelines for rapid inclusion in the configuration file for instant implementation and deployment.

Furthermore, the various types of preconfigured displays, LCD's, and projectors have been examined, as have the locations of the various preconfigured *XF86Config-4* files and the directories that store them and the various scripts and miscellaneous data, [DMX](#) configuration files, and [Perfly](#) start-up scripts.

Finally, the use and implementation of certain more complex issues such as the hardware graphics [compositor](#), [OpenGL Performer's Perfly](#) and [OpenGL Multipipe](#) have been examined and will enable users and system administrator alike to deploy and use virtual desktops and run single-pipe applications in multipipe mode.

The [Onyx4](#) is a powerful and versatile computing platform. Its complexity, unfortunately, is due to the many ways in which it can be configured to perform a variety of tasks. Nevertheless, it can be used to drive complex simulations and perform numerically intensive calculations.

The only issue that has not been examined due to its excessive complexity is the direct implementation and exploitation of the hardware graphics [compositor](#) via [OpenGL Multipipe](#). SGI's technical references are suitable for this task, although they nevertheless are vague on the subject.

Users and system administrators alike, new to new [Onyx4](#) visualization server now have enough practical information available at their disposal to make good use of this computing platform without having to toil through complex technical documentation.

References

- [1] SGI. Silicon Graphics Onyx4 UltimateVision User's Guide. Technical reference guide. SGI. June 2004. Document number 007-4634-002. http://techpubs.sgi.com/library/tpl/cgi-bin/browse.cgi?coll=hdwr&db=bks&cmd=toc&pth=/SGI_EndUser/Voyager_UG.
- [2] SGI. SGI F181 18-inch Flat Panel Display User Guide. Technical reference document. SGI. August 2003. Document number 860-0358-001. <http://techpubs.sgi.com/library/tpl/cgi-bin/download.cgi?coll=hdwr&db=bks&docnumber=860-0358-001>.
- [3] ViewSonic. VP2290b. Product description web page. ViewSonic. 2006. <http://www.viewsonic.com/support/desktopdisplays/lcddisplays/proseries/vp2290b/>.
- [4] SGI. Silicon Graphics UltimateVision Graphics Porting Guide. Technical reference guide. SGI. June 2004. Document number 007-4297-001. http://techpubs.sgi.com/library/tpl/cgi-bin/browse.cgi?coll=hdwr&db=bks&cmd=toc&pth=/SGI_EndUser/Onyx4_Port.
- [5] Dawes, David. XFree86Config(5) manual page. Online Man page. XFree86. 2006. <http://www.xfree86.org/current/XFree86Config.5.html>.
- [6] Wikipedia. XFree86 Modeline. Online encyclopedic reference. Wikimedia Foundation Inc. November 2006. http://en.wikipedia.org/wiki/XFree86_Modeline.
- [7] Raymond, Eric Steven. XFree86 Video Timings HOWTO. How-to. Revision 6.4. The Linux Documentation Project & Thyrus Enterprises. July 2004. <http://en.tldp.org/HOWTO/XFree86-Video-Timings-HOWTO/index.html>.
- [8] SGI. Setmon Man page. Online Man page. SGI. 2006. <http://techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi?coll=linux&db=man&fname=/usr/share/catman/man1/setmon.1.html>.
- [9] SGI. SGI OpenGL Multipipe User's Guide Multipipe. Technical reference guide. SGI. September 2005. Document number 007-4318-017. <http://techpubs.sgi.com/library/manuals/4000/007-4318-017/pdf/007-4318-017.pdf>.
- [10] SGI. OpenGL Multipipe FAQ. FAQ descriptive web page. SGI. 2006. <http://www.sgi.com/products/software/multipipe/faq.html>.
- [11] SGI. OpenGL Performer Getting Started Guide. Technical reference guide. SGI. December 2004. Document number 007-3560-005. http://techpubs.sgi.com/library/tpl/cgi-bin/browse.cgi?coll=0650&db=bks&cmd=toc&pth=/SGI_Developer/Perf_GetStarted.
- [12] SGI. Perfly Man page. Online Man page. SGI. 2006. <http://techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi?coll=linux&db=man&fname=/usr/share/catman/man1/perfly.1pf.html&srch=perfly%20man>.

This page intentionally left blank.

Annex A Example XF86Config-4 configuration files

A.1 The system default XF86Config-4

Below is the listing for the system default *XF86Config-4* configuration file. This configuration file does not enable 3D Stereo or FSAA support. It is running at 1280x1024@60Hz. It is configured for dual-channel automatic display detection and is configured for displaying in a horizontal line from left to right without the implementation of channel cloning or desktop extension.

The configuration file is as follows:

```
# $XFree86: xc/programs/Xserver/hw/xfree86/XF86Conf.cpp,v 3.44 2001/12/17
20:52:29 dawes Exp $
#
# Copyright (c) 1994-1998 by The XFree86 Project, Inc.
#
# Permission is hereby granted, free of charge, to any person obtaining a
# copy of this software and associated documentation files (the "Software"),
# to deal in the Software without restriction, including without limitation
# the rights to use, copy, modify, merge, publish, distribute, sublicense,
# and/or sell copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in
# all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
# KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
# MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO
# EVENT SHALL
# THE XFREE86 PROJECT BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF
# OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
# DEALINGS IN THE
# SOFTWARE.
#
# Except as contained in this notice, the name of the XFree86 Project shall
# not be used in advertising or otherwise to promote the sale, use or other
# dealings in this Software without prior written authorization from the
# XFree86 Project.
#
```

```

# $XConsortium: XF86Conf.cpp /main/22 1996/10/23 11:43:51 kaleb $

# *****
# This is a sample configuration file only, intended to illustrate
# what a config file might look like. Refer to the XF86Config(4/5)
# man page for details about the format of this file. This man page
# is installed as /usr/X11R6/man/man4/XF86Config.4
# *****

# The ordering of sections is not important in version 4.0 and later.

# *****
# Files section. This allows default font and rgb paths to be set
# *****

Section "Files"

# The location of the RGB database. Note, this is the name of the
# file minus the extension (like ".txt" or ".db"). There is normally
# no need to change the default.

    RgbPath    "/usr/lib/X11/rgb"

# Multiple FontPath entries are allowed (which are concatenated together),
# as well as specifying multiple comma-separated entries in one FontPath
# command (or a combination of both methods)

    FontPath  "/usr/lib/X11/fonts/100dpi:unscaled"
    FontPath  "/usr/lib/X11/fonts/75dpi:unscaled"
    FontPath  "/usr/lib/X11/fonts/misc/"
    FontPath  "/usr/lib/X11/fonts/Type1/"
    FontPath  "/usr/lib/X11/fonts/Speedo/"
    FontPath  "/usr/lib/X11/fonts/CID/"
    FontPath  "/usr/lib/X11/fonts/100dpi/"
    FontPath  "/usr/lib/X11/fonts/75dpi/"

# ModulePath can be used to set a search path for the X server modules.
# The default path is shown here.

    ModulePath "/usr/X11R6/lib/modules"

EndSection

# *****
# Module section -- this is an optional section which is used to specify
# which run-time loadable modules to load when the X server starts up.
# *****

```

Section "Module"

This loads the DBE extension module.

Load "dbe"

This loads the miscellaneous extensions module, and disables
initialisation of the XFree86-DGA extension within that module.

SubSection "extmod"
Option "omit xfree86-dga"
EndSubSection

This loads the the various modules required for X server operation

Load "type1"
Load "freetype"
Load "drm"
Load "fb"
Load "xaa"
Load "glx"
Load "dri"
Load "fglrx_glx"

EndSection

Server flags section. This contains various server-wide Options.

Section "ServerFlags"

Uncomment this to cause a core dump at the spot where a signal is
received. This may leave the console in an unusable state, but may
provide a better stack trace in the core dump to aid in debugging

Option "NoTrapSignals"

Uncomment this to disable the <Ctrl><Alt><BS> server abort sequence
This allows clients to receive this key event.

Option "DontZap"

Uncomment this to disable the <Ctrl><Alt><KP_+>/<KP_-> mode switching
sequences. This allows clients to receive these key events.

```

# Option "DontZoom"

# Uncomment this to disable tuning with the xvidtune client. With
# it the client can still run and fetch card and monitor attributes,
# but it will not be allowed to change them. If it tries it will
# receive a protocol error.

# Option "DisableVidModeExtension"

# Uncomment this to enable the use of a non-local xvidtune client.

# Option "AllowNonLocalXvidtune"

# Uncomment this to disable dynamically modifying the input device
# (mouse and keyboard) settings.

# Option "DisableModInDev"

# Uncomment this to enable the use of a non-local client to
# change the keyboard or mouse settings (currently only xset).

# Option "AllowNonLocalModInDev"

# Set the basic blanking screen saver timeout.

    Option "blank time" "10" # 10 minutes

# Set the DPMS timeouts. These are set here because they are global
# rather than screen-specific. These settings alone don't enable DPMS.
# It is enabled per-screen (or per-monitor), and even then only when
# the driver supports it.

    Option "standby time" "20"
    Option "suspend time" "30"
    Option "off time" "60"

# On some platform the server needs to estimate the sizes of PCI
# memory and pio ranges. This is done by assuming that PCI ranges
# don't overlap. Some broken BIOSes tend to set ranges of inactive
# devices wrong. Here one can adjust how aggressive the assumptions
# should be. Default is 0.

# Option "EstimateSizesAggressively" "0"

EndSection

# *****
# Input devices

```

```

# *****

# *****
# Core keyboard's InputDevice section
# *****

Section "InputDevice"

    Identifier "Keyboard1"
    Driver     "keyboard"

# For most OSs the protocol can be omitted (it defaults to "Standard").
# When using XQUEUE (only for SVR3 and 1, but not Solaris), comment
# out the above line, and uncomment the following line.

# Option     "Protocol"     "Xqueue"

# Set the keyboard auto repeat parameters. Not all platforms implement
# this.

    Option    "AutoRepeat"   "500 50"

# Specify which keyboard LEDs can be user-controlled (eg, with xset(1)).

# Option     "Xleds"        "1 2 3"

# To disable the XKEYBOARD extension, uncomment XkbDisable.

# Option     "XkbDisable"

# To customise the XKB settings to suit your keyboard, modify the
# lines below (which are the defaults). For example, for a European
# keyboard, you will probably want to use one of:
#
# Option     "XkbModel"     "pc102"
# Option     "XkbModel"     "pc105"
#
# If you have a Microsoft Natural keyboard, you can use:
#
# Option     "XkbModel"     "microsoft"
#
# If you have a US "windows" keyboard you will want:
#
# Option     "XkbModel"     "pc104"
#
# Then to change the language, change the Layout setting.
# For example, a german layout can be obtained with:
#

```

```

# Option "XkbLayout" "de"
#
# or:
#
# Option "XkbLayout" "de"
# Option "XkbVariant" "nodeadkeys"
#
# If you'd like to switch the positions of your capslock and
# control keys, use:
#
# Option "XkbOptions" "ctrl:swapcaps"

# These are the default XKB settings for XFree86
#
# Option "XkbRules" "xfree86"
# Option "XkbModel" "pc101"
# Option "XkbLayout" "us"
# Option "XkbVariant" ""
# Option "XkbOptions" ""

EndSection

# *****
# Core Pointer's InputDevice section
# *****

Section "InputDevice"

# Identifier and driver

    Identifier "Mouse1"
    Driver "mouse"

# The mouse protocol and device. The device is normally set to /dev/mouse,
# which is usually a symbolic link to the real device.

    Option "Protocol" "PS/2"
    Option "Device" "/dev/mouse"

# On platforms where PnP mouse detection is supported the following
# protocol setting can be used when using a newer PnP mouse:

# Option "Protocol" "Auto"

# When using mouse connected to a PS/2 port (aka "MousePort), set the
# the protocol as follows. On some platforms some other settings may

```

```

# be available.

# Option "Protocol" "PS/2"

# When using XQUEUE (only for SVR3 and 1, but not Solaris), use
# the following instead of any of the lines above. The Device line
# is not required in this case.

# Option "Protocol" "Xqueue"

# Baudrate and SampleRate are only for some older Logitech mice. In
# almost every case these lines should be omitted.

# Option "BaudRate" "9600"
# Option "SampleRate" "150"

# Emulate3Buttons is an option for 2-button mice
# Emulate3Timeout is the timeout in milliseconds (default is 50ms)

# Option "Emulate3Buttons"
# Option "Emulate3Timeout" "50"

# ChordMiddle is an option for some 3-button Logitech mice, or any
# 3-button mouse where the middle button generates left+right button
# events.

# Option "ChordMiddle"

EndSection

# Some examples of extended input devices

# Section "InputDevice"
# Identifier "spaceball"
# Driver "magellan"
# Option "Device" "/dev/cua0"
# EndSection
#
# Section "InputDevice"
# Identifier "spaceball2"
# Driver "spaceorb"
# Option "Device" "/dev/cua0"
# EndSection
#
# Section "InputDevice"
# Identifier "touchscreen0"
# Driver "microtouch"
# Option "Device" "/dev/ttyS0"

```

```

# Option "MinX"      "1412"
# Option "MaxX"      "15184"
# Option "MinY"      "15372"
# Option "MaxY"      "1230"
# Option "ScreenNumber" "0"
# Option "ReportingMode" "Scaled"
# Option "ButtonNumber" "1"
# Option "SendCoreEvents"
# EndSection
#
# Section "InputDevice"
# Identifier "touchscreen1"
# Driver "elo2300"
# Option "Device"      "/dev/ttyS0"
# Option "MinX"        "231"
# Option "MaxX"        "3868"
# Option "MinY"        "3858"
# Option "MaxY"        "272"
# Option "ScreenNumber" "0"
# Option "ReportingMode" "Scaled"
# Option "ButtonThreshold" "17"
# Option "ButtonNumber" "1"
# Option "SendCoreEvents"
# EndSection

# *****
# Monitor section
# *****

# Any number of monitor sections may be present

Section "Monitor"

# The identifier line must be present.

    Identifier "Generic Monitor"

# HorizSync is in kHz unless units are specified.
# HorizSync may be a comma separated list of discrete values, or a
# comma separated list of ranges of values.
# NOTE: THE VALUES HERE ARE EXAMPLES ONLY. REFER TO YOUR
MONITOR'S
# USER MANUAL FOR THE CORRECT NUMBERS.

# HorizSync 31.5 # typical for a single frequency fixed-sync monitor
# HorizSync 30-64 # multisync
# HorizSync 31.5, 35.2 # multiple fixed sync frequencies
# HorizSync 15-25, 30-50 # multiple ranges of sync frequencies

```

HorizSync 30-130 # multisync

VertRefresh is in Hz unless units are specified.
VertRefresh may be a comma separated list of discrete values, or a
comma separated list of ranges of values.
NOTE: THE VALUES HERE ARE EXAMPLES ONLY. REFER TO YOUR
MONITOR'S
USER MANUAL FOR THE CORRECT NUMBERS.

VertRefresh 60 # typical for a single frequency fixed-sync monitor

VertRefresh 50-100 # multisync

VertRefresh 60, 65 # multiple fixed sync frequencies

VertRefresh 40-50, 80-100 # multiple ranges of sync frequencies

VertRefresh 50-160 # multisync

Modes can be specified in two formats. A compact one-line format, or
a multi-line format.

A generic VGA 640x480 mode (hsync = 31.5kHz, refresh = 60Hz)

These two are equivalent

ModeLine "640x480" 25.175 640 664 760 800 480 491 493 525

Mode "640x480"

DotClock 25.175

HTimings 640 664 760 800

VTimings 480 491 493 525

EndMode

These two are equivalent

ModeLine "1024x768i" 45 1024 1048 1208 1264 768 776 784 817 Interlace

Mode "1024x768i"

DotClock 45

HTimings 1024 1048 1208 1264

VTimings 768 776 784 817

Flags "Interlace"

EndMode

ModeLine "1280x1024_96" 164.966 1280 1300 1460 1600 1024 1026 1029 1074

ModeLine "1280x1024_120" 233.793 1280 1384 1528 1776 1024 1025 1028 1097

Modeline "1920x1200_60p" 157.39 1920 1950 2072 2112 1200 1203 1206 1242

framelock timings, identified by the trailing 'f'

```

# (==> denotes available source video framelock formats)
# (example: setmon -L ntsc 1280x1024_5994f )

Modeline "1280x1024_50f" 104 1280 1336 1472 1664 1024 1027 1030 1250 -hsync -
vsync
# ==> 625i_50 (alias: pal), 1080p_25, 1080i_50

Modeline "1280x1024_5994f" 115.2 1280 1296 1440 1560 1024 1027 1030 1232 -hsync
-vsync
# ==> 525i_5994 (alias: ntsc), 720p_5994, 1080p_2997, 1080i_5994

Modeline "1280x1024_60f" 121.5 1280 1396 1540 1800 1024 1027 1030 1125 -hsync
-vsync
# ==> 720p_60, 1080p_30, 1080i_60

Modeline "1920x1154_48f" 144 1920 1968 2160 2400 1154 1157 1160 1250 -hsync -
vsync
# ==> 1080p_24, 1080i_48, this is a cvt modified format

Modeline "1920x1154_50f" 144 1920 1968 2160 2400 1154 1157 1160 1200 -hsync -
vsync
# ==> 625i_50 (alias: pal), 1080p_25, 1080i_50, this is a cvt modified format

Modeline "1920x1154_5994f" 192 1920 1992 2192 2464 1154 1157 1160 1300 -hsync -
vsync
# ==> 525i_5994 (alias: ntsc), 720p_5994, 1080p_2997, 1080i_5994, this is a cvt
modified format

Modeline "1920x1154_60f" 172.8 1920 1968 2160 2400 1154 1157 1160 1200 -hsync
-vsync
# ==> 720p_60, 1080p_30, 1080i_60, this is a cvt modified format

Modeline "1920x1200_60f" 180 1920 1968 2160 2400 1200 1203 1206 1250 -hsync -
vsync
# ==> 720p_60, 1080p_30, 1080i_60, this is a cvt modified format

Modeline "1920x1200_50f" 150 1920 1968 2160 2400 1200 1203 1206 1250 -hsync -
vsync
# ==> 1080p_25, 1080i_50, this is a cvt modified format

# Framelock timings with reduced blanking for formats frequency higher than dvi10 max
freq(165Mhz)

Modeline "1920x1154_5994f_p" 156 1920 1968 2000 2200 1154 1157 1160 1283 -
hsync -vsync
# ==> 525i_5994 (alias: ntsc), 720p_5994, 1080p_2997, 1080i_5994

```

```
Modeline "1920x1154_60f_p" 162 1920 1968 2000 2160 1154 1157 1160 1250 -hsync -vsync
```

```
# ==> 720p_60, 1080p_30, 1080i_60, this is a cvt modified format
```

```
Modeline "1920x1200_60f_p" 162 1920 1968 2000 2160 1200 1203 1206 1250 -hsync -vsync
```

```
# ==> 720p_60, 1080p_30, 1080i_60, this is a cvt modified format
```

```
#
```

```
# The following are standard VESA formats modified to be usable for a genlock source  
# through the "C SYNC" cable. The "C SYNC" cable combines sync's into a composite  
sync,
```

```
# but requires the sync levels to be negative. These ModeLines are distinguished by
```

```
# an "n" (for "negative" sync levels) appended to the ModeLine Name.
```

```
#
```

```
ModeLine "1280x1024_60n" 108 1280 1328 1440 1688 1024 1025 1028 1066 -hsync -vsync
```

```
ModeLine "1280x1024_75n" 135 1280 1296 1440 1688 1024 1025 1028 1066 -hsync -vsync
```

```
ModeLine "1280x1024_85n" 157 1280 1344 1504 1728 1024 1025 1028 1072 -hsync -vsync
```

```
ModeLine "1600x1200_60n" 162 1600 1664 1856 2160 1200 1201 1204 1250 -hsync -vsync
```

```
ModeLine "1600x1200_75n" 202 1600 1664 1856 2160 1200 1201 1204 1250 -hsync -vsync
```

```
ModeLine "1600x1200_85n" 229 1600 1664 1856 2160 1200 1201 1204 1250 -hsync -vsync
```

```
ModeLine "1920x1080_72n" 211 1920 2056 2264 2608 1080 1081 1084 1126 -hsync -vsync
```

```
ModeLine "1920x1200_60n" 193 1920 2048 2256 2592 1200 1201 1203 1242 -hsync -vsync
```

```
ModeLine "1920x1200_75n" 246 1920 2064 2272 2624 1200 1201 1203 1253 -hsync +vsync
```

```
# If a monitor has DPMS support, that can be indicated here. This will
```

```
# enable DPMS when the monitor is used with drivers that support it.
```

```
# Option "dpms"
```

```
# If a monitor requires that the sync signals be superimposed on the
```

```
# green signal, the following option will enable this when used with
```

```
# drivers that support it. Only a relatively small range of hardware
```

```
# (and drivers) actually support this.
```

```
# Option "sync on green"
```

EndSection

```
# *****  
# Graphics device section  
# *****  
  
# Any number of graphics device sections may be present  
  
#Section "Device"  
  
# The Identifier must be present.  
  
# Identifier "Generic VGA"  
  
# The Driver line must be present. When using run-time loadable driver  
# modules, this line instructs the server to load the specified driver  
# module. Even when not using loadable driver modules, this line  
# indicates which driver should interpret the information in this section.  
  
# Driver "vga"  
  
# The chipset line is optional in most cases. It can be used to override  
# the driver's chipset detection, and should not normally be specified.  
  
# Chipset "generic"  
  
# Various other lines can be specified to override the driver's automatic  
# detection code. In most cases they are not needed.  
  
# VideoRam 256  
# Clocks 25.2 28.3  
  
# The BusID line is used to specify which of possibly multiple devices  
# this section is intended for. When this line isn't present, a device  
# section can only match up with the primary video device. For PCI  
# devices a line like the following could be used. This line should not  
# normally be included unless there is more than one video device  
# intalled.  
  
# BusID "PCI:0:10:0"  
  
# Various option lines can be added here as required. Some options  
# are more appropriate in Screen sections, Display subsections or even  
# Monitor sections.  
  
# Option "hw cursor" "off"
```

```

#EndSection

Section "DRI"
    Mode    0666
EndSection

Section "Device"
    Identifier "SGI SG-0"
    Driver     "fglrx"
    BusId      "PCI:2:0:0"
# === Screen Management ===
    Option "DesktopSetup"      "0x00000000"
    Option "MonitorLayout"     "AUTO, AUTO"
    Option "IgnoreEDID"        "off"
    Option "HSync2"             "unspecified"
    Option "VRefresh2"         "unspecified"
    Option "ScreenOverlap"     "0"
# === ImageSync options: none, default, (brickid:bus:slot) [ex: "(1c10:0:2)"]
    Option "ImageSync"         "default"
# === OpenGL specific profiles/settings ===
    Option "Capabilities"       "0x00000000"
# === OpenGL Overlay ===
    Option "OpenGLOverlay"     "off"
# === Pseudo Color Visuals (8-bit visuals) ===
    Option "PseudoColorVisuals" "off"
# === QBS Management ===
    Option "Stereo"             "off"
    Option "StereoSyncEnable"   "1"
# === FSAA Management ===
    Option "FSAAScale"         "0"
    Option "FSAADisableGamma"  "no"
    Option "FSAACustomizeMSPos" "yes"
    Option "FSAAMSPosX0"       "0.250000"
    Option "FSAAMSPosY0"       "0.416666"
    Option "FSAAMSPosX1"       "0.083333"
    Option "FSAAMSPosY1"       "0.083333"
    Option "FSAAMSPosX2"       "0.416666"
    Option "FSAAMSPosY2"       "0.750000"
    Option "FSAAMSPosX3"       "0.750000"
    Option "FSAAMSPosY3"       "0.916666"
    Option "FSAAMSPosX4"       "0.583333"
    Option "FSAAMSPosY4"       "0.250000"
    Option "FSAAMSPosX5"       "0.916666"
    Option "FSAAMSPosY5"       "0.583333"
EndSection

Section "Device"
    Identifier "SGI SG-1"

```

```

Driver    "fglrx"
BusId     "PCI:2:1:0"
# ==== Screen Management ====
Option "DesktopSetup"      "0x00000000"
Option "MonitorLayout"     "AUTO, AUTO"
Option "IgnoreEDID"        "off"
Option "HSync2"             "unspecified"
Option "VRefresh2"         "unspecified"
Option "ScreenOverlap"     "0"
# ==== ImageSync options: none, default, (brickid:bus:slot) [ex: "(1c10:0:2)"]
Option "ImageSync"         "default"
# ==== OpenGL specific profiles/settings ====
Option "Capabilities"      "0x00000000"
# ==== OpenGL Overlay ====
Option "OpenGLOverlay"     "off"
# ==== Pseudo Color Visuals (8-bit visuals) ====
Option "PseudoColorVisuals" "off"
# ==== QBS Management ====
Option "Stereo"             "off"
Option "StereoSyncEnable"  "1"
# ==== FSAA Management ====
Option "FSAAScale"         "0"
Option "FSAADisableGamma"  "no"
Option "FSAACustomizeMSPos" "yes"
Option "FSAAMSPosX0"       "0.250000"
Option "FSAAMSPosY0"       "0.416666"
Option "FSAAMSPosX1"       "0.083333"
Option "FSAAMSPosY1"       "0.083333"
Option "FSAAMSPosX2"       "0.416666"
Option "FSAAMSPosY2"       "0.750000"
Option "FSAAMSPosX3"       "0.750000"
Option "FSAAMSPosY3"       "0.916666"
Option "FSAAMSPosX4"       "0.583333"
Option "FSAAMSPosY4"       "0.250000"
Option "FSAAMSPosX5"       "0.916666"
Option "FSAAMSPosY5"       "0.583333"
EndSection

```

```

Section "Device"
Identifier "SGI SG-2"
Driver    "fglrx"
BusId     "PCI:2:2:0"
# ==== Screen Management ====
Option "DesktopSetup"      "0x00000000"
Option "MonitorLayout"     "AUTO, AUTO"
Option "IgnoreEDID"        "off"
Option "HSync2"             "unspecified"
Option "VRefresh2"         "unspecified"

```

```

Option "ScreenOverlap"      "0"
# === ImageSync options: none, default, (brickid:bus:slot) [ex: "(1c10:0:2)"]
Option "ImageSync"         "default"
# === OpenGL specific profiles/settings ===
Option "Capabilities"       "0x00000000"
# === OpenGL Overlay ===
Option "OpenGLOverlay"     "off"
# === Pseudo Color Visuals (8-bit visuals) ===
Option "PseudoColorVisuals" "off"
# === QBS Management ===
Option "Stereo"             "off"
Option "StereoSyncEnable"  "1"
# === FSAAs Management ===
Option "FSAAScale"         "0"
Option "FSAADisableGamma"  "no"
Option "FSAACustomizeMSPos" "yes"
Option "FSAAMSPosX0"       "0.250000"
Option "FSAAMSPosY0"       "0.416666"
Option "FSAAMSPosX1"       "0.083333"
Option "FSAAMSPosY1"       "0.083333"
Option "FSAAMSPosX2"       "0.416666"
Option "FSAAMSPosY2"       "0.750000"
Option "FSAAMSPosX3"       "0.750000"
Option "FSAAMSPosY3"       "0.916666"
Option "FSAAMSPosX4"       "0.583333"
Option "FSAAMSPosY4"       "0.250000"
Option "FSAAMSPosX5"       "0.916666"
Option "FSAAMSPosY5"       "0.583333"
EndSection

```

Section "Device"

```

Identifier "SGI SG-3"
Driver     "fglrx"
BusId      "PCI:2:3:0"
# === Screen Management ===
Option "DesktopSetup"      "0x00000000"
Option "MonitorLayout"     "AUTO, AUTO"
Option "IgnoreEDID"        "off"
Option "HSync2"             "unspecified"
Option "VRefresh2"         "unspecified"
Option "ScreenOverlap"     "0"
# === ImageSync options: none, default, (brickid:bus:slot) [ex: "(1c10:0:2)"]
Option "ImageSync"         "default"
# === OpenGL specific profiles/settings ===
Option "Capabilities"       "0x00000000"
# === OpenGL Overlay ===
Option "OpenGLOverlay"     "off"
# === Pseudo Color Visuals (8-bit visuals) ===

```

```

Option "PseudoColorVisuals"    "off"
# === QBS Management ===
Option "Stereo"                "off"
Option "StereoSyncEnable"      "1"
# === FSAA Management ===
Option "FSAAScale"             "0"
Option "FSAADisableGamma"      "no"
Option "FSAACustomizeMSPos"    "yes"
Option "FSAAMSPosX0"           "0.250000"
Option "FSAAMSPosY0"           "0.416666"
Option "FSAAMSPosX1"           "0.083333"
Option "FSAAMSPosY1"           "0.083333"
Option "FSAAMSPosX2"           "0.416666"
Option "FSAAMSPosY2"           "0.750000"
Option "FSAAMSPosX3"           "0.750000"
Option "FSAAMSPosY3"           "0.916666"
Option "FSAAMSPosX4"           "0.583333"
Option "FSAAMSPosY4"           "0.250000"
Option "FSAAMSPosX5"           "0.916666"
Option "FSAAMSPosY5"           "0.583333"
EndSection

```

```

# *****
# Screen sections.
# *****

```

```

# Any number of screen sections may be present. Each describes
# the configuration of a single screen. A single specific screen section
# may be specified from the X server command line with the "-screen"
# option.

```

```

#Section "Screen"

```

```

# The Identifier, Device and Monitor lines must be present

```

```

# Identifier "Screen 1"
# Device  "Generic VGA"
# Monitor "Generic Monitor"
#

```

```

# The favoured Depth and/or Bpp may be specified here

```

```

# DefaultDepth 8
#
# SubSection "Display"
#   Depth      8
#   Modes      "640x480"
#   ViewPort   0 0

```

```

#   Virtual      800 600
#   EndSubSection
#
#   SubSection "Display"
#     Depth      4
#     Modes       "640x480"
#   EndSubSection
#
#   SubSection "Display"
#     Depth      1
#     Modes       "640x480"
#   EndSubSection
#
#EndSection

Section "Screen"
  Identifier      "Screen SG-0"
  Device          "SGI SG-0"
  Monitor         "Generic Monitor"
  DefaultDepth    24
  SubSection      "Display"
    Depth 24
    Modes "1280x1024"
  EndSubSection
EndSection

Section "Screen"
  Identifier      "Screen SG-1"
  Device          "SGI SG-1"
  Monitor         "Generic Monitor"
  DefaultDepth    24
  SubSection      "Display"
    Depth 24
    Modes "1280x1024"
  EndSubSection
EndSection

Section "Screen"
  Identifier      "Screen SG-2"
  Device          "SGI SG-2"
  Monitor         "Generic Monitor"
  DefaultDepth    24
  SubSection      "Display"
    Depth 24
    Modes "1280x1024"
  EndSubSection
EndSection

```

```

Section "Screen"
    Identifier    "Screen SG-3"
    Device        "SGI SG-3"
    Monitor       "Generic Monitor"
    DefaultDepth  24
    SubSection    "Display"
        Depth    24
        Modes    "1280x1024"
    EndSubSection
EndSection

```

```

# *****
# ServerLayout sections.
# *****

```

```

# Any number of ServerLayout sections may be present. Each describes
# the way multiple screens are organised. A specific ServerLayout
# section may be specified from the X server command line with the
# "-layout" option. In the absence of this, the first section is used.
# When now ServerLayout section is present, the first Screen section
# is used alone.

```

```

Section "ServerLayout"

```

```

# The Identifier line must be present

```

```

    Identifier "Main Layout"

```

```

# Each Screen line specifies a Screen section name, and optionally
# the relative position of other screens. The four names after
# primary screen name are the screens to the top, bottom, left and right
# of the primary screen. In this example, screen 2 is located to the
# right of screen 1.

```

```

    Screen "Screen SG-0"    ""    ""    ""    "Screen SG-1"
    Screen "Screen SG-1"    ""    ""    "Screen SG-0" "Screen SG-2"
    Screen "Screen SG-2"    ""    ""    "Screen SG-1" "Screen SG-3"
    Screen "Screen SG-3"    ""    ""    "Screen SG-2"    ""

```

```

# Each InputDevice line specifies an InputDevice section name and
# optionally some options to specify the way the device is to be
# used. Those options include "CorePointer", "CoreKeyboard" and
# "SendCoreEvents". In this example, "Mouse1" is the core pointer,
# and "Mouse2" is an extended input device that also generates core
# pointer events (i.e., both mice will move the standard pointer).

```

```

    InputDevice "Mouse1" "CorePointer"
    InputDevice "Keyboard1" "CoreKeyboard"

```

EndSection

A.2 XF86Config-4 CAVE configuration file

Below is the listing for an *XF86Config-4* configuration designed for displaying in CAVE display configuration. It is running at 1400x1050@96 Hz (the same combination used by the projectors). The configuration file support enabled for 3D Stereo and disabled for FSAA. The primary channels are disabled but the secondary channels are enabled. They are configured for displaying to CRT-based displays. Channel cloning and desktop extension are also disabled.

The configuration file is as follows:

```
# $XFree86: xc/programs/Xserver/hw/xfree86/XF86Conf.cpp,v 3.44 2001/12/17
20:52:29 dawes Exp $
#
# Copyright (c) 1994-1998 by The XFree86 Project, Inc.
#
# Permission is hereby granted, free of charge, to any person obtaining a
# copy of this software and associated documentation files (the "Software"),
# to deal in the Software without restriction, including without limitation
# the rights to use, copy, modify, merge, publish, distribute, sublicense,
# and/or sell copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in
# all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
# KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
# MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO
# EVENT SHALL
# THE XFREE86 PROJECT BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF
# OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
# DEALINGS IN THE
# SOFTWARE.
#
# Except as contained in this notice, the name of the XFree86 Project shall
# not be used in advertising or otherwise to promote the sale, use or other
# dealings in this Software without prior written authorization from the
# XFree86 Project.
#
```

```

# $XConsortium: XF86Conf.cpp /main/22 1996/10/23 11:43:51 kaleb $

# *****
# This is a sample configuration file only, intended to illustrate
# what a config file might look like. Refer to the XF86Config(4/5)
# man page for details about the format of this file. This man page
# is installed as /usr/X11R6/man/man4/XF86Config.4
# *****

# The ordering of sections is not important in version 4.0 and later.

# *****
# Files section. This allows default font and rgb paths to be set
# *****

Section "Files"

# The location of the RGB database. Note, this is the name of the
# file minus the extension (like ".txt" or ".db"). There is normally
# no need to change the default.

    RgbPath        "/usr/lib/X11/rgb"

# Multiple FontPath entries are allowed (which are concatenated together),
# as well as specifying multiple comma-separated entries in one FontPath
# command (or a combination of both methods)

    FontPath       "/usr/lib/X11/fonts/100dpi:unscaled"
    FontPath       "/usr/lib/X11/fonts/75dpi:unscaled"
    FontPath       "/usr/lib/X11/fonts/misc/"
    FontPath       "/usr/lib/X11/fonts/Type1/"
    FontPath       "/usr/lib/X11/fonts/Speedo/"
    FontPath       "/usr/lib/X11/fonts/CID/"
    FontPath       "/usr/lib/X11/fonts/100dpi/"
    FontPath       "/usr/lib/X11/fonts/75dpi/"

# ModulePath can be used to set a search path for the X server modules.
# The default path is shown here.

    ModulePath     "/usr/X11R6/lib/modules"

EndSection

# *****
# Module section -- this is an optional section which is used to specify
# which run-time loadable modules to load when the X server starts up.
# *****

```

Section "Module"

This loads the DBE extension module.

Load "dbe"

This loads the miscellaneous extensions module, and disables
initialisation of the XFree86-DGA extension within that module.

SubSection "extmod"
Option "omit xfree86-dga"
EndSubSection

This loads the the various modules required for X server operation

Load "type1"
Load "freetype"
Load "drm"
Load "fb"
Load "xaa"
Load "glx"
Load "dri"
Load "fglrx_glx"

EndSection

Server flags section. This contains various server-wide Options.

Section "ServerFlags"

Uncomment this to cause a core dump at the spot where a signal is
received. This may leave the console in an unusable state, but may
provide a better stack trace in the core dump to aid in debugging

Option "NoTrapSignals"

Uncomment this to disable the <Ctrl><Alt><BS> server abort sequence
This allows clients to receive this key event.

Option "DontZap"

Uncomment this to disable the <Ctrl><Alt><KP_+>/<KP_-> mode switching
sequences. This allows clients to receive these key events.

```

# Option      "DontZoom"

# Uncomment this to disable tuning with the xvidtune client. With
# it the client can still run and fetch card and monitor attributes,
# but it will not be allowed to change them. If it tries it will
# receive a protocol error.

# Option      "DisableVidModeExtension"

# Uncomment this to enable the use of a non-local xvidtune client.

# Option      "AllowNonLocalXvidtune"

# Uncomment this to disable dynamically modifying the input device
# (mouse and keyboard) settings.

# Option      "DisableModInDev"

# Uncomment this to enable the use of a non-local client to
# change the keyboard or mouse settings (currently only xset).

# Option      "AllowNonLocalModInDev"

# Set the basic blanking screen saver timeout.

Option      "blank time"    "10"    # 10 minutes

# Set the DPMS timeouts.  These are set here because they are global
# rather than screen-specific.  These settings alone don't enable DPMS.
# It is enabled per-screen (or per-monitor), and even then only when
# the driver supports it.

Option      "standby time"  "20"
Option      "suspend time"  "30"
Option      "off time"      "60"

# On some platform the server needs to estimate the sizes of PCI
# memory and pio ranges. This is done by assuming that PCI ranges
# don't overlap. Some broken BIOSes tend to set ranges of inactive
# devices wrong. Here one can adjust how aggressive the assumptions
# should be. Default is 0.

# Option      "EstimateSizesAggressively" "0"

EndSection

# *****
# Input devices

```

```

# *****

# *****
# Core keyboard's InputDevice section
# *****

Section "InputDevice"

    Identifier    "Keyboard1"
    Driver        "keyboard"

# For most OSs the protocol can be omitted (it defaults to "Standard").
# When using XQUEUE (only for SVR3 and 1, but not Solaris), comment
# out the above line, and uncomment the following line.

# Option        "Protocol"        "Xqueue"

# Set the keyboard auto repeat parameters. Not all platforms implement
# this.

    Option        "AutoRepeat"    "500 50"

# Specify which keyboard LEDs can be user-controlled (eg, with xset(1)).

# Option        "Xleds" "1 2 3"

# To disable the XKEYBOARD extension, uncomment XkbDisable.

# Option        "XkbDisable"

# To customise the XKB settings to suit your keyboard, modify the
# lines below (which are the defaults). For example, for a European
# keyboard, you will probably want to use one of:
#
# Option        "XkbModel"        "pc102"
# Option        "XkbModel"        "pc105"
#
# If you have a Microsoft Natural keyboard, you can use:
#
# Option        "XkbModel"        "microsoft"
#
# If you have a US "windows" keyboard you will want:
#
# Option        "XkbModel"        "pc104"
#
# Then to change the language, change the Layout setting.
# For example, a german layout can be obtained with:
#

```

```

# Option      "XkbLayout"    "de"
#
# or:
#
# Option      "XkbLayout"    "de"
# Option      "XkbVariant"   "nodeadkeys"
#
# If you'd like to switch the positions of your capslock and
# control keys, use:
#
# Option      "XkbOptions"   "ctrl:swapcaps"

# These are the default XKB settings for XFree86
#
# Option      "XkbRules"     "xfree86"
# Option      "XkbModel"     "pc101"
# Option      "XkbLayout"    "us"
# Option      "XkbVariant"   ""
# Option      "XkbOptions"   ""

EndSection

# *****
# Core Pointer's InputDevice section
# *****

Section "InputDevice"

# Identifier and driver

    Identifier "Mouse1"
    Driver     "mouse"

# The mouse protocol and device.  The device is normally set to /dev/mouse,
# which is usually a symbolic link to the real device.

    Option     "Protocol"     "PS/2"
    Option     "Device"       "/dev/mouse"

# On platforms where PnP mouse detection is supported the following
# protocol setting can be used when using a newer PnP mouse:

# Option     "Protocol"     "Auto"

# When using mouse connected to a PS/2 port (aka "MousePort), set the
# the protocol as follows.  On some platforms some other settings may

```

```

# be available.

# Option "Protocol" "PS/2"

# When using XQUEUE (only for SVR3 and 1, but not Solaris), use
# the following instead of any of the lines above. The Device line
# is not required in this case.

# Option "Protocol" "Xqueue"

# Baudrate and SampleRate are only for some older Logitech mice. In
# almost every case these lines should be omitted.

# Option "BaudRate" "9600"
# Option "SampleRate" "150"

# Emulate3Buttons is an option for 2-button mice
# Emulate3Timeout is the timeout in milliseconds (default is 50ms)

# Option "Emulate3Buttons"
# Option "Emulate3Timeout" "50"

# ChordMiddle is an option for some 3-button Logitech mice, or any
# 3-button mouse where the middle button generates left+right button
# events.

# Option "ChordMiddle"

EndSection

# Some examples of extended input devices

# Section "InputDevice"
# Identifier "spaceball"
# Driver "magellan"
# Option "Device" "/dev/cua0"
# EndSection
#
# Section "InputDevice"
# Identifier "spaceball2"
# Driver "spaceorb"
# Option "Device" "/dev/cua0"
# EndSection
#
# Section "InputDevice"
# Identifier "touchscreen0"
# Driver "microtouch"
# Option "Device" "/dev/ttyS0"

```

```

# Option      "MinX"          "1412"
# Option      "MaxX"          "15184"
# Option      "MinY"          "15372"
# Option      "MaxY"          "1230"
# Option      "ScreenNumber"  "0"
# Option      "ReportingMode"  "Scaled"
# Option      "ButtonNumber"   "1"
# Option      "SendCoreEvents"
# EndSection
#
# Section "InputDevice"
# Identifier   "touchscreen1"
# Driver       "elo2300"
# Option      "Device"        "/dev/ttyS0"
# Option      "MinX"          "231"
# Option      "MaxX"          "3868"
# Option      "MinY"          "3858"
# Option      "MaxY"          "272"
# Option      "ScreenNumber"  "0"
# Option      "ReportingMode"  "Scaled"
# Option      "ButtonThreshold" "17"
# Option      "ButtonNumber"   "1"
# Option      "SendCoreEvents"
# EndSection

# *****
# Monitor section
# *****

# Any number of monitor sections may be present

Section "Monitor"

# The identifier line must be present.

    Identifier   "Generic Monitor"

# HorizSync is in kHz unless units are specified.
# HorizSync may be a comma separated list of discrete values, or a
# comma separated list of ranges of values.
# NOTE: THE VALUES HERE ARE EXAMPLES ONLY. REFER TO YOUR
MONITOR'S
# USER MANUAL FOR THE CORRECT NUMBERS.

# HorizSync 31.5 # typical for a single frequency fixed-sync monitor
# HorizSync 30-64 # multisync
# HorizSync 31.5, 35.2 # multiple fixed sync frequencies
# HorizSync 15-25, 30-50 # multiple ranges of sync frequencies

```

HorizSync 30-130 # multisync

VertRefresh is in Hz unless units are specified.
VertRefresh may be a comma separated list of discrete values, or a
comma separated list of ranges of values.
NOTE: THE VALUES HERE ARE EXAMPLES ONLY. REFER TO YOUR
MONITOR'S
USER MANUAL FOR THE CORRECT NUMBERS.

VertRefresh 60 # typical for a single frequency fixed-sync monitor

VertRefresh 50-100 # multisync
VertRefresh 60, 65 # multiple fixed sync frequencies
VertRefresh 40-50, 80-100 # multiple ranges of sync frequencies

VertRefresh 50-160 # multisync

Modes can be specified in two formats. A compact one-line format, or
a multi-line format.

A generic VGA 640x480 mode (hsync = 31.5kHz, refresh = 60Hz)
These two are equivalent

ModeLine "640x480" 25.175 640 664 760 800 480 491 493 525

Mode "640x480"
DotClock 25.175
HTimings 640 664 760 800
VTimings 480 491 493 525
EndMode

These two are equivalent

ModeLine "1024x768i" 45 1024 1048 1208 1264 768 776 784 817 Interlace

Mode "1024x768i"
DotClock 45
HTimings 1024 1048 1208 1264
VTimings 768 776 784 817
Flags "Interlace"
EndMode

ModeLine "1280x1024_96" 164.966 1280 1300 1460 1600 1024 1026 1029 1074
ModeLine "1280x1024_120" 233.793 1280 1384 1528 1776 1024 1025 1028 1097
Modeline "1920x1200_60p" 157.39 1920 1950 2072 2112 1200 1203 1206 1242
Modeline "1400x1050_100" 214.39 1400 1512 1664 1928 1050 1051 1054 1112

```

# framelock timings, identified by the trailing 'f'
# (==> denotes available source video framelock formats)
# (example: setmon -L ntsc 1280x1024_5994f )

Modeline "1280x1024_50f" 104 1280 1336 1472 1664 1024 1027 1030 1250 -hsync -
vsync
# ==> 625i_50 (alias: pal), 1080p_25, 1080i_50

Modeline "1280x1024_5994f" 115.2 1280 1296 1440 1560 1024 1027 1030 1232 -hsync
-vsync
# ==> 525i_5994 (alias: ntsc), 720p_5994, 1080p_2997, 1080i_5994

Modeline "1280x1024_60f" 121.5 1280 1396 1540 1800 1024 1027 1030 1125 -hsync
-vsync
# ==> 720p_60, 1080p_30, 1080i_60

Modeline "1920x1154_48f" 144 1920 1968 2160 2400 1154 1157 1160 1250 -hsync -
vsync
# ==> 1080p_24, 1080i_48, this is a cvt modified format

Modeline "1920x1154_50f" 144 1920 1968 2160 2400 1154 1157 1160 1200 -hsync -
vsync
# ==> 625i_50 (alias: pal), 1080p_25, 1080i_50, this is a cvt modified format

Modeline "1920x1154_5994f" 192 1920 1992 2192 2464 1154 1157 1160 1300 -hsync -
vsync
# ==> 525i_5994 (alias: ntsc), 720p_5994, 1080p_2997, 1080i_5994, this is a cvt
modified format

Modeline "1920x1154_60f" 172.8 1920 1968 2160 2400 1154 1157 1160 1200 -hsync
-vsync
# ==> 720p_60, 1080p_30, 1080i_60, this is a cvt modified format

Modeline "1920x1200_60f" 180 1920 1968 2160 2400 1200 1203 1206 1250 -hsync -
vsync
# ==> 720p_60, 1080p_30, 1080i_60, this is a cvt modified format

Modeline "1920x1200_50f" 150 1920 1968 2160 2400 1200 1203 1206 1250 -hsync -
vsync
# ==> 1080p_25, 1080i_50, this is a cvt modified format

# Framelock timings with reduced blanking for formats frequency higher than dvi10 max
freq(165Mhz)

Modeline "1920x1154_5994f_p" 156 1920 1968 2000 2200 1154 1157 1160 1283 -
hsync -vsync
# ==> 525i_5994 (alias: ntsc), 720p_5994, 1080p_2997, 1080i_5994

```

```
Modeline "1920x1154_60f_p" 162 1920 1968 2000 2160 1154 1157 1160 1250 -hsync -vsync
```

```
# ==> 720p_60, 1080p_30, 1080i_60, this is a cvt modified format
```

```
Modeline "1920x1200_60f_p" 162 1920 1968 2000 2160 1200 1203 1206 1250 -hsync -vsync
```

```
# ==> 720p_60, 1080p_30, 1080i_60, this is a cvt modified format
```

```
#
```

```
# The following are standard VESA formats modified to be usable for a genlock source  
# through the "C SYNC" cable. The "C SYNC" cable combines sync's into a composite  
sync,
```

```
# but requires the sync levels to be negative. These ModeLines are distinguished by
```

```
# an "n" (for "negative" sync levels) appended to the ModeLine Name.
```

```
#
```

```
ModeLine "1280x1024_60n" 108 1280 1328 1440 1688 1024 1025 1028 1066 -hsync -vsync
```

```
ModeLine "1280x1024_75n" 135 1280 1296 1440 1688 1024 1025 1028 1066 -hsync -vsync
```

```
ModeLine "1280x1024_85n" 157 1280 1344 1504 1728 1024 1025 1028 1072 -hsync -vsync
```

```
ModeLine "1600x1200_60n" 162 1600 1664 1856 2160 1200 1201 1204 1250 -hsync -vsync
```

```
ModeLine "1600x1200_75n" 202 1600 1664 1856 2160 1200 1201 1204 1250 -hsync -vsync
```

```
ModeLine "1600x1200_85n" 229 1600 1664 1856 2160 1200 1201 1204 1250 -hsync -vsync
```

```
ModeLine "1920x1080_72n" 211 1920 2056 2264 2608 1080 1081 1084 1126 -hsync -vsync
```

```
ModeLine "1920x1200_60n" 193 1920 2048 2256 2592 1200 1201 1203 1242 -hsync -vsync
```

```
ModeLine "1920x1200_75" 246 1920 2064 2272 2624 1200 1201 1203 1253 +hsync +vsync
```

```
Modeline "1400x1050_100" 214 1400 1512 1664 1928 1050 1051 1054 1112 -hsync +vsync
```

```
ModeLine "1400x1050_96" 205 1400 1512 1664 1928 1050 1051 1054 1110 -hsync +vsync
```

```
ModeLine "1920x1440_75" 297 1920 2064 2288 2640 1440 1441 1444 1500 -hsync +vsync
```

```
ModeLine "1920x1200_75" 246 1920 2064 2272 2624 1200 1201 1203 1253 +hsync +vsync
```

```
# If a monitor has DPMS support, that can be indicated here. This will
```

```
# enable DPMS when the monitor is used with drivers that support it.
```

```

# Option      "dpms"

# If a monitor requires that the sync signals be superimposed on the
# green signal, the following option will enable this when used with
# drivers that support it. Only a relatively small range of hardware
# (and drivers) actually support this.

# Option      "sync on green"

EndSection

# *****
# Graphics device section
# *****

# Any number of graphics device sections may be present

#Section "Device"

# The Identifier must be present.

# Identifier   "Generic VGA"

# The Driver line must be present. When using run-time loadable driver
# modules, this line instructs the server to load the specified driver
# module. Even when not using loadable driver modules, this line
# indicates which driver should interpret the information in this section.

# Driver      "vga"

# The chipset line is optional in most cases. It can be used to override
# the driver's chipset detection, and should not normally be specified.

# Chipset     "generic"

# Various other lines can be specified to override the driver's automatic
# detection code. In most cases they are not needed.

# VideoRam   256
# Clocks     25.2 28.3

# The BusID line is used to specify which of possibly multiple devices
# this section is intended for. When this line isn't present, a device
# section can only match up with the primary video device. For PCI
# devices a line like the following could be used. This line should not
# normally be included unless there is more than one video device
# intalled.

```

```

# BusID      "PCI:0:10:0"

# Various option lines can be added here as required.  Some options
# are more appropriate in Screen sections, Display subsections or even
# Monitor sections.

# Option      "hw cursor" "off"

#EndSection

Section "DRI"
    Mode      0666
EndSection

Section "Device"
    Identifier  "SGI SG-0"
    Driver      "fglrx"
    BusId       "PCI:2:0:0"
# === Screen Management ===
    Option "DesktopSetup"      "0x00000000"
    Option "MonitorLayout"     "NONE, CRT"
    Option "IgnoreEDID"       "off"
    Option "HSync2"           "unspecified"
    Option "VRefresh2"        "unspecified"
    Option "ScreenOverlap"    "0"
# === ImageSync options: none, default, (brickid:bus:slot) [ex: "(1c10:0:2)"]
    Option "ImageSync"        "default"
# === OpenGL specific profiles/settings ===
    Option "Capabilities"     "0x00000000"
# === OpenGL Overlay ===
    Option "OpenGLOverlay"    "off"
# === Pseudo Color Visuals (8-bit visuals) ===
    Option "PseudoColorVisuals" "off"
# === QBS Management ===
    Option "Stereo"           "1"
    Option "StereoSyncEnable" "1"
# === FSAA Management ===
    Option "FSAAScale"        "0"
    Option "FSAADisableGamma" "no"
    Option "FSAACustomizeMSPos" "yes"
    Option "FSAAMSPosX0"      "0.250000"
    Option "FSAAMSPosY0"      "0.416666"
    Option "FSAAMSPosX1"      "0.083333"
    Option "FSAAMSPosY1"      "0.083333"
    Option "FSAAMSPosX2"      "0.416666"
    Option "FSAAMSPosY2"      "0.750000"
    Option "FSAAMSPosX3"      "0.750000"
    Option "FSAAMSPosY3"      "0.916666"

```

```

Option "FSAAMSPosX4"      "0.583333"
Option "FSAAMSPosY4"      "0.250000"
Option "FSAAMSPosX5"      "0.916666"
Option "FSAAMSPosY5"      "0.583333"
EndSection

```

Section "Device"

```

Identifier "SGI SG-1"
Driver     "fglrx"
BusId     "PCI:2:1:0"
# === Screen Management ===
Option "DesktopSetup"      "0x00000000"
Option "MonitorLayout"     "NONE, CRT"
Option "IgnoreEDID"        "off"
Option "HSync2"            "unspecified"
Option "VRefresh2"         "unspecified"
Option "ScreenOverlap"     "0"
# === ImageSync options: none, default, (brickid:bus:slot) [ex: "(1c10:0:2)"]
Option "ImageSync"         "default"
# === OpenGL specific profiles/settings ===
Option "Capabilities"      "0x00000000"
# === OpenGL Overlay ===
Option "OpenGLOverlay"     "off"
# === Pseudo Color Visuals (8-bit visuals) ===
Option "PseudoColorVisuals" "off"
# === QBS Management ===
Option "Stereo"            "1"
Option "StereoSyncEnable"  "1"
# === FSAAs Management ===
Option "FSAAScale"         "0"
Option "FSAADisableGamma"  "no"
Option "FSAACustomizeMSPos" "yes"
Option "FSAAMSPosX0"       "0.250000"
Option "FSAAMSPosY0"       "0.416666"
Option "FSAAMSPosX1"       "0.083333"
Option "FSAAMSPosY1"       "0.083333"
Option "FSAAMSPosX2"       "0.416666"
Option "FSAAMSPosY2"       "0.750000"
Option "FSAAMSPosX3"       "0.750000"
Option "FSAAMSPosY3"       "0.916666"
Option "FSAAMSPosX4"       "0.583333"
Option "FSAAMSPosY4"       "0.250000"
Option "FSAAMSPosX5"       "0.916666"
Option "FSAAMSPosY5"       "0.583333"
EndSection

```

```

Section "Device"
    Identifier    "SGI SG-2"
    Driver        "fglrx"
    BusId        "PCI:2:2:0"
# === Screen Management ===
    Option "DesktopSetup"    "0x00000000"
    Option "MonitorLayout"   "NONE, CRT"
    Option "IgnoreEDID"     "off"
    Option "HSync2"         "unspecified"
    Option "VRefresh2"      "unspecified"
    Option "ScreenOverlap"  "0"
# === ImageSync options: none, default, (brickid:bus:slot) [ex: "(1c10:0:2)"]
    Option "ImageSync"      "default"
# === OpenGL specific profiles/settings ===
    Option "Capabilities"    "0x00000000"
# === OpenGL Overlay ===
    Option "OpenGLOverlay"  "off"
# === Pseudo Color Visuals (8-bit visuals) ===
    Option "PseudoColorVisuals" "off"
# === QBS Management ===
    Option "Stereo"         "1"
    Option "StereoSyncEnable" "1"
# === FSAA Management ===
    Option "FSAAScale"      "0"
    Option "FSAADisableGamma" "no"
    Option "FSAACustomizeMSPos" "yes"
    Option "FSAAMSPosX0"    "0.250000"
    Option "FSAAMSPosY0"    "0.416666"
    Option "FSAAMSPosX1"    "0.083333"
    Option "FSAAMSPosY1"    "0.083333"
    Option "FSAAMSPosX2"    "0.416666"
    Option "FSAAMSPosY2"    "0.750000"
    Option "FSAAMSPosX3"    "0.750000"
    Option "FSAAMSPosY3"    "0.916666"
    Option "FSAAMSPosX4"    "0.583333"
    Option "FSAAMSPosY4"    "0.250000"
    Option "FSAAMSPosX5"    "0.916666"
    Option "FSAAMSPosY5"    "0.583333"
EndSection

```

```

Section "Device"
    Identifier    "SGI SG-3"
    Driver        "fglrx"
    BusId        "PCI:2:3:0"
# === Screen Management ===
    Option "DesktopSetup"    "0x00000000"
    Option "MonitorLayout"   "NONE, CRT"

```

```

Option "IgnoreEDID"          "off"
Option "HSync2"              "unspecified"
Option "VRefresh2"          "unspecified"
Option "ScreenOverlap"      "0"
# === ImageSync options: none, default, (brickid:bus:slot) [ex: "(1c10:0:2)"]
Option "ImageSync"          "default"
# === OpenGL specific profiles/settings ===
Option "Capabilities"        "0x00000000"
# === OpenGL Overlay ===
Option "OpenGLOverlay"      "off"
# === Pseudo Color Visuals (8-bit visuals) ===
Option "PseudoColorVisuals" "off"
# === QBS Management ===
Option "Stereo"              "1"
Option "StereoSyncEnable"    "1"
# === FSAA Management ===
Option "FSAAScale"           "0"
Option "FSAADisableGamma"    "no"
Option "FSAACustomizeMSPos"  "yes"
Option "FSAAMSPosX0"         "0.250000"
Option "FSAAMSPosY0"         "0.416666"
Option "FSAAMSPosX1"         "0.083333"
Option "FSAAMSPosY1"         "0.083333"
Option "FSAAMSPosX2"         "0.416666"
Option "FSAAMSPosY2"         "0.750000"
Option "FSAAMSPosX3"         "0.750000"
Option "FSAAMSPosY3"         "0.916666"
Option "FSAAMSPosX4"         "0.583333"
Option "FSAAMSPosY4"         "0.250000"
Option "FSAAMSPosX5"         "0.916666"
Option "FSAAMSPosY5"         "0.583333"
EndSection

```

```

# *****
# Screen sections.
# *****

```

```

# Any number of screen sections may be present. Each describes
# the configuration of a single screen. A single specific screen section
# may be specified from the X server command line with the "-screen"
# option.

```

```

#Section "Screen"

```

```

# The Identifier, Device and Monitor lines must be present

```

```

# Identifier "Screen 1"
# Device "Generic VGA"
# Monitor "Generic Monitor"
#
# The favoured Depth and/or Bpp may be specified here

# DefaultDepth 8
#
# SubSection "Display"
# Depth 8
# Modes "640x480"
# ViewPort 0 0
# Virtual 800 600
# EndSubsection
#
# SubSection "Display"
# Depth 4
# Modes "640x480"
# EndSubSection
#
# SubSection "Display"
# Depth 1
# Modes "640x480"
# EndSubSection
#
#EndSection

Section "Screen"
Identifier "Screen SG-0"
Device "SGI SG-0"
Monitor "Generic Monitor"
DefaultDepth 24
SubSection "Display"
Depth 24
Modes "1400x1050_96"
EndSubSection
EndSection

Section "Screen"
Identifier "Screen SG-1"
Device "SGI SG-1"
Monitor "Generic Monitor"
DefaultDepth 24
SubSection "Display"
Depth 24
Modes "1400x1050_96"
EndSubSection
EndSection

```

```

Section "Screen"
    Identifier      "Screen SG-2"
    Device          "SGI SG-2"
    Monitor         "Generic Monitor"
    DefaultDepth 24
    SubSection      "Display"
        Depth      24
        Modes       "1400x1050_96"
    EndSubSection
EndSection

```

```

Section "Screen"
    Identifier      "Screen SG-3"
    Device          "SGI SG-3"
    Monitor         "Generic Monitor"
    DefaultDepth 24
    SubSection      "Display"
        Depth      24
        Modes       "1400x1050_96"
    EndSubSection
EndSection

```

```

# *****
# ServerLayout sections.
# *****

```

```

# Any number of ServerLayout sections may be present. Each describes
# the way multiple screens are organised. A specific ServerLayout
# section may be specified from the X server command line with the
# "-layout" option. In the absence of this, the first section is used.
# When now ServerLayout section is present, the first Screen section
# is used alone.

```

```

Section "ServerLayout"

```

```

# The Identifier line must be present

```

```

    Identifier      "Main Layout"

```

```

# Each Screen line specifies a Screen section name, and optionally
# the relative position of other screens. The four names after
# primary screen name are the screens to the top, bottom, left and right
# of the primary screen. In this example, screen 2 is located to the
# right of screen 1.

```

```

    Screen "Screen SG-0" ""           ""           ""           "Screen SG-1"
    Screen "Screen SG-1" ""           "Screen SG-3" "Screen SG-0" "Screen SG-2"

```

```
Screen "Screen SG-2" "" "" "Screen SG-1" ""
Screen "Screen SG-3" "Screen SG-1" "" "Screen SG-2" ""
```

Each InputDevice line specifies an InputDevice section name and
optionally some options to specify the way the device is to be
used. Those options include "CorePointer", "CoreKeyboard" and
"SendCoreEvents". In this example, "Mouse1" is the core pointer,
and "Mouse2" is an extended input device that also generates core
pointer events (i.e., both mice will move the standard pointer).

```
InputDevice "Mouse1" "CorePointer"
InputDevice "Keyboard1" "CoreKeyboard"
EndSection
```

A.3 XF86Config-4 2x2 ViewSonic VP2290b configuration file

Below is the listing for the system a CAVE configured XF86Config-4 configuration file running at 1400x1050@96 Hz (the same type to be used by the CAVE projectors). The displays have been configured for CAVE displaying using a 3x1-display configuration (three horizontal and one floor display) display aspect. The configuration file currently has 3D Stereo support enabled and FSAA support disabled. The monitors are configured for secondary-channel CRT support and are not configured for primary channel activation. Channel cloning is also disabled.

```
# $XFree86: xc/programs/Xserver/hw/xfree86/XF86Conf.cpp,v 3.44 2001/12/17
20:52:29 dawes Exp $
#
# Copyright (c) 1994-1998 by The XFree86 Project, Inc.
#
# Permission is hereby granted, free of charge, to any person obtaining a
# copy of this software and associated documentation files (the "Software"),
# to deal in the Software without restriction, including without limitation
# the rights to use, copy, modify, merge, publish, distribute, sublicense,
# and/or sell copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in
# all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO
EVENT SHALL
# THE XFREE86 PROJECT BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF
# OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE
# SOFTWARE.
#
# Except as contained in this notice, the name of the XFree86 Project shall
# not be used in advertising or otherwise to promote the sale, use or other
# dealings in this Software without prior written authorization from the
# XFree86 Project.
#
# $XConsortium: XF86Conf.cpp /main/22 1996/10/23 11:43:51 kaleb $
```

```

# *****
# This is a sample configuration file only, intended to illustrate
# what a config file might look like. Refer to the XF86Config(4/5)
# man page for details about the format of this file. This man page
# is installed as /usr/X11R6/man/man4/XF86Config.4
# *****

```

The ordering of sections is not important in version 4.0 and later.

```

# *****
# Files section. This allows default font and rgb paths to be set
# *****

```

Section "Files"

The location of the RGB database. Note, this is the name of the
file minus the extension (like ".txt" or ".db"). There is normally
no need to change the default.

```

    RgbPath    "/usr/lib/X11/rgb"

```

Multiple FontPath entries are allowed (which are concatenated together),
as well as specifying multiple comma-separated entries in one FontPath
command (or a combination of both methods)

```

    FontPath  "/usr/lib/X11/fonts/100dpi:unscaled"
    FontPath  "/usr/lib/X11/fonts/75dpi:unscaled"
    FontPath  "/usr/lib/X11/fonts/misc/"
    FontPath  "/usr/lib/X11/fonts/Type1/"
    FontPath  "/usr/lib/X11/fonts/Speedo/"
    FontPath  "/usr/lib/X11/fonts/CID/"
    FontPath  "/usr/lib/X11/fonts/100dpi/"
    FontPath  "/usr/lib/X11/fonts/75dpi/"

```

ModulePath can be used to set a search path for the X server modules.
The default path is shown here.

```

    ModulePath "/usr/X11R6/lib/modules"

```

EndSection

```

# *****
# Module section -- this is an optional section which is used to specify
# which run-time loadable modules to load when the X server starts up.
# *****

```

Section "Module"

```

# This loads the DBE extension module.

    Load    "dbe"

# This loads the miscellaneous extensions module, and disables
# initialisation of the XFree86-DGA extension within that module.

    SubSection "extmod"
        Option "omit xfree86-dga"
    EndSubSection

# This loads the the various modules required for X server operation

    Load    "type1"
    Load    "freetype"
    Load    "drm"
    Load    "fb"
    Load    "xaa"
    Load    "glx"
    Load    "dri"
    Load    "fglrx_glx"

EndSection

# *****
# Server flags section. This contains various server-wide Options.
# *****

Section "ServerFlags"

# Uncomment this to cause a core dump at the spot where a signal is
# received. This may leave the console in an unusable state, but may
# provide a better stack trace in the core dump to aid in debugging

# Option    "NoTrapSignals"

# Uncomment this to disable the <Ctrl><Alt><BS> server abort sequence
# This allows clients to receive this key event.

# Option    "DontZap"

# Uncomment this to disable the <Ctrl><Alt><KP_+>/<KP_-> mode switching
# sequences. This allows clients to receive these key events.

# Option    "DontZoom"

```

```

# Uncomment this to disable tuning with the xvidtune client. With
# it the client can still run and fetch card and monitor attributes,
# but it will not be allowed to change them. If it tries it will
# receive a protocol error.

# Option "DisableVidModeExtension"

# Uncomment this to enable the use of a non-local xvidtune client.

# Option "AllowNonLocalXvidtune"

# Uncomment this to disable dynamically modifying the input device
# (mouse and keyboard) settings.

# Option "DisableModInDev"

# Uncomment this to enable the use of a non-local client to
# change the keyboard or mouse settings (currently only xset).

# Option "AllowNonLocalModInDev"

# Set the basic blanking screen saver timeout.

Option "blank time" "10" # 10 minutes

# Set the DPMS timeouts. These are set here because they are global
# rather than screen-specific. These settings alone don't enable DPMS.
# It is enabled per-screen (or per-monitor), and even then only when
# the driver supports it.

Option "standby time" "20"
Option "suspend time" "30"
Option "off time" "60"

# On some platform the server needs to estimate the sizes of PCI
# memory and pio ranges. This is done by assuming that PCI ranges
# don't overlap. Some broken BIOSes tend to set ranges of inactive
# devices wrong. Here one can adjust how aggressive the assumptions
# should be. Default is 0.

# Option "EstimateSizesAggresively" "0"

EndSection

# *****
# Input devices
# *****

```

```

# *****
# Core keyboard's InputDevice section
# *****

Section "InputDevice"

    Identifier "Keyboard1"
    Driver     "keyboard"

# For most OSs the protocol can be omitted (it defaults to "Standard").
# When using XQUEUE (only for SVR3 and 1, but not Solaris), comment
# out the above line, and uncomment the following line.

# Option     "Protocol"     "Xqueue"

# Set the keyboard auto repeat parameters. Not all platforms implement
# this.

    Option    "AutoRepeat"   "500 50"

# Specify which keyboard LEDs can be user-controlled (eg, with xset(1)).

# Option     "Xleds"        "1 2 3"

# To disable the XKEYBOARD extension, uncomment XkbDisable.

# Option     "XkbDisable"

# To customise the XKB settings to suit your keyboard, modify the
# lines below (which are the defaults). For example, for a European
# keyboard, you will probably want to use one of:
#
# Option     "XkbModel"     "pc102"
# Option     "XkbModel"     "pc105"
#
# If you have a Microsoft Natural keyboard, you can use:
#
# Option     "XkbModel"     "microsoft"
#
# If you have a US "windows" keyboard you will want:
#
# Option     "XkbModel"     "pc104"
#
# Then to change the language, change the Layout setting.
# For example, a german layout can be obtained with:
#
# Option     "XkbLayout"    "de"
#

```

```

# or:
#
# Option  "XkbLayout"    "de"
# Option  "XkbVariant"   "nodeadkeys"
#
# If you'd like to switch the positions of your capslock and
# control keys, use:
#
# Option  "XkbOptions"   "ctrl:swapcaps"

# These are the default XKB settings for XFree86
#
# Option  "XkbRules"     "xfree86"
# Option  "XkbModel"     "pc101"
# Option  "XkbLayout"    "us"
# Option  "XkbVariant"   ""
# Option  "XkbOptions"   ""

EndSection

# *****
# Core Pointer's InputDevice section
# *****

Section "InputDevice"

# Identifier and driver

    Identifier "Mouse1"
    Driver     "mouse"

# The mouse protocol and device.  The device is normally set to /dev/mouse,
# which is usually a symbolic link to the real device.

    Option    "Protocol"    "PS/2"
    Option    "Device"      "/dev/mouse"

# On platforms where PnP mouse detection is supported the following
# protocol setting can be used when using a newer PnP mouse:

# Option    "Protocol"     "Auto"

# When using mouse connected to a PS/2 port (aka "MousePort), set the
# the protocol as follows.  On some platforms some other settings may
# be available.

```

```

# Option "Protocol" "PS/2"

# When using XQUEUE (only for SVR3 and 1, but not Solaris), use
# the following instead of any of the lines above. The Device line
# is not required in this case.

# Option "Protocol" "Xqueue"

# Baudrate and SampleRate are only for some older Logitech mice. In
# almost every case these lines should be omitted.

# Option "BaudRate" "9600"
# Option "SampleRate" "150"

# Emulate3Buttons is an option for 2-button mice
# Emulate3Timeout is the timeout in milliseconds (default is 50ms)

# Option "Emulate3Buttons"
# Option "Emulate3Timeout" "50"

# ChordMiddle is an option for some 3-button Logitech mice, or any
# 3-button mouse where the middle button generates left+right button
# events.

# Option "ChordMiddle"

EndSection

# Some examples of extended input devices

# Section "InputDevice"
# Identifier "spaceball"
# Driver "magellan"
# Option "Device" "/dev/cua0"
# EndSection
#
# Section "InputDevice"
# Identifier "spaceball2"
# Driver "spaceorb"
# Option "Device" "/dev/cua0"
# EndSection
#
# Section "InputDevice"
# Identifier "touchscreen0"
# Driver "microtouch"
# Option "Device" "/dev/ttyS0"
# Option "MinX" "1412"
# Option "MaxX" "15184"

```

```

# Option "MinY"      "15372"
# Option "MaxY"      "1230"
# Option "ScreenNumber" "0"
# Option "ReportingMode" "Scaled"
# Option "ButtonNumber" "1"
# Option "SendCoreEvents"
# EndSection
#
# Section "InputDevice"
# Identifier "touchscreen1"
# Driver "elo2300"
# Option "Device"      "/dev/ttyS0"
# Option "MinX"        "231"
# Option "MaxX"        "3868"
# Option "MinY"        "3858"
# Option "MaxY"        "272"
# Option "ScreenNumber" "0"
# Option "ReportingMode" "Scaled"
# Option "ButtonThreshold" "17"
# Option "ButtonNumber" "1"
# Option "SendCoreEvents"
# EndSection

# *****
# Monitor section
# *****

# Any number of monitor sections may be present

Section "Monitor"

# The identifier line must be present.

Identifier "Generic Monitor"

# HorizSync is in kHz unless units are specified.
# HorizSync may be a comma separated list of discrete values, or a
# comma separated list of ranges of values.
# NOTE: THE VALUES HERE ARE EXAMPLES ONLY. REFER TO YOUR
MONITOR'S
# USER MANUAL FOR THE CORRECT NUMBERS.

# HorizSync 31.5 # typical for a single frequency fixed-sync monitor
# HorizSync 30-64 # multisync
# HorizSync 31.5, 35.2 # multiple fixed sync frequencies
# HorizSync 15-25, 30-50 # multiple ranges of sync frequencies

HorizSync 30-130 # multisync

```

```

# VertRefresh is in Hz unless units are specified.
# VertRefresh may be a comma separated list of discrete values, or a
# comma separated list of ranges of values.
# NOTE: THE VALUES HERE ARE EXAMPLES ONLY. REFER TO YOUR
MONITOR'S
# USER MANUAL FOR THE CORRECT NUMBERS.

#  VertRefresh      60 # typical for a single frequency fixed-sync monitor

#  VertRefresh      50-100    # multisync
#  VertRefresh      60, 65    # multiple fixed sync frequencies
#  VertRefresh      40-50, 80-100 # multiple ranges of sync frequencies

    VertRefresh      50-160 # multisync

# Modes can be specified in two formats. A compact one-line format, or
# a multi-line format.

# A generic VGA 640x480 mode (hsync = 31.5kHz, refresh = 60Hz)
# These two are equivalent

#  ModeLine "640x480" 25.175 640 664 760 800 480 491 493 525

#  Mode "640x480"
#    DotClock      25.175
#    HTimings      640 664 760 800
#    VTimings      480 491 493 525
#  EndMode

# These two are equivalent

#  ModeLine "1024x768i" 45 1024 1048 1208 1264 768 776 784 817 Interlace

#  Mode "1024x768i"
#    DotClock      45
#    HTimings      1024 1048 1208 1264
#    VTimings      768 776 784 817
#    Flags         "Interlace"
#  EndMode

Modeline "1280x1024_50" 104 1280 1328 1440 1664 1024 1025 1028 1250
Modeline "1280x1024_5994" 106.2 1280 1296 1440 1500 1024 1027 1030 1200
Modeline "1280x1024_96" 164.966 1280 1300 1460 1600 1024 1026 1029 1074
Modeline "1280x1024_120" 233.793 1280 1384 1528 1776 1024 1025 1028 1097
Modeline "1920x1154_50" 141 1920 1996 2136 2256 1154 1156 1159 1250
Modeline "1920x1154_5994" 172.8 1920 1935 2075 2240 1154 1167 1170 1287
Modeline "1920x1200_60p" 157.39 1920 1950 2072 2112 1200 1203 1206 1242

```

If a monitor has DPMS support, that can be indicated here. This will
enable DPMS when the monitor is used with drivers that support it.

Option "dpms"

If a monitor requires that the sync signals be superimposed on the
green signal, the following option will enable this when used with
drivers that support it. Only a relatively small range of hardware
(and drivers) actually support this.

Option "sync on green"

EndSection

Section "Monitor"

Identifier "VP2290b"
VendorName "Viewsonic"
ModelName "VP2290b"
HorizSync 22-105
VertRefresh 9-64

#Modeline description: pxlclk Hres Hfp Hsnc Htot Vres Vfp Vsnc Vtot

#-----

#1/4 screen 3840x2400 for 4@960x2400_41 stripes

ModeLine "960x2400_41" 104.78 960 968 1000 1056 2400 2402 2404 2424

#1/4 screen 3840x2400 for 4@1920x1200_41 tiles

ModeLine "1920x1200_41" 104.78 1920 1928 1960 2112 1200 1201 1203 1212

ModeLine "1920x1200_30" 89.67 1920 1992 2184 2448 1200 1201 1204 1221

#1/2 screen 3840x2400 for 2@1920x2400_30

ModeLine "1920x2400_20" 99.54 1920 1928 1980 2044 2400 2402 2405 2435

ModeLine "1920x2400_24" 119.40 1920 1928 1980 2044 2400 2402 2405 2435

ModeLine "1920x2400_25" 125.02 1920 1928 1980 2044 2400 2402 2404 2434

ModeLine "1920x2400_30" 149.31 1920 1928 1980 2044 2400 2402 2405 2435

#full screen 3840x2400 @ 12Hz for single pipe

ModeLine "3840x2400_12" 148.00 3840 3944 4328 4816 2400 2401 2404 2418

endsection

Graphics device section

Any number of graphics device sections may be present

```

#Section "Device"

# The Identifier must be present.

# Identifier "Generic VGA"

# The Driver line must be present. When using run-time loadable driver
# modules, this line instructs the server to load the specified driver
# module. Even when not using loadable driver modules, this line
# indicates which driver should interpret the information in this section.

# Driver "vga"

# The chipset line is optional in most cases. It can be used to override
# the driver's chipset detection, and should not normally be specified.

# Chipset "generic"

# Various other lines can be specified to override the driver's automatic
# detection code. In most cases they are not needed.

# VideoRam 256
# Clocks 25.2 28.3

# The BusID line is used to specify which of possibly multiple devices
# this section is intended for. When this line isn't present, a device
# section can only match up with the primary video device. For PCI
# devices a line like the following could be used. This line should not
# normally be included unless there is more than one video device
# intalled.

# BusID "PCI:0:10:0"

# Various option lines can be added here as required. Some options
# are more appropriate in Screen sections, Display subsections or even
# Monitor sections.

# Option "hw cursor" "off"

#EndSection

Section "DRI"
    Mode 0666
EndSection

Section "Device"
    Identifier "SGI SG-0"
    Driver "fglrx"

```

```

BusId    "PCI:2:0:0"
# === Screen Management ===
Option "DesktopSetup"    "0x00000000"
Option "MonitorLayout"   "TMDS, NONE"
Option "IgnoreEDID"     "off"
Option "HSync2"          "unspecified"
Option "VRefresh2"       "unspecified"
Option "ScreenOverlap"   "0"
# === ImageSync options: none, default, (brickid:bus:slot) [ex: "(1c10:0:2)"]
Option "ImageSync"       "default"
# === OpenGL specific profiles/settings ===
Option "Capabilities"    "0x00000000"
# === OpenGL Overlay ===
Option "OpenGLOverlay"   "off"
# === Pseudo Color Visuals (8-bit visuals) ===
Option "PseudoColorVisuals" "off"
# === QBS Management ===
Option "Stereo"          "1"
Option "StereoSyncEnable" "1"
# === FSAA Management ===
Option "FSAAScale"       "1"
Option "FSAADisableGamma" "no"
Option "FSAACustomizeMSPos" "yes"
Option "FSAAMSPosX0"     "0.250000"
Option "FSAAMSPosY0"     "0.416666"
Option "FSAAMSPosX1"     "0.083333"
Option "FSAAMSPosY1"     "0.083333"
Option "FSAAMSPosX2"     "0.416666"
Option "FSAAMSPosY2"     "0.750000"
Option "FSAAMSPosX3"     "0.750000"
Option "FSAAMSPosY3"     "0.916666"
Option "FSAAMSPosX4"     "0.583333"
Option "FSAAMSPosY4"     "0.250000"
Option "FSAAMSPosX5"     "0.916666"
Option "FSAAMSPosY5"     "0.583333"
EndSection

```

Section "Device"

```

Identifier "SGI SG-1"
Driver    "fglrx"
BusId    "PCI:2:1:0"
# === Screen Management ===
Option "DesktopSetup"    "0x00000000"
Option "MonitorLayout"   "TMDS, NONE"
Option "IgnoreEDID"     "off"
Option "HSync2"          "unspecified"
Option "VRefresh2"       "unspecified"
Option "ScreenOverlap"   "0"

```

```

# === ImageSync options: none, default, (brickid:bus:slot) [ex: "(1c10:0:2)"]
Option "ImageSync"          "default"
# === OpenGL specific profiles/settings ===
Option "Capabilities"       "0x00000000"
# === OpenGL Overlay ===
Option "OpenGLOverlay"     "off"
# === Pseudo Color Visuals (8-bit visuals) ===
Option "PseudoColorVisuals" "off"
# === QBS Management ===
Option "Stereo"             "1"
Option "StereoSyncEnable"  "1"
# === FSAA Management ===
Option "FSAAScale"         "1"
Option "FSAADisableGamma"  "no"
Option "FSAACustomizeMSPos" "yes"
Option "FSAAMSPosX0"       "0.250000"
Option "FSAAMSPosY0"       "0.416666"
Option "FSAAMSPosX1"       "0.083333"
Option "FSAAMSPosY1"       "0.083333"
Option "FSAAMSPosX2"       "0.416666"
Option "FSAAMSPosY2"       "0.750000"
Option "FSAAMSPosX3"       "0.750000"
Option "FSAAMSPosY3"       "0.916666"
Option "FSAAMSPosX4"       "0.583333"
Option "FSAAMSPosY4"       "0.250000"
Option "FSAAMSPosX5"       "0.916666"
Option "FSAAMSPosY5"       "0.583333"
EndSection

```

Section "Device"

```

Identifier "SGI SG-2"
Driver    "fglrx"
BusId     "PCI:2:2:0"
# === Screen Management ===
Option "DesktopSetup"       "0x00000000"
Option "MonitorLayout"     "TMDS, NONE"
Option "IgnoreEDID"        "off"
Option "HSync2"             "unspecified"
Option "VRefresh2"         "unspecified"
Option "ScreenOverlap"     "0"
# === ImageSync options: none, default, (brickid:bus:slot) [ex: "(1c10:0:2)"]
Option "ImageSync"          "default"
# === OpenGL specific profiles/settings ===
Option "Capabilities"       "0x00000000"
# === OpenGL Overlay ===
Option "OpenGLOverlay"     "off"
# === Pseudo Color Visuals (8-bit visuals) ===
Option "PseudoColorVisuals" "off"

```

```

# === QBS Management ===
Option "Stereo"          "1"
Option "StereoSyncEnable"  "1"
# === FSAA Management ===
Option "FSAAScale"       "1"
Option "FSAADisableGamma"  "no"
Option "FSAACustomizeMSPos"  "yes"
Option "FSAAMSPosX0"     "0.250000"
Option "FSAAMSPosY0"     "0.416666"
Option "FSAAMSPosX1"     "0.083333"
Option "FSAAMSPosY1"     "0.083333"
Option "FSAAMSPosX2"     "0.416666"
Option "FSAAMSPosY2"     "0.750000"
Option "FSAAMSPosX3"     "0.750000"
Option "FSAAMSPosY3"     "0.916666"
Option "FSAAMSPosX4"     "0.583333"
Option "FSAAMSPosY4"     "0.250000"
Option "FSAAMSPosX5"     "0.916666"
Option "FSAAMSPosY5"     "0.583333"
EndSection

Section "Device"
Identifier "SGI SG-3"
Driver    "fglrx"
BusId    "PCI:2:3:0"
# === Screen Management ===
Option "DesktopSetup"     "0x00000000"
Option "MonitorLayout"    "TMDS, NONE"
Option "IgnoreEDID"       "off"
Option "HSync2"           "unspecified"
Option "VRefresh2"        "unspecified"
Option "ScreenOverlap"    "0"
# === ImageSync options: none, default, (brickid:bus:slot) [ex: "(1c10:0:2)"]
Option "ImageSync"        "default"
# === OpenGL specific profiles/settings ===
Option "Capabilities"     "0x00000000"
# === OpenGL Overlay ===
Option "OpenGLOverlay"    "off"
# === Pseudo Color Visuals (8-bit visuals) ===
Option "PseudoColorVisuals"  "off"
# === QBS Management ===
Option "Stereo"          "1"
Option "StereoSyncEnable"  "1"
# === FSAA Management ===
Option "FSAAScale"       "1"
Option "FSAADisableGamma"  "no"
Option "FSAACustomizeMSPos"  "yes"
Option "FSAAMSPosX0"     "0.250000"

```

```

Option "FSAAMSPosY0"      "0.416666"
Option "FSAAMSPosX1"      "0.083333"
Option "FSAAMSPosY1"      "0.083333"
Option "FSAAMSPosX2"      "0.416666"
Option "FSAAMSPosY2"      "0.750000"
Option "FSAAMSPosX3"      "0.750000"
Option "FSAAMSPosY3"      "0.916666"
Option "FSAAMSPosX4"      "0.583333"
Option "FSAAMSPosY4"      "0.250000"
Option "FSAAMSPosX5"      "0.916666"
Option "FSAAMSPosY5"      "0.583333"
EndSection

# *****
# Screen sections.
# *****

# Any number of screen sections may be present. Each describes
# the configuration of a single screen. A single specific screen section
# may be specified from the X server command line with the "-screen"
# option.

#Section "Screen"

# The Identifier, Device and Monitor lines must be present

# Identifier "Screen 1"
# Device "Generic VGA"
# Monitor "Generic Monitor"
#
# The favoured Depth and/or Bpp may be specified here

# DefaultDepth 8
#
# SubSection "Display"
# Depth 8
# Modes "640x480"
# ViewPort 0 0
# Virtual 800 600
# EndSubsection
#
# SubSection "Display"
# Depth 4
# Modes "640x480"
# EndSubSection
#
# SubSection "Display"

```

```
# Depth 1
# Modes "640x480"
# EndSubSection
#
#EndSection
```

```
Section "Screen"
    Identifier "Screen SG-0"
    Device "SGI SG-0"
    Monitor "VP2290b"
    DefaultDepth 24
    SubSection "Display"
        Depth 24
        Modes "1920x1200_41"
    EndSubSection
EndSection
```

```
Section "Screen"
    Identifier "Screen SG-1"
    Device "SGI SG-1"
    Monitor "VP2290b"
    DefaultDepth 24
    SubSection "Display"
        Depth 24
        Modes "1920x1200_41"
    EndSubSection
EndSection
```

```
Section "Screen"
    Identifier "Screen SG-2"
    Device "SGI SG-2"
    Monitor "VP2290b"
    DefaultDepth 24
    SubSection "Display"
        Depth 24
        Modes "1920x1200_41"
    EndSubSection
EndSection
```

```
Section "Screen"
    Identifier "Screen SG-3"
    Device "SGI SG-3"
    Monitor "VP2290b"
    DefaultDepth 24
    SubSection "Display"
        Depth 24
        Modes "1920x1200_41"
    EndSubSection
```

EndSection

ServerLayout sections.

Any number of ServerLayout sections may be present. Each describes
the way multiple screens are organised. A specific ServerLayout
section may be specified from the X server command line with the
"-layout" option. In the absence of this, the first section is used.
When now ServerLayout section is present, the first Screen section
is used alone.

Section "ServerLayout"

The Identifier line must be present

Identifier "Main Layout"

Each Screen line specifies a Screen section name, and optionally
the relative position of other screens. The four names after
primary screen name are the screens to the top, bottom, left and right
of the primary screen. In this example, screen 2 is located to the
right of screen 1.

Screen "Screen SG-0" "" "Screen SG-2" "" "Screen SG-1"
Screen "Screen SG-1" "" "Screen SG-3" "Screen SG-0" ""
Screen "Screen SG-2" "Screen SG-0" "" "" "Screen SG-3"
Screen "Screen SG-3" "Screen SG-1" "" "Screen SG-2" ""

Each InputDevice line specifies an InputDevice section name and
optionally some options to specify the way the device is to be
used. Those options include "CorePointer", "CoreKeyboard" and
"SendCoreEvents". In this example, "Mouse1" is the core pointer,
and "Mouse2" is an extended input device that also generates core
pointer events (i.e., both mice will move the standard pointer).

InputDevice "Mouse1" "CorePointer"
InputDevice "Keyboard1" "CoreKeyboard"

EndSection

A.4 XF86Config-4 hardware graphics compositor configuration file

Below is the listing for an *XF86Config-4* configuration file that is suitable for using the hardware graphics [compositor](#). This configuration file is set to run at 1280x1024@60 Hz. Only the first channel for each pipe is enabled and they are configured to emit a digital signal. Channel cloning and extension are disabled. Support for 3D Stereo and FSAA are enabled in this configuration file. It is as follows:

```
# $XFree86: xc/programs/Xserver/hw/xfree86/XF86Conf.cpp,v 3.44 2001/12/17
20:52:29 dawes Exp $
#
# Copyright (c) 1994-1998 by The XFree86 Project, Inc.
#
# Permission is hereby granted, free of charge, to any person obtaining a
# copy of this software and associated documentation files (the "Software"),
# to deal in the Software without restriction, including without limitation
# the rights to use, copy, modify, merge, publish, distribute, sublicense,
# and/or sell copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in
# all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY
# KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
# MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO
# EVENT SHALL
# THE XFREE86 PROJECT BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF
# OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
# DEALINGS IN THE
# SOFTWARE.
#
# Except as contained in this notice, the name of the XFree86 Project shall
# not be used in advertising or otherwise to promote the sale, use or other
# dealings in this Software without prior written authorization from the
# XFree86 Project.
#
```

```

# $XConsortium: XF86Conf.cpp /main/22 1996/10/23 11:43:51 kaleb $

# *****
# This is a sample configuration file only, intended to illustrate
# what a config file might look like. Refer to the XF86Config(4/5)
# man page for details about the format of this file. This man page
# is installed as /usr/X11R6/man/man4/XF86Config.4
# *****

# The ordering of sections is not important in version 4.0 and later.

# *****
# Files section. This allows default font and rgb paths to be set
# *****

Section "Files"

# The location of the RGB database. Note, this is the name of the
# file minus the extension (like ".txt" or ".db"). There is normally
# no need to change the default.

    RgbPath        "/usr/lib/X11/rgb"

# Multiple FontPath entries are allowed (which are concatenated together),
# as well as specifying multiple comma-separated entries in one FontPath
# command (or a combination of both methods)

    FontPath       "/usr/lib/X11/fonts/100dpi:unscaled"
    FontPath       "/usr/lib/X11/fonts/75dpi:unscaled"
    FontPath       "/usr/lib/X11/fonts/misc/"
    FontPath       "/usr/lib/X11/fonts/Type1/"
    FontPath       "/usr/lib/X11/fonts/Speedo/"
    FontPath       "/usr/lib/X11/fonts/CID/"
    FontPath       "/usr/lib/X11/fonts/100dpi/"
    FontPath       "/usr/lib/X11/fonts/75dpi/"

# ModulePath can be used to set a search path for the X server modules.
# The default path is shown here.

    ModulePath     "/usr/X11R6/lib/modules"

EndSection

# *****
# Module section -- this is an optional section which is used to specify
# which run-time loadable modules to load when the X server starts up.
# *****

```

Section "Module"

This loads the DBE extension module.

Load "dbe"

This loads the miscellaneous extensions module, and disables
initialisation of the XFree86-DGA extension within that module.

SubSection "extmod"
Option "omit xfree86-dga"
EndSubSection

This loads the the various modules required for X server operation

Load "type1"
Load "freetype"
Load "drm"
Load "fb"
Load "xaa"
Load "glx"
Load "dri"
Load "fglrx_glx"

EndSection

Server flags section. This contains various server-wide Options.

Section "ServerFlags"

Uncomment this to cause a core dump at the spot where a signal is
received. This may leave the console in an unusable state, but may
provide a better stack trace in the core dump to aid in debugging

Option "NoTrapSignals"

Uncomment this to disable the <Ctrl><Alt><BS> server abort sequence
This allows clients to receive this key event.

Option "DontZap"

Uncomment this to disable the <Ctrl><Alt><KP_+>/<KP_-> mode switching
sequences. This allows clients to receive these key events.

```

# Option      "DontZoom"

# Uncomment this to disable tuning with the xvidtune client. With
# it the client can still run and fetch card and monitor attributes,
# but it will not be allowed to change them. If it tries it will
# receive a protocol error.

# Option      "DisableVidModeExtension"

# Uncomment this to enable the use of a non-local xvidtune client.

# Option      "AllowNonLocalXvidtune"

# Uncomment this to disable dynamically modifying the input device
# (mouse and keyboard) settings.

# Option      "DisableModInDev"

# Uncomment this to enable the use of a non-local client to
# change the keyboard or mouse settings (currently only xset).

# Option      "AllowNonLocalModInDev"

# Set the basic blanking screen saver timeout.

    Option      "blank time"      "10"      # 10 minutes

# Set the DPMS timeouts. These are set here because they are global
# rather than screen-specific. These settings alone don't enable DPMS.
# It is enabled per-screen (or per-monitor), and even then only when
# the driver supports it.

    Option      "standby time"    "20"
    Option      "suspend time"    "30"
    Option      "off time"        "60"

# On some platform the server needs to estimate the sizes of PCI
# memory and pio ranges. This is done by assuming that PCI ranges
# don't overlap. Some broken BIOSes tend to set ranges of inactive
# devices wrong. Here one can adjust how aggressive the assumptions
# should be. Default is 0.

# Option      "EstimateSizesAggressively" "0"

EndSection

# *****
# Input devices

```

```

# *****

# *****
# Core keyboard's InputDevice section
# *****

Section "InputDevice"

    Identifier    "Keyboard1"
    Driver        "keyboard"

# For most OSs the protocol can be omitted (it defaults to "Standard").
# When using XQUEUE (only for SVR3 and 1, but not Solaris), comment
# out the above line, and uncomment the following line.

# Option        "Protocol"        "Xqueue"

# Set the keyboard auto repeat parameters. Not all platforms implement
# this.

    Option        "AutoRepeat"    "500 50"

# Specify which keyboard LEDs can be user-controlled (eg, with xset(1)).

# Option        "Xleds" "1 2 3"

# To disable the XKEYBOARD extension, uncomment XkbDisable.

# Option        "XkbDisable"

# To customise the XKB settings to suit your keyboard, modify the
# lines below (which are the defaults). For example, for a European
# keyboard, you will probably want to use one of:
#
# Option        "XkbModel"        "pc102"
# Option        "XkbModel"        "pc105"
#
# If you have a Microsoft Natural keyboard, you can use:
#
# Option        "XkbModel"        "microsoft"
#
# If you have a US "windows" keyboard you will want:
#
# Option        "XkbModel"        "pc104"
#
# Then to change the language, change the Layout setting.
# For example, a german layout can be obtained with:
#

```

```

# Option      "XkbLayout"    "de"
#
# or:
#
# Option      "XkbLayout"    "de"
# Option      "XkbVariant"   "nodeadkeys"
#
# If you'd like to switch the positions of your capslock and
# control keys, use:
#
# Option      "XkbOptions"   "ctrl:swapcaps"

# These are the default XKB settings for XFree86
#
# Option      "XkbRules"     "xfree86"
# Option      "XkbModel"     "pc101"
# Option      "XkbLayout"    "us"
# Option      "XkbVariant"   ""
# Option      "XkbOptions"   ""

EndSection

# *****
# Core Pointer's InputDevice section
# *****

Section "InputDevice"

# Identifier and driver

    Identifier  "Mouse1"
    Driver      "mouse"

# The mouse protocol and device.  The device is normally set to /dev/mouse,
# which is usually a symbolic link to the real device.

    Option      "Protocol"     "PS/2"
    Option      "Device"       "/dev/mouse"

# On platforms where PnP mouse detection is supported the following
# protocol setting can be used when using a newer PnP mouse:

# Option      "Protocol"     "Auto"

# When using mouse connected to a PS/2 port (aka "MousePort), set the
# the protocol as follows.  On some platforms some other settings may

```

```

# be available.

# Option "Protocol" "PS/2"

# When using XQUEUE (only for SVR3 and 1, but not Solaris), use
# the following instead of any of the lines above. The Device line
# is not required in this case.

# Option "Protocol" "Xqueue"

# Baudrate and SampleRate are only for some older Logitech mice. In
# almost every case these lines should be omitted.

# Option "BaudRate" "9600"
# Option "SampleRate" "150"

# Emulate3Buttons is an option for 2-button mice
# Emulate3Timeout is the timeout in milliseconds (default is 50ms)

# Option "Emulate3Buttons"
# Option "Emulate3Timeout" "50"

# ChordMiddle is an option for some 3-button Logitech mice, or any
# 3-button mouse where the middle button generates left+right button
# events.

# Option "ChordMiddle"

EndSection

# Some examples of extended input devices

# Section "InputDevice"
# Identifier "spaceball"
# Driver "magellan"
# Option "Device" "/dev/cua0"
# EndSection
#
# Section "InputDevice"
# Identifier "spaceball2"
# Driver "spaceorb"
# Option "Device" "/dev/cua0"
# EndSection
#
# Section "InputDevice"
# Identifier "touchscreen0"
# Driver "microtouch"
# Option "Device" "/dev/ttyS0"

```

```

# Option      "MinX"          "1412"
# Option      "MaxX"          "15184"
# Option      "MinY"          "15372"
# Option      "MaxY"          "1230"
# Option      "ScreenNumber"  "0"
# Option      "ReportingMode"  "Scaled"
# Option      "ButtonNumber"   "1"
# Option      "SendCoreEvents"
# EndSection
#
# Section "InputDevice"
# Identifier   "touchscreen1"
# Driver       "elo2300"
# Option      "Device"         "/dev/ttyS0"
# Option      "MinX"          "231"
# Option      "MaxX"          "3868"
# Option      "MinY"          "3858"
# Option      "MaxY"          "272"
# Option      "ScreenNumber"  "0"
# Option      "ReportingMode"  "Scaled"
# Option      "ButtonThreshold" "17"
# Option      "ButtonNumber"   "1"
# Option      "SendCoreEvents"
# EndSection

# *****
# Monitor section
# *****

# Any number of monitor sections may be present

Section "Monitor"

# The identifier line must be present.

    Identifier   "Generic Monitor"

# HorizSync is in kHz unless units are specified.
# HorizSync may be a comma separated list of discrete values, or a
# comma separated list of ranges of values.
# NOTE: THE VALUES HERE ARE EXAMPLES ONLY. REFER TO YOUR
MONITOR'S
# USER MANUAL FOR THE CORRECT NUMBERS.

# HorizSync 31.5 # typical for a single frequency fixed-sync monitor
# HorizSync 30-64 # multisync
# HorizSync 31.5, 35.2 # multiple fixed sync frequencies
# HorizSync 15-25, 30-50 # multiple ranges of sync frequencies

```

HorizSync 30-130 # multisync

VertRefresh is in Hz unless units are specified.
VertRefresh may be a comma separated list of discrete values, or a
comma separated list of ranges of values.
NOTE: THE VALUES HERE ARE EXAMPLES ONLY. REFER TO YOUR
MONITOR'S
USER MANUAL FOR THE CORRECT NUMBERS.

VertRefresh 60 # typical for a single frequency fixed-sync monitor

VertRefresh 50-100 # multisync
VertRefresh 60, 65 # multiple fixed sync frequencies
VertRefresh 40-50, 80-100 # multiple ranges of sync frequencies

VertRefresh 50-160 # multisync

Modes can be specified in two formats. A compact one-line format, or
a multi-line format.

A generic VGA 640x480 mode (hsync = 31.5kHz, refresh = 60Hz)
These two are equivalent

ModeLine "640x480" 25.175 640 664 760 800 480 491 493 525

Mode "640x480"
DotClock 25.175
HTimings 640 664 760 800
VTimings 480 491 493 525
EndMode

These two are equivalent

ModeLine "1024x768i" 45 1024 1048 1208 1264 768 776 784 817 Interlace

Mode "1024x768i"
DotClock 45
HTimings 1024 1048 1208 1264
VTimings 768 776 784 817
Flags "Interlace"
EndMode

ModeLine "1280x1024_96" 164.966 1280 1300 1460 1600 1024 1026 1029 1074
ModeLine "1280x1024_120" 233.793 1280 1384 1528 1776 1024 1025 1028 1097
Modeline "1920x1200_60p" 157.39 1920 1950 2072 2112 1200 1203 1206 1242
Modeline "1400x1050_100" 214.39 1400 1512 1664 1928 1050 1051 1054 1112

```

# framelock timings, identified by the trailing 'f'
# (==> denotes available source video framelock formats)
# (example: setmon -L ntsc 1280x1024_5994f )

Modeline "1280x1024_50f" 104 1280 1336 1472 1664 1024 1027 1030 1250 -hsync -
vsync
# ==> 625i_50 (alias: pal), 1080p_25, 1080i_50

Modeline "1280x1024_5994f" 115.2 1280 1296 1440 1560 1024 1027 1030 1232 -hsync
-vsync
# ==> 525i_5994 (alias: ntsc), 720p_5994, 1080p_2997, 1080i_5994

Modeline "1280x1024_60f" 121.5 1280 1396 1540 1800 1024 1027 1030 1125 -hsync
-vsync
# ==> 720p_60, 1080p_30, 1080i_60

Modeline "1920x1154_48f" 144 1920 1968 2160 2400 1154 1157 1160 1250 -hsync -
vsync
# ==> 1080p_24, 1080i_48, this is a cvt modified format

Modeline "1920x1154_50f" 144 1920 1968 2160 2400 1154 1157 1160 1200 -hsync -
vsync
# ==> 625i_50 (alias: pal), 1080p_25, 1080i_50, this is a cvt modified format

Modeline "1920x1154_5994f" 192 1920 1992 2192 2464 1154 1157 1160 1300 -hsync -
vsync
# ==> 525i_5994 (alias: ntsc), 720p_5994, 1080p_2997, 1080i_5994, this is a cvt
modified format

Modeline "1920x1154_60f" 172.8 1920 1968 2160 2400 1154 1157 1160 1200 -hsync
-vsync
# ==> 720p_60, 1080p_30, 1080i_60, this is a cvt modified format

Modeline "1920x1200_60f" 180 1920 1968 2160 2400 1200 1203 1206 1250 -hsync -
vsync
# ==> 720p_60, 1080p_30, 1080i_60, this is a cvt modified format

Modeline "1920x1200_50f" 150 1920 1968 2160 2400 1200 1203 1206 1250 -hsync -
vsync
# ==> 1080p_25, 1080i_50, this is a cvt modified format

# Framelock timings with reduced blanking for formats frequency higher than dvi10 max
freq(165Mhz)

Modeline "1920x1154_5994f_p" 156 1920 1968 2000 2200 1154 1157 1160 1283 -
hsync -vsync
# ==> 525i_5994 (alias: ntsc), 720p_5994, 1080p_2997, 1080i_5994

```

```
Modeline "1920x1154_60f_p" 162 1920 1968 2000 2160 1154 1157 1160 1250 -hsync -vsync
```

```
# ==> 720p_60, 1080p_30, 1080i_60, this is a cvt modified format
```

```
Modeline "1920x1200_60f_p" 162 1920 1968 2000 2160 1200 1203 1206 1250 -hsync -vsync
```

```
# ==> 720p_60, 1080p_30, 1080i_60, this is a cvt modified format
```

```
#
```

```
# The following are standard VESA formats modified to be usable for a genlock source  
# through the "C SYNC" cable. The "C SYNC" cable combines sync's into a composite  
# sync,
```

```
# but requires the sync levels to be negative. These ModeLines are distinguished by
```

```
# an "n" (for "negative" sync levels) appended to the ModeLine Name.
```

```
#
```

```
ModeLine "1280x1024_60" 108 1280 1328 1440 1688 1024 1025 1028 1066  
+hsync +vsync
```

```
ModeLine "1280x1024_75n" 135 1280 1296 1440 1688 1024 1025 1028 1066 -  
hsync -vsync
```

```
ModeLine "1280x1024_85n" 157 1280 1344 1504 1728 1024 1025 1028 1072 -  
hsync -vsync
```

```
ModeLine "1600x1200_60n" 162 1600 1664 1856 2160 1200 1201 1204 1250 -  
hsync -vsync
```

```
ModeLine "1600x1200_75n" 202 1600 1664 1856 2160 1200 1201 1204 1250 -  
hsync -vsync
```

```
ModeLine "1600x1200_85n" 229 1600 1664 1856 2160 1200 1201 1204 1250 -  
hsync -vsync
```

```
ModeLine "1920x1080_72n" 211 1920 2056 2264 2608 1080 1081 1084 1126 -  
hsync -vsync
```

```
ModeLine "1920x1200_60n" 193 1920 2048 2256 2592 1200 1201 1203 1242 -  
hsync -vsync
```

```
ModeLine "1920x1200_75" 246 1920 2064 2272 2624 1200 1201 1203 1253  
+hsync +vsync
```

```
Modeline "1400x1050_100" 214 1400 1512 1664 1928 1050 1051 1054 1112 -  
hsync +vsync
```

```
ModeLine "1920x1440_75" 297 1920 2064 2288 2640 1440 1441 1444 1500 -  
hsync +vsync
```

```
ModeLine "1920x1200_75" 246 1920 2064 2272 2624 1200 1201 1203 1253  
+hsync +vsync
```

```
# If a monitor has DPMS support, that can be indicated here. This will
```

```
# enable DPMS when the monitor is used with drivers that support it.
```

```
# Option "dpms"
```

```

# If a monitor requires that the sync signals be superimposed on the
# green signal, the following option will enable this when used with
# drivers that support it. Only a relatively small range of hardware
# (and drivers) actually support this.

# Option      "sync on green"

EndSection

# *****
# Graphics device section
# *****

# Any number of graphics device sections may be present

#Section "Device"

# The Identifier must be present.

# Identifier   "Generic VGA"

# The Driver line must be present. When using run-time loadable driver
# modules, this line instructs the server to load the specified driver
# module. Even when not using loadable driver modules, this line
# indicates which driver should interpret the information in this section.

# Driver      "vga"

# The chipset line is optional in most cases. It can be used to override
# the driver's chipset detection, and should not normally be specified.

# Chipset     "generic"

# Various other lines can be specified to override the driver's automatic
# detection code. In most cases they are not needed.

# VideoRam   256
# Clocks     25.2 28.3

# The BusID line is used to specify which of possibly multiple devices
# this section is intended for. When this line isn't present, a device
# section can only match up with the primary video device. For PCI
# devices a line like the following could be used. This line should not
# normally be included unless there is more than one video device
# intalled.

# BusID      "PCI:0:10:0"

```

Various option lines can be added here as required. Some options
are more appropriate in Screen sections, Display subsections or even
Monitor sections.

Option "hw cursor" "off"

#EndSection

Section "DRI"

Mode 0666

EndSection

Section "Device"

Identifier "SGI SG-0"

Driver "fglrx"

BusId "PCI:2:0:0"

=== Screen Management ===

Option "DesktopSetup" "0x00000000"

Option "MonitorLayout" "TMDS, NONE"

Option "IgnoreEDID" "off"

Option "HSync2" "unspecified"

Option "VRefresh2" "unspecified"

Option "ScreenOverlap" "0"

=== ImageSync options: none, default, (brickid:bus:slot) [ex: "(1c10:0:2)"]

Option "ImageSync" "default"

=== OpenGL specific profiles/settings ===

Option "Capabilities" "0x00000000"

=== OpenGL Overlay ===

Option "OpenGLOverlay" "off"

=== Pseudo Color Visuals (8-bit visuals) ===

Option "PseudoColorVisuals" "off"

=== QBS Management ===

Option "Stereo" "1"

Option "StereoSyncEnable" "1"

=== FSAAM Management ===

Option "FSAAScale" "1"

Option "FSAADisableGamma" "no"

Option "FSAACustomizeMSPos" "yes"

Option "FSAAMSPosX0" "0.250000"

Option "FSAAMSPosY0" "0.416666"

Option "FSAAMSPosX1" "0.083333"

Option "FSAAMSPosY1" "0.083333"

Option "FSAAMSPosX2" "0.416666"

Option "FSAAMSPosY2" "0.750000"

Option "FSAAMSPosX3" "0.750000"

Option "FSAAMSPosY3" "0.916666"

Option "FSAAMSPosX4" "0.583333"

Option "FSAAMSPosY4" "0.250000"

```

Option "FSAAMSPosX5"      "0.916666"
Option "FSAAMSPosY5"      "0.583333"
EndSection

```

Section "Device"

```

Identifier    "SGI SG-1"
Driver        "fglrx"
BusId         "PCI:2:1:0"
# === Screen Management ===
Option "DesktopSetup"      "0x00000000"
Option "MonitorLayout"     "TMDS, NONE"
Option "IgnoreEDID"        "off"
Option "HSync2"             "unspecified"
Option "VRefresh2"         "unspecified"
Option "ScreenOverlap"     "0"
# === ImageSync options: none, default, (brickid:bus:slot) [ex: "(1c10:0:2)"]
Option "ImageSync"         "default"
# === OpenGL specific profiles/settings ===
Option "Capabilities"      "0x00000000"
# === OpenGL Overlay ===
Option "OpenGLOverlay"     "off"
# === Pseudo Color Visuals (8-bit visuals) ===
Option "PseudoColorVisuals" "off"
# === QBS Management ===
Option "Stereo"             "1"
Option "StereoSyncEnable"  "1"
# === FSAAs Management ===
Option "FSAAScale"         "1"
Option "FSAADisableGamma"  "no"
Option "FSAACustomizeMSPos" "yes"
Option "FSAAMSPosX0"       "0.250000"
Option "FSAAMSPosY0"       "0.416666"
Option "FSAAMSPosX1"       "0.083333"
Option "FSAAMSPosY1"       "0.083333"
Option "FSAAMSPosX2"       "0.416666"
Option "FSAAMSPosY2"       "0.750000"
Option "FSAAMSPosX3"       "0.750000"
Option "FSAAMSPosY3"       "0.916666"
Option "FSAAMSPosX4"       "0.583333"
Option "FSAAMSPosY4"       "0.250000"
Option "FSAAMSPosX5"       "0.916666"
Option "FSAAMSPosY5"       "0.583333"
EndSection

```

Section "Device"

```

Identifier    "SGI SG-2"

```

```

Driver      "fglrx"
BusId      "PCI:2:2:0"
# === Screen Management ===
Option "DesktopSetup"      "0x00000000"
Option "MonitorLayout"    "TMDS, NONE"
Option "IgnoreEDID"       "off"
Option "HSync2"           "unspecified"
Option "VRefresh2"        "unspecified"
Option "ScreenOverlap"    "0"
# === ImageSync options: none, default, (brickid:bus:slot) [ex: "(1c10:0:2)"]
Option "ImageSync"        "default"
# === OpenGL specific profiles/settings ===
Option "Capabilities"     "0x00000000"
# === OpenGL Overlay ===
Option "OpenGLOverlay"    "off"
# === Pseudo Color Visuals (8-bit visuals) ===
Option "PseudoColorVisuals" "off"
# === QBS Management ===
Option "Stereo"           "1"
Option "StereoSyncEnable" "1"
# === FSAA Management ===
Option "FSAAScale"        "1"
Option "FSAADisableGamma" "no"
Option "FSAACustomizeMSPos" "yes"
Option "FSAAMSPosX0"     "0.250000"
Option "FSAAMSPosY0"     "0.416666"
Option "FSAAMSPosX1"     "0.083333"
Option "FSAAMSPosY1"     "0.083333"
Option "FSAAMSPosX2"     "0.416666"
Option "FSAAMSPosY2"     "0.750000"
Option "FSAAMSPosX3"     "0.750000"
Option "FSAAMSPosY3"     "0.916666"
Option "FSAAMSPosX4"     "0.583333"
Option "FSAAMSPosY4"     "0.250000"
Option "FSAAMSPosX5"     "0.916666"
Option "FSAAMSPosY5"     "0.583333"
EndSection

```

```

Section "Device"
Identifier "SGI SG-3"
Driver    "fglrx"
BusId    "PCI:2:3:0"
# === Screen Management ===
Option "DesktopSetup"      "0x00000000"
Option "MonitorLayout"    "TMDS, NONE"
Option "IgnoreEDID"       "off"
Option "HSync2"           "unspecified"

```

```

Option "VRefresh2"          "unspecified"
Option "ScreenOverlap"      "0"
# === ImageSync options: none, default, (brickid:bus:slot) [ex: "(1c10:0:2)"]
Option "ImageSync"         "default"
# === OpenGL specific profiles/settings ===
Option "Capabilities"      "0x00000000"
# === OpenGL Overlay ===
Option "OpenGLOverlay"     "off"
# === Pseudo Color Visuals (8-bit visuals) ===
Option "PseudoColorVisuals" "off"
# === QBS Management ===
Option "Stereo"            "1"
Option "StereoSyncEnable"  "1"
# === FSAA Management ===
Option "FSAAScale"         "1"
Option "FSAADisableGamma"  "no"
Option "FSAACustomizeMSPos" "yes"
Option "FSAAMSPosX0"       "0.250000"
Option "FSAAMSPosY0"       "0.416666"
Option "FSAAMSPosX1"       "0.083333"
Option "FSAAMSPosY1"       "0.083333"
Option "FSAAMSPosX2"       "0.416666"
Option "FSAAMSPosY2"       "0.750000"
Option "FSAAMSPosX3"       "0.750000"
Option "FSAAMSPosY3"       "0.916666"
Option "FSAAMSPosX4"       "0.583333"
Option "FSAAMSPosY4"       "0.250000"
Option "FSAAMSPosX5"       "0.916666"
Option "FSAAMSPosY5"       "0.583333"
EndSection

# *****
# Screen sections.
# *****

# Any number of screen sections may be present. Each describes
# the configuration of a single screen. A single specific screen section
# may be specified from the X server command line with the "-screen"
# option.

#Section "Screen"

# The Identifier, Device and Monitor lines must be present

# Identifier "Screen 1"
# Device     "Generic VGA"

```

```

# Monitor    "Generic Monitor"
#
# The favoured Depth and/or Bpp may be specified here

# DefaultDepth 8
#
# SubSection "Display"
#   Depth      8
#   Modes      "640x480"
#   ViewPort 0 0
#   Virtual    800 600
# EndSubsection
#
# SubSection "Display"
#   Depth      4
#   Modes      "640x480"
# EndSubSection
#
# SubSection "Display"
#   Depth      1
#   Modes      "640x480"
# EndSubSection
#
#EndSection

Section "Screen"
    Identifier      "Screen SG-0"
    Device          "SGI SG-0"
    Monitor         "Generic Monitor"
    DefaultDepth 24
    SubSection      "Display"
        Depth      24
        Modes      "1280x1024_60"
    EndSubSection
EndSection

Section "Screen"
    Identifier      "Screen SG-1"
    Device          "SGI SG-1"
    Monitor         "Generic Monitor"
    DefaultDepth 24
    SubSection      "Display"
        Depth      24
        Modes      "1280x1024_60"
    EndSubSection
EndSection

Section "Screen"

```

```

Identifier      "Screen SG-2"
Device          "SGI SG-2"
Monitor         "Generic Monitor"
DefaultDepth 24
SubSection     "Display"
    Depth      24
    Modes      "1280x1024_60"
EndSubSection
EndSection

```

```

Section "Screen"
Identifier      "Screen SG-3"
Device          "SGI SG-3"
Monitor         "Generic Monitor"
DefaultDepth 24
SubSection     "Display"
    Depth      24
    Modes      "1280x1024_60"
EndSubSection
EndSection

```

```

# *****
# ServerLayout sections.
# *****

```

```

# Any number of ServerLayout sections may be present. Each describes
# the way multiple screens are organised. A specific ServerLayout
# section may be specified from the X server command line with the
# "-layout" option. In the absence of this, the first section is used.
# When now ServerLayout section is present, the first Screen section
# is used alone.

```

```

Section "ServerLayout"

```

```

# The Identifier line must be present

```

```

Identifier      "Main Layout"

```

```

# Each Screen line specifies a Screen section name, and optionally
# the relative position of other screens. The four names after
# primary screen name are the screens to the top, bottom, left and right
# of the primary screen. In this example, screen 2 is located to the
# right of screen 1.

```

```

Screen "Screen SG-0"      ""      ""      ""      "Screen SG-1"
Screen "Screen SG-1"      ""      ""      "Screen SG-0" "Screen SG-2"
Screen "Screen SG-2"      ""      ""      "Screen SG-1" "Screen SG-3"
Screen "Screen SG-3"      ""      ""      "Screen SG-2" ""

```

Each InputDevice line specifies an InputDevice section name and
optionally some options to specify the way the device is to be
used. Those options include "CorePointer", "CoreKeyboard" and
"SendCoreEvents". In this example, "Mouse1" is the core pointer,
and "Mouse2" is an extended input device that also generates core
pointer events (i.e., both mice will move the standard pointer).

```
InputDevice "Mouse1" "CorePointer"  
InputDevice "Keyboard1" "CoreKeyboard"
```

EndSection

List of symbols/abbreviations/acronyms/initialisms

3D	3 Dimensional
API	Application Programming Interface
CAVE	Cave Automatic Virtual Environment
COTS	Commercial Off The Shelf
CRT	Cathode Ray Tube
DMX	Distributed Multihead X
DRDC	Defence Research Development Canada
DRI	Direct Rendering Infrastructure
DVI-I	Digital Video Interface - Integrated
DWM	Desktop Window Manager
FSAA	Full Scene Anti-Aliasing
GPU	Graphics Processor Unit
I/O	Input/Output
LCD	Liquid Crystal Display
LIDS	Laboratoire d'intégration des systèmes
OpenGL	Open Graphics Library
PC	Personal Computer
RAM	Random Access Memory
RGB	Red Green Blue
SDK	Software Development Kit/Toolkit
SGI	Silicon Graphics Inc.
TTY	Teletypewriter
VGA	Video Graphics Array
VR	Virtual Reality

Glossary

3D Stereo

3D itself refers to images, models, simulations, or the displaying thereof that use the perception of volume (height x length x width) and shading and texturing to present to the viewer a sense of realism and depth. 3D Stereo (a.k.a. stereoscopy) is the technique whereby the illusion of depth is created by the display of synchronous left-right eye images that are flashed one after the other. First, a left-eyed image is flashed to the screen(s) and then a right-eyed image, and then it repeats over again. Using 3D Stereo goggles the images can be superimposed to provide a more realistic impression of the images displayed from the monitors or screens.

CAVE

This is the common anachronism applied to immersive reality environments where projectors are used to project rendered images, scenes, model, and simulations from a computer that are projected onto wall-sized displays. It is an environment where realism is highly important to place the user experience as the driving factor for the immersion. It is common to incorporate realistic 3D graphics and stereo imaging (using 3D Stereo goggles) as well as surround sound to create an even more realistic environment.

Cloning

A process through which X Windows recreates the same data that is displayed on a graphics pipe's left channel to its right channel. Both images are exactly the same. Furthermore, both channels must be using the same display resolution and refresh rate. Cloning is enabled on a pipe-by-pipe basis; it is not all-inclusive.

Compositor (a.k.a. hardware graphics compositor)

A compositor is a special hardware device that can take multiple video input signals and combine to provide a single video output. By using OpenGL Performer and OpenGL Multipipe, it is possible for each graphics pipe to perform only a portion of the graphics rendering whereby each pipe will perform a part of the task. Each pipe's portion is then recombined in the compositor with where enhanced performance can be seen because each pipe now has to perform a portion of the work. However, the pipes must all be set to the same resolution and the compositor cannot be used to create larger images than the sum of the portions it receives from its input signals.

DMX [configuration file]

This text-based configuration file is used to create virtual desktops. This file defines the logical layout of the variously associated pipes with respect to their position using a Cartesian coordinate system where the position of the pipes are reliant on the size and position of the first pipe defined in the configuration file.

Extension

A process through which the X Windows system creates a desktop that spans both channels of a given pipe. This desktop enables the displaying of images and data across both channels in a seamless fashion. It is enabled on a pipe-by-pipe basis. Furthermore, both channels must be displaying at the same resolution and refresh rate.

FSAA

Full Screen Anti-Aliasing is a hardware method implemented by the video card and its driver that helps to reduce or eliminate aliasing and jagged edges in computer graphics by over-rendering each frame so that the frames are sampled and an average is computed to plot the necessary graphic pixels to be rendered for final display to the screen or monitor. FSAA is very compute-intensive and is not suggested for real-time visualization such as with OpenGL Performer.

IRIX

IRIX is SGI's flagship proprietary UNIX operating system. It is a very high-performance system designed for massive parallel computation and supports very large system configurations with thousands of processors and terabytes of memory. Furthermore, OpenGL was designed under IRIX and it is an integral part of the operating system and window management system.

Linux

Linux is a free and open source software operating system based on UNIX. It was originally conceived and developed (the kernel) by Linus Torvalds, who holds and maintains the copyright to Linux. With the assistance of thousands of developers around the world, it has turned into a robust, stable, and secure operating system, not unlike its other UNIX-based counterparts. Other than the kernel, it is mainly composed of developer-contributed software, much from the FSF (Free Software Foundation).

Multipipe

In short, multipipe is the ability for an application to use multiple graphics pipes. Some applications are specifically programmed using various API's while others are completely naïve of how directly display to multiple pipes. Multipipe-based capability enables an application to run on as many pipes as defined for a given work environment or by that program's code or configuration file. Multipipe application development can be programmed directly into an application using API's from libraries such as OpenGL Multipipe and OpenGL Performer. Multipipe-aware applications are suitable for realistic and immersive simulations where massive quantities of visual data are to be rendered across multiple displays on multiple pipes, maximizing the available visual workspace. Multipipe naïve applications can be made multipipe-enabled by using an X-based proxy service provided by OpenGL Multipipe.

Multipipe-aware

These types of applications actually understand and can directly manage multipipe-based graphics systems and display and render without the use of any additional X-based proxy

services. Multipipe-aware applications can be based on API's from either OpenGL Multipipe or OpenGL Performer.

Multipipe-enabled

These types of applications are actually single-pipe programs that through an OpenGL Multipipe X-based proxy service are able to display and render to multiple pipes. The X-based proxy service intercepts and retranslates OpenGL-based calls into a multipipe-based call.

OpenGL

Developed by SGI, it is a high performance C-based graphics library and API that is multi-platform and can be used to develop 3D applications, games, and other visualization-intensive applications. It is standard almost all UNIX platforms and directly competes with Microsoft's DirectX. OpenGL is a technology-neutral library; the type of hardware being used for visualization does not matter, as it does not directly deal with these very low-level issues.

OpenGL Multipipe

Multipipe is a toolkit and SDK that enables non-multipipe aware applications to function in a multipipe environment and benefit from all that can be done through the use of multipipe-based visualization. OpenGL Multipipe provides a tool for running non-multipipe aware application endowing them with ability to scale across as many graphics pipes and channels as available in a given environment. The SDK can be used to develop multipipe aware application using higher-level API's. OpenGL Multipipe is fully OpenGL and OpenGL Performer compatible. Furthermore, OpenGL Multipipe is required to provide the necessary splitting and recombining of images for a hardware graphics compositor. However, OpenGL Performer drives and manages image rendering across the multiple inputs video signals used in the hardware graphics compositor.

OpenGL Performer

Formerly known as IRIS Performer, it is built on top of SGI's OpenGL's graphics library and API, and is use for the programming and development of real-hard time applications and other visualization applications. As with OpenGL, it is multi-platform and does not have any direct competitors. Its main use is in scenegraph visualization. Performer offers direct support for hardware compositing. OpenGL Performer is also used to drive and manage the image rendering across the multiple pipes of a hardware graphics compositor. OpenGL Performer is fully OpenGL and OpenGL Multipipe compatible.

Refresh rate

The refresh rate is the vertical scan rate; this is the number of times per second that the display is illuminated by the projection system. More succinctly, it is the number of times per second the image is redrawn to the display.

Scenegraph

A scenegraph is 3D model database using vector-based graphic data structures. There are many different types of scenegraphs; however, they are all the same. Furthermore, their data structures are object-oriented. For this report, the only scenegraphs used are based on the Multigen Paradigm OpenFlight scenegraph format. This scenegraph is the de facto standard for high-end data visualization of 3D models and interactive simulations.

UNIX

A multi-user, multi-tasking, multi-threaded operating system based on a kernel that provides a consistent interface to the user for both interactive and background job processing. UNIX is multi-platform and hardware independent, and supports advanced API's. It is generally considered by the computing industry as the hallmark of scalable, robust, secure, and reliable computing. It was originally developed at AT&T Labs by Dennis Ritchie and Ken Thompson.

Virtual desktop

An extended virtual desktop environment created using OpenGL Multipipe and DMX text-based configurations. Using a command line application the DMX configuration file is loaded and according to the positions it specifies for the system's defined displays creates a desktop that spans those specific pipes.

X Windows

X Windows is a popular graphical windowing system commonly available on UNIX-based platforms. There are also several Windows-based ports. It features a server and a client. The X Windows server portion serves the graphical display to either the local system (connected to the system by the graphics console). The X Windows client accepts X-based connections that can occur either locally or from remote connections. X Windows is very plain by itself and requires a windowing manager to provide advanced graphical features and interfaces. X Windows is a TCP/IP-based protocol.

XF86Config-4

This file is the basic configuration file that controls the behaviour and appearance of the X Windows client that the user sees displayed to the monitor(s). This was the standard configuration file used by all the XFree86-based X Window platforms until around version 4.4.0 when the project was disbanded and continued under the X.org project. It is also the standard X Windows configuration file that is used for the SGI Onyx4 platform, which also uses the XFree86 X Windows system.

XFree86

XFree86 was the de facto implementation and started for the X Windows implementation in x86-based UNIX-based operating systems up until version 4.4.0. At that time, X.org took over the responsibility for x86-based X Windows development.

Distribution list

Document No.: DRDC Valcartier TN 2006-730

LIST PART 1: Internal Distribution by Centre:

- 3 Document Library
- 1 Richard Carbone (author)
- 1 Clement L'Heureux
- 1 Francois Bernier
- 1 Francois Rioux
- 1 Murielle Mokhtari
- 1 Roger Fortin
- 1 Claire Lalancette
- 1 Guy Turcotte
- 1 Yves van Chestein
- 1 Luc Chouinard

13 TOTAL LIST PART 1

LIST PART 2: External Distribution by DRDKIM

- 1 Directorate R & D – Knowledge and Information Management
-
- 1 TOTAL LIST PART 2

14 TOTAL COPIES REQUIRED

This page intentionally left blank.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.) DRDC Valcartier		2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.) Unclassified	
3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C, R or U) in parentheses after the title.) (U) Specialty configurations for the SGI Onyx4: Configuring graphics and applications for VR			
4. AUTHORS (last name, followed by initials – ranks, titles, etc. not to be used) Carbone, R.			
5. DATE OF PUBLICATION (Month and year of publication of document.) January 2007	6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.) 126	6b. NO. OF REFS (Total cited in document.) 12	
7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.) Technical Note			
8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.) LIDS			
9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)		9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)	
10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.) DRDC Valcartier TN 2006-730		10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)	
11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.) (X) Unlimited distribution () Defence departments and defence contractors; further distribution only as approved () Defence departments and Canadian defence contractors; further distribution only as approved () Government departments and agencies; further distribution only as approved () Defence departments; further distribution only as approved () Other (please specify):			
12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.)			

13. **ABSTRACT** (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

(U) During the construction of the new virtual reality (VR) facility at DRDC Valcartier, it was determined early on that this facility would require one key component to drive it, the SGI Onyx4 visualization server. However, this computer system required certain specialized configurations in order for it to function correctly within this environment. Consequently, this required the reconfiguration of certain graphics-based configuration files that controlled the system's high-performance graphics subsystem. The SGI's role in the VR environment is to provide near real-time simulation-based capabilities for various high-end 3D models by leveraging the system's integrated hardware and software subsystems. Furthermore, the main applications and toolkits provided with the system had to be fine-tuned to allow for the high level of performance required. This document, therefore, describes all the necessary system configuration and administration-related information required for configuring the system's hardware and software subsystems that will be required for powering the VR environment.

14. **KEYWORDS, DESCRIPTORS or IDENTIFIERS** (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

SGI, Onyx4, perfly, performer, multipipe, open gl, opengl, X Windows, Xfree86, compositor, VR, virtual reality, cave, display, multi head, multi channel, UltimateVision, DMX, distributed multihead x, x.org, xfree86, LIDS

Defence R&D Canada

Canada's Leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale



WWW.drdc-rddc.gc.ca

