

PX Windows Software Manual

VGP Series PMC Graphics Boards (VFX-M, VCQ-M, VFG-M)

August 10, 1999

Peritek Corporation
5550 Redwood Road
Oakland, CA 94619

(510) 531-6500
FAX (510) 530-8563

email: support@peritek.com
web: <http://www.peritek.com>

Notices

Information contained in this manual is disclosed in confidence and may not be duplicated in full or in part by any person without prior approval of Peritek Corporation. Its sole purpose is to provide detailed documentation to effectively install and operate Peritek equipment. The use of this document for any other purpose is specifically prohibited.

The information in this document is subject to change without notice. The specifications of **PX Windows** and other components described in this manual are subject to change without notice. Peritek Corporation assumes no responsibility for any errors or omissions that may occur in this manual.

Peritek Corporation assumes no responsibility for the use or reliability of software or hardware that is not supplied by Peritek, or which has not been installed in accordance with this manual.

PX Windows and **Peritek** are trademarks of Peritek Corporation. LynxOS, Solaris, SunOS, Tru64 Unix, and VxWorks are registered trademarks of Lynx, Sun, Sun, Compaq, and Wind River Systems, respectively.

Table of Contents

1	Introduction.....	7
1.1	Overview of PX Windows	7
1.2	Platforms supported	9
1.3	Getting help	9
1.4	Using this manual	10
2	Software installation.....	11
2.1	The Xvga server distribution	11
2.2	Xvga server and driver installation (Linux, VCQ-M and VFG-M)	12
	Installing the Xvga server (Linux).....	12
	Using the BT848a on the VFG-M (Linux)	13
2.3	Xvga server and client installation (LynxOS, VFX-M).....	14
	Installing the Xvga server (LynxOS).....	14
	Configuring the driver (LynxOS)	14
2.4	Xvga server and driver installation (VxWorks, VFX-M)	15
2.5	Driver for X Windows on Tru64 Unix.....	16
3	Configuring the X server.....	18
3.1	Xvga configuration file types and location.....	18
3.2	Xvga configuration file grammar	18
3.3	Configuration using Xvga.ini	20
3.4	Monitor description files.....	25
3.5	Board description files.....	25
3.6	Command-line configuration	26
3.7	Kernel configuration (VxWorks).....	26
3.8	Xdec server and ws configuration (Tru64 Unix only)	27
4	Running the Xvga server	28
4.1	Command line start-up (LynxOS and Unix).....	28
4.2	X Display Manager start-up (LynxOS and Unix)	28
4.3	Start-up scripts (VxWorks).....	29
4.4	Security	30
4.5	Using xdpinfo to test the server	31
5	Man pages.....	33
6	Bibliography.....	39
	X Window System References	39
	Peritek VGP-series Hardware and Software Manuals	39
7	Appendix A.....	40
	Syntax of VESA Display Information Format files (.vda files).....	40
8	Index	43

List of Figures

Figure 1–1 X Window System architecture.....	7
Figure 3–1 Top-level headings of Xvga.ini configuration file	20
Figure 3–2 Structure of a monitor description (.vda) file.	25
Figure 3–3 Overall structure of board description (.xqa) files.....	26

List of Tables

Table 1-1 Platforms currently supported by PX Windows	9
Table 2-1 Top level directories of the PX Windows Xvga distribution	11
Table 2-2 Subdirectories of the configuration directory, PeritekX.....	12
Table 3-1 Location of Xvga configuration files	18
Table 3-2 Configuration files in \$XVGPHOME (PeritekX)	18
Table 3-3 Keymap file names.....	21

1 Introduction

This document explains how to install Peritek's PX Windows, the X server software for Peritek's **VGP-series** graphics boards, and how to configure and run the software in different system environments. Graphics boards in the **VGP-series** include the **VFX-M**, **VCQ-M** and **VFG-M** boards. In addition to the X server software described in this manual, all boards in the **VGP-series** are also supported by Peritek's **Graphics Library Package**; contact Peritek or check the *Graphics Library Package Software Manual* on our web page for more information.

This introductory chapter contains an overview of the software that is provided with the X server distribution and information about the platforms that Peritek supports, how to get technical support, the organization of this manual, and the typographical conventions used throughout the manual.

1.1 Overview of PX Windows

PX Windows for Peritek's **VGP-series** graphics boards is an implementation of an X server based on the X11R6 X Window System and specifically optimized for Peritek boards. The VGP (Video Graphics for PMC) boards may be used in VMEbus systems with PMC connectors and in Compact PCI systems. These boards are based on a variety of chip-sets to meet different performance requirements, but characterized by uniform Peritek software support for basic graphics functions. More information about each board's capabilities can be found in their hardware manuals, which are listed in the **Bibliography**.

Unlike the PX Windows X server for the previous generation of TMS34020-based Peritek boards, this X server is executed by the system processor, aided where possible by graphics acceleration functions on the Peritek board. However, the overall performance of the PX Windows X server will be dependent on the processor speed as well as other available resources.

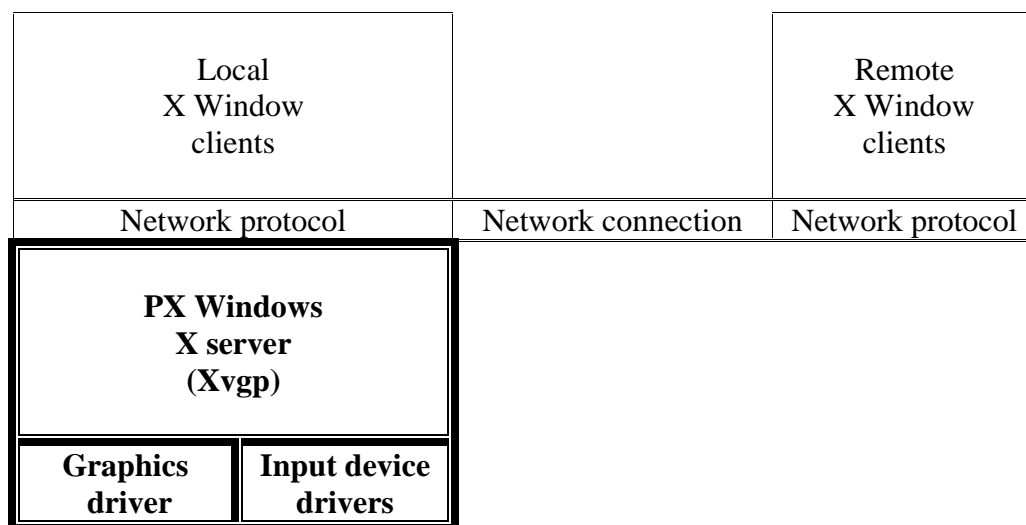


Figure 1-1 X Window System architecture

Peritek provides an X server designed to function as a part of your system's X Window System installation. The X11R6 X Window System software is a public domain, machine independent, network-based windowing system for graphics workstations. It runs on computers ranging from PCs to supercomputers. X supports a large number of bit mapped color and monochrome display controllers and workstations, many text fonts (both fixed and variable width), keyboards, track balls, touchscreens and mice. X was originally developed as a joint project of MIT's Project Athena, IBM, and Digital Equipment Corporation. Many other companies have subsequently made contributions to X.

X uses a client-server architecture. Figure 1-1 illustrates the overall software structure of this architecture—the server and device driver software provided by Peritek is outlined in bold. The server controls hardware-dependent functions, such as a mouse, keyboard, track ball, or graphics display, through operating system-dependent device drivers. Clients include programs the user interacts with, such as a clock, a text window, or a performance indicator, as well as window managers, such as MWM (Motif Window Manager) or TWM, terminal emulators (such as DECterm or xterm), desktop publishing programs, such as Interleaf, or image processing packages, such as PV-Wave. Client packages can be bought from many software vendors, and are often supplied along with the operating system. Customers often write their own clients which they use to customize their own products.

The server and the clients may be on the same computer or they may be connected through a network. The connection between the clients and server is through a network protocol, whether the client is local or remote. The client images (user programs) are usually linked with the standard X Library (XLIB) which manages communications with the server. Other libraries, called tool kits, are available from software suppliers. Tool kits expand the functionality of the X Windows programming environment. To find out more about X Windows, refer to the Bibliography.

Peritek's PX Windows provides an X server, along with device drivers to support the graphics and input device functions on the Peritek board. ***Client libraries and packages must be obtained either with your operating system or from some other software vendor.***

The PX Windows X server supports only Peritek graphics boards; graphics cards by other vendors are not supported. However, the X server can be configured to support a variety of input devices, depending on the availability of drivers for your system. These input devices may either be on the Peritek board, or separate. The VFX-M includes PS/2 mouse and keyboard ports; the handling of interrupts for those ports is tightly coupled with the graphics display, as the VFX-M is a single PCI device, so drivers are supplied as part of the X server distribution. On the VCQ-M and VFG-M, two USB ports are provided on a secondary PCI bus that can also have multiple graphics display controllers. These ports should be detected by your system drivers in the same way as any other PCI device.

Please note that even local clients, running on the same computer as the PX Windows X server, must use the X protocol with its corresponding overhead to communicate with the X server and the Peritek hardware. This is the price paid for transparent execution of X clients on local and remote X servers, and for the platform-independence of X applications. If you are not interested in remote client execution, or in

pre-existing X applications, you may wish to use Peritek's **Graphics Library Package** for your application; contact Peritek for more information.

1.2 Platforms supported

Peritek continually develops new ports of PX Windows. Table 1-1 shows the platforms currently supported and described in this manual. If your platform, configuration or system is not listed, call Peritek. Peritek often creates customized installations.

The support for **Compaq Tru64 Unix** is still in the form of a simple unoptimized graphics driver to interface with the **Xdec** server and **ws** driver on that operating system. For **LynxOS**, **VxWorks** and **Linux** systems, an optimized X server, **Xvgp**, is provided for the VFX-M, VCQ-M and VFG-M. Drivers are provided for the VFX-M's on-board PS/2 mouse and keyboard input. For the **LynxOS Cestia CVME603/604** systems, drivers for the processor board's PS 2 keyboard and mouse are also available. The USB ports on the VFG-M and VCQ-M rely on the operating system's drivers for support.

Current software development includes a port of **Xvgp** to Solaris Sparc and x86 systems, and further development of the **Xvgp** to support all **VGP-series** boards on all the currently supported systems.

Operating System	Processor Board	Peritek Board	Peritek Software
Compaq Tru64 Unix	Alpha boards	VFX-M	Graphics driver only
Linux 2.2	Intel x86	VCQ-M, VFG-M	Server
LynxOS 3.0	Motorola 1600/2600 series	VFX-M	Server, drivers
LynxOS 3.0	Cestia CVME603/604	VFX-M	Server, drivers
Wind River VxWorks 5.3	Motorola 1600/2600 series	VFX-M	Server, drivers
Wind River VxWorks 5.3	Cestia CVME603/604	VFX-M	Server, drivers
Wind River VxWorks 5.3	Radstone RS60x	VFX-M	Server, drivers

Table 1-1 Platforms currently supported by PX Windows

1.3 Getting help

This installation manual gives specific steps to take to install your Peritek graphics board's software. There are, however, variables specific to your computer configuration and monitor that this manual cannot address. Normally, the default values given in this manual will work. If you have trouble installing or configuring your system, contact Peritek technical support at

Peritek Main Number: **(510) 531-6500**
Peritek Fax Number: **(510) 530-8563**
Peritek E-mail: support@peritek.com

1.4 Using this manual

This manual is designed to be read in order: first this chapter's introductory overview, then software installation, followed by configuration information, then by information on how to start running the X server and connect clients. Within each chapter, information that is specific to a particular operating system is organized into a clearly marked section—you may skip reading sections marked for operating systems other than your own. The final **Man Pages** chapter gives a reference to command-line options for the **Xvga** server. The detailed **Table of Contents** should allow you to find the information relevant to your system quickly, and a **Bibliography** contains references both to books and to documents on web-sites that contain more information about the X Window system and about Peritek hardware and software. You will also find certain topics listed in the **Index**, such as “resolution” or “pixel format.”

Different fonts are used as follows:

Bold face type (like this) indicates a special name within running text, such as a command name, file name or company name.

Plain fixed width text (like this) indicates a text in an ASCII configuration file or script, or text as it is typed or displayed on a command line.

Italic type (like this) is used in the usual way, for documents titles and for emphasis.

2 Software installation

The first section of this chapters applies to all systems using **PX Windows'** optimized server for the **VGP series** graphics boards, **Xvgp**, and should be read by everyone, unless you are installing the driver for **Compaq's Tru64 Unix**. The remaining sections in this chapter are operating system-specific. Section 2.5 applies only to the driver for **Compaq's Tru64 Unix**.

2.1 The Xvgp server distribution

The **Xvgp** server software is distributed on CD ROM. The top-level directories on in the **dist** directory on the CD ROM are shown in Table 2-1.

PeritekX	This directory and its subdirectories are needed by the running Xvgp server in order to obtain configuration information.
Xserver	Subdirectories, named by operating system and processor, contain the executable Xvgp file for that system type
clients	Subdirectories containing source for sample clients exercising particular hardware features of Peritek graphics boards.
doc	Contains Acrobat .pdf files for this manual and for hardware manuals for the VGP-series graphics boards
drv	Subdirectories, named by operating system and processor, contain the files needed to install device drivers.
fonts	Has subdirectories for several font families, as a convenience, in case fonts are not already available on your system's X installation.
install	For systems with fixed installation locations, this directory has a script that copies files to the correct location
os_specific	Operating system specific start-up scripts and configuration files.

Table 2-1 Top level directories of the PX Windows Xvgp distribution

Several different kinds of configuration files are accessed by the **Xvgp** server when it starts up, as described in Chapter 3, *Configuring the X server*, page 18. The **PeritekX** directory contains these configuration files. This directory must be copied from the CD ROM to a place on the system known to the **Xvgp** server. On systems with fixed locations, this is done by a script in the **install** directory; on other systems the user must set environment variables to describe the location of these files. The subdirectories of **PeritekX** are shown in Table 2-2.

For each system described in the following sections, the install process has four basic steps:

- 1) Copy the correct executable **Xvgp** file for that system from the CD ROM to the correct place.

- 2) Copy the configuration file directory, **PeritekX** and all its subdirectories to the correct place.
- 3) Copy the driver files to the correct place and install the drivers. On some systems it is necessary to recompile the drivers.
- 4) Copy any system-dependent start-up or configuration files.

You may prefer to place the entire Peritek distribution on your own system and proceed with the install from there, rather than installing directly from the CD ROM. For convenience in doing so, the file **dist.tar** is provided at the top-level of the CD ROM. On systems with the **tar** command available, make a directory named, for example, **peritek_dist**, where you wish to place all the directories in Table 2-1. Then do the following:

```
# cd peritek_dist
# tar xvf /cdrom/dist.tar
```

Then when you follow the installation instructions in the following section, use the name of the **peritek_dist** directory instead of **/cdrom/dist**.

arch	Has two subdirectories, LynxOS and VxWorks, each containing a sample file Xvgp.ini , with driver names appropriate to the system.
bin	Empty, Xvgp for appropriate system gets copied here as part of installation.
boards	The number9 subdirectory contains a board description file for the VFX-M, vfx-m.xqa . The cirrus subdirectory contains board description files for the VFG-M, vfg-m.xqa and the VCQ-M, vcq-m.xqa .
etc	Has the following files: Xcodes , Xkeyboards , Xlogo.im1 , Xlogo.im8 , Xlogo.im8g , Xrgb.txt , Xsiteinfo , Xtimings , xterm.termcap , xterm.terminfo .
fonts	Has one subdirectory, misc , which contains a small set of fonts necessary for the Xvgp server's own use.
keyboards	Contains the keymap .xkm files; these are the same for all supported PowerPC systems.
log	Empty, used for error messages from the server on some systems.
run	Empty, used to record server process ID on some systems.

Table 2-2 Subdirectories of the configuration directory, **PeritekX**

2.2 Xvgp server and driver installation (Linux, VCQ-M and VFG-M)

Installing the Xvgp server (Linux)

For Linux, you must have root privilege to perform the installation. Go to the **install** directory on the CD ROM and run the script **install.sh**, as shown:

```
# cd /mnt/cdrom/dist/install
# sh install.sh
```

The install script does the following:

- 1) Copy **dist/Xserver/linux/Xvgp** file from the CD ROM to **/usr/X11R6/bin**
- 2) Copy the configuration file directory, **dist/PeritekX** and all its subdirectories **/usr/X11R6/lib/X11/PeritekX**.

The install script assumes that that X11 file hierarchy from the Linux distribution is already in place before it is executed. At Peritek we compile and test our server using the Red Hat 6.0 distribution.

After running the install script, before you can run the **Xvgp** server, you will need to:

- 1) Choose the configuration file in **/usr/X11R6/lib/X11/PeritekX/arch/LINUX** that is closest to the configuration you want for your display. Copy this file to **/etc/Xvgp.ini** and change the font paths, mouse set-up, screen depth and monitor type as needed. (see Chapter 3 for a description of **Xvgp.ini** and other configuration files).
- 2) If you wish to run **startx** or **xdm**, change the original X11 set-up so that it knows about the **Xvgp** server. See Sections 4.1 and 4.2.

Using the BT848a on the VFG-M (Linux)

If you wish to use the capabilities for framegrabbing and displaying TV video provided on the VFG-M by the BT848A chips, you will need to install the version of the Linux **bttv** driver that is included in the Peritek distribution as **drv/linux/bttv/bttv.tar**. You must have a Linux kernel with loadable modules enabled and be working as root in order to do this.

```
# cp /mnt/cdrom/dist/drv/linux/bttv/bttv.tar /usr/src/linux/drivers
# tar xvf bttv.tar
# cd bttv
# make ins
```

This will install a version of the **bttv** driver that recognizes the Peritek card. The last command will need to be done whenever the system is rebooted.

To get started on your application, the version of the publicly distributed **xawtv** client that was used at Peritek when testing the hardware is provided in the Peritek distribution as **clients/linux/tvclients.tar**. Choose a directory where you wish to install this client and do the following:

```
# cp /mnt/cdrom/dist/clients/linux/tvclients.tar .
# tar xvf tvclients.tar
# make
# make install
```

A man page for the **xawtv** client will be installed as well as the executable; see also the README files in the **xawtv** source directory. The file **linux/clients/sample.xawtv** should be copied to the users home directory, and edited if necessary to choose the correct defaults. Start watching TV with the command:

```
# xawtv -c /dev/video0
```

and click with the right mouse button on the video window to bring up a pop-menu that will allow you to change settings and turn capture off and on. Currently only the **grabdisplay** method of capture is supported with the VFG-M.

2.3 Xvgp server and client installation (LynxOS, VFX-M)

Installing the Xvgp server (LynxOS)

For LynxOS, you must have root privilege to perform the installation. First mount the CD ROM (the sample commands below assume the CD ROM is SCSI device 6), then go to the **install** directory and run the script **install.sh**, as shown:

```
bash# /sbin/mount /dev/sd6 /cdrom
bash# cd /cdrom/dist/install
bash# sh install.sh
```

After running the install script, before you can run the **Xvgp** server, you will need to:

- 1) Check the configuration file **Xvgp.ini** that has been installed in **/usr/lib/X11/PeritekX/arch/LynxOS** to see that the font paths and monitor type are correct for your system (see Chapter 3).
- 2) Configure the driver (see below).
- 3) If you wish to run **xdm**, change the original X11 set-up so that it knows about the **Xvgp** server (see Section 4.1). Files in **/cdrom/dist/os_specific/lynx** are samples of these changes; you may wish to copy them to a reference directory after the install script has run; the install script does not copy them, in order to avoid destroying your previous X setup.

The install script does the following:

- 3) Copy **/cdrom/dist/Xserver/lynx/Xvgp** file from the CD ROM to **/usr/bin/X11**.
- 4) Copy the configuration file directory, **/cdrom/dist/PeritekX** and all its subdirectories **/usr/lib/X11/PeritekX**.
- 5) Copy the driver file directory **/cdrom/dist/drv/lynx/** to **/sys/drivers/pk_vfx**.

The install script assumes that that X11 file hierarchy from the LynxOS distribution is already in place before it is executed.

Configuring the driver (LynxOS)

On LynxOS, the device driver for the VFX-M board is loaded dynamically, without recompiling the kernel. However, the driver code must be recompiled on your system, because the specifics of the PCI bus and slot number for the Peritek board must be compiled into the **.info** files used when creating devices. Choose the correct makefile for your system (e.g., **Makefile.mv2604**), make a copy of this file to **Makefile** and edit it so that the bus and slot number given to the **vfx_info** command that creates the **.info** files are the correct ones for the location of the VFX-M on your system. See the comments in the **Makefile** for more information.

When examining the **Makefile**, you will see that **nm** is used to process the object file from the compilation of **vfx.c** in order to get the references that need to be placed in the **.import** file for linking. Note that on LynxOS 2.5.1 systems, the **.import** file must be created WITH leading periods where they occur in the output of **nm**, but on LynxOS 3.0 systems it must be created WITHOUT these leading periods—check to see that the **Makefile** you are using does this correctly for your OS version.

Type **make all** in the driver directory to do all the compilation. When you are ready to install the driver and create the devices, type **make load**. This will install the VFX-M driver, install three devices—one for the graphics driver, one for the mouse, and one for the keyboard—and associate them with the device files **/dev/i128**, **/dev/ps2mouse** and **/dev/ps2keyboard**. Whenever the system is rebooted, this **make load** command must be given before starting the **Xvgp** server. See Section 4.1 for information on how to make this part of the system start-up sequence.

2.4 Xvgp server and driver installation (VxWorks, VFX-M)

For VxWorks there is no install script, and no fixed place where files must be installed. The following installation procedure assumes you have a file system available for your VxWorks system, either remote or local. Contact Peritek if you wish to run on a system with no file system available.

Decide where you wish to keep the files associated with the **Xvgp** server—in the instructions below we will call that directory **PeritekX**, and we will refer to files on the CD ROM with Unix-style pathnames starting from **/cdrom**. Then do the following:

- 1) Copy **/cdrom/dist/Xserver/vxworks/Xvgp** file from the CD ROM to **PeritekX**.
- 2) Copy the configuration file directory, **/cdrom/dist/PeritekX** and all its subdirectories to **PeritekX**. If you do not already have X fonts on your system, you may also wish to copy **/cdrom/dist/fonts** and all its subdirectories to a directory on your system.
- 3) Choose the **vfxdriver.o_xxx** file for your system (e.g., **vfxdriver.o_mv2604**) from **/cdrom/dist/drv/vxworks/** and copy it to **PeritekX/vfxdriver.o**. The source files are also in that directory, but you should not need to recompile the driver unless your kernel has a non-standard memory mapping.
- 4) Copy the start-up script **/cdrom/dist/os_specific/vxworks/vfx_startx.scr** to **PeritekX**. If neither the firmware nor the OS on your system programs PCI headers for devices in the PMC slots, you will also need the script for configuring the PCI header, **/cdrom/dist/os_specific/vxworks/vfx_config.scr**. Check to see if there is a subdirectory of **/cdrom/dist/os_specific/vxworks** with the name of

your BSP and, if so, check the **README** in that directory and copy the files to the appropriate places.

See Chapter 3 for information on the configuration files in **PeritekX**; you will need to change the path names in the **PeritekX/arch/VxWorks/Xvgp.ini** file for your system, and you may also need to change the monitor type. See Section 3.7 for information on changes that may be needed to your VxWorks kernel in order to run the **Xvgp** server, and Section 4.3 for a start-up script to create devices and start the **Xvgp** server.

2.5 Driver for X Windows on Tru64 Unix

Peritek's Tru64 UNIX **PERVFX1.0.0** binary installation kit for the VFX-M driver allows the VFX-M to be easily integrated into the Compaq Tru64 UNIX X Windows WorkStation (WS) environment. After the installation procedure, the VFX-M can be used to provide the X Windows login console and to execute standard X applications.

Note that you must have root permission for the installation and kernel rebuild. Be sure to save your previous kernel file, **/vmunix** in case of any failures during the kernel rebuild.

To install the **PERVFX1.0.0** binary kit for the VFX-M PMC/PCI graphics card, first mount the CD-ROM. In these instructions we will refer to the mount point as **/cdrom**. For example, if the CD-ROM device is **/dev/rz4c**,

```
#mount -r /dev/rz4c /cdrom
```

Then use the **setld** command to install the driver (**setld** is usually in **/usr/sbin**, if you find it is not in your root path):

```
#setld -l /cdrom/install
```

A new directory with the name **PERVFX100** should have been created in **/usr/opt** on your system. The last 3 digits of the name are the version number, e.g. 100 is 1.0.0. Check this directory to make sure that it includes the following files:

```
% ls /usr/opt/PERVFX100
total 40
1 Xserver.conf          1 files          35 vfx.mod
2 config.file          1 sysconfigtab
```

In addition the above files, the installation process should have added a "device" entry to **/usr/var/X11/Xserver.conf**, as follows:

```
device <
      < _dec_ws lib_dec_ws.so fbInitProc Vfx >
>
```

Please check this file to make sure this entry has been added, and that there are no other conflicting device entries.

After the driver is installed, you will need to recompile your kernel and reboot the system. The system name “HOSTNAME” is used in the example below, you will need to replace “HOSTNAME” everywhere it appears in the example with the name you are using for your configured kernel.

Note: If there is already a **HOSTNAME.list** file in **/sys/conf** that does not include a **PERVFX100** line, then it must either be added to the existing **HOSTNAME.list** file, or the **HOSTNAME.list** file must be deleted and the **doconfig** command will recreate it. If you add the line by hand then simply copy the line from the **/sys/conf/.product.list** file, which should have been altered to include a reference to the Peritek driver by the install process when **setld** was run.

```
# doconfig -c HOSTNAME
# mv /vmunix /vmunix.sav
# cp /sys/HOSTNAME/vmunix /vmunix
# shutdown -r +5 "rebooting to pick up VFX Device Drivers"
```

When the system has rebooted, you should see the **Vfx** device detected during the probe for workstation devices as part of the boot-up process. If the Tru64 UNIX X Windows environment has been properly configured for login, the console login should appear on the monitor connected to the VFX-M.

3 Configuring the X server

For simple configuration of the **Xvgp** server, there are sample **Xvgp.ini** files in the **Linux**, **LynxOS** and **VxWorks** subdirectories of the **PeritekX/arch** directory from the CD ROM that are put in the right place by the installation procedures. These **Xvgp.ini** files should be usable to get started on your system with only a change in file path names, in mouse setup and in the monitor type. The **Xvgp** configuration files are described in detail in Sections 3.1 to 3.5. Section 3.6 describes command-line configuration, including how to use a VxWorks specific configuration file for the equivalent of command line flags on VxWorks. Only Section 3.8 has information that applies to **Compaq Tru64 Unix**.

3.1 Xvgp configuration file types and location

The Peritek **Xvgp** server uses a few key configuration files, all in ASCII form. **Xvgp.ini** is the top level configuration file, and points to the **.xqa** (extended query architecture) graphics board description files, and to the **.vda** (VESA display information format) monitor description files. Files referenced in **Xvgp.ini** are found by the server relative to **\$XVGPHOME** (the **PeritekX** directory on the CD ROM distribution is the source of these files). For LynxOS and Linux, these paths are fixed; for VxWorks the names are settable by environment variables. See Table 3-1

File or directory	System	Path
Xvgp.ini	LynxOS	/usr/lib/X11/PeritekX/arch/LynxOS/Xvgp.ini
	Linux	/etc/Xvgp.ini
	VxWorks	\$XVGPHOME
\$XVGPHOME	LynxOS	/usr/lib/X11/PeritekX
	Linux	/usr/X11R6/lib/X11/PeritekX
	VxWorks	\$XVGPHOME

Table 3-1 Location of Xvgp configuration files

The configuration files found relative to **\$XVGPHOME** are shown in Table 3-2.

Directory name	Contents
\$XVGPHOME/boards	Graphics board configuration files
\$XVGPHOME/etc	Server support files
\$XVGPHOME/keyboards	Keyboard layouts
\$XVGPHOME/monitors	Monitor configuration files

Table 3-2 Configuration files in \$XVGPHOME (PeritekX)

3.2 Xvgp configuration file grammar

All the Xvga ASCII configuration files (**Xvga.ini**, **.xqa** and **.vda**) use a similar grammar. The core types in the ASCII files are:

<integer>	non-negative numbers
<scaled>	<integer> or <integer>.<integer>
<string>	characters or “characters”
<date>	<integer>/<integer>/<integer> (year/month/day)

<integer> can be given as either a decimal number or as a hex number in C style. An <integer> is always treated internally as 32 bit unsigned number.

The <scaled> type is used for representing a floating point value. Since floating point usage within the files is limited, there is no provision for exponents.

A <string> is composed of 7 bit ASCII characters. Normally strings are case insensitive which mean DATE, Date, date and dATE all represent the same string. However, if you quote a string with double quote marks “”, then no case translation takes place and it is used literally.

All keywords used are *case insensitive*. That means you can use uppercase and lowercase letters to make the file more readable. Individual tokens must be separated by white space(s). White space may be any of the following characters

- a space
- a tab character
- a comma
- an open or close parenthesis ()
- an open or close brace { }

Comments may appear anywhere in the file. Following C++ language conventions, comments begin with a // (two slashes, no space between) and continue to the end of the current line.

The end of a line is denoted by a **LF** or a **CR**. This is compatible with both UNIX and MS-DOS file conventions. There is no explicit End-Of-File **EOF** character used. The **^Z** used by MS-DOS editors as an **EOF** is ignored.

The overall structure of the configuration files is composed from a few syntactically simple elements. A file is structured into a number of *sections*. These sections build up the information hierarchy of a configuration file. A section name is a keyword that is always enclosed in square brackets (e.g. [TOKEN]) on a line by itself.

The information within a section can be structured in several ways (depending upon the semantics of a section):

Direct form:

```
<value> ;
```

Statement form:

```
<keyword> = <value> ;
```

Array form (may span multiple lines):

```
<value>, <value>, <value> ;
```

In case of the **statement form** not all <keywords> need be given. If a statement is missing, a default value will be used. For the special case of vda-files, the VESA standard permits no omissions.

3.3 Configuration using Xvgp.ini

The **Xvgp.ini** file controls the font paths, input and graphic devices, pixel format, visual types and other settings for the **Xvgp** server. The top-level headings for the **Xvgp.ini** file are shown in Figure 3–1.

```
[ FONTPATH ]
[ RGBPATH ]
[ KEYBOARD ]
[ MOUSE ] | [ TABLET ] | [ TOUCHSCREEN ]
[ SETTINGS ]

[ SCREEN ]
  [ RESOLUTIONS ]

[ SCREEN ]
  [ RESOLUTIONS ]
```

Figure 3–1 Top-level headings of Xvgp.ini configuration file

Note that there may be more than one [SCREEN] section per **Xvgp.ini** file. Multiple SCREEN sections tell the server to use more than one graphics board.

For a sample **Xvgp.ini** file, with the correct device names for your system, look in the directory **PeritekX/arch/<your OS name>** on the PX Windows distribution.. You will need to change the search path for the fonts and for the RGB database, and may wish to change other parameters as well. The syntax of each section under the top-level headings is given below, followed by a description of the parameters used in that section.

```
[ FONTPATH ]
  <strings>;
```

Search path for fonts. This path is a comma separated list of directories which the server searches for font databases (**fonts.dir**). An X11R5 or 6 font server may be specified in the font path.

```
[ RGBPATH ]
  <string>;
```

Name of the RGB color database (ASCII file version), usually **\$XVGPHOME/etc/Xrgb**.

```
[ KEYBOARD ]
  Layout           = <string>;
  HotKeyExit       = YES | NO;
  HotKeyVT         = YES | NO;
```

```
HotKeyZoom      = YES | NO;
ServerNumLock   = YES | NO;
```

Controls keyboard layout and key sequences.

Layout specifies the keyboard layout to use at start-up time. Possible choices for the keymap file name are shown in Table 3-3.

HotKeyExit enables or disables the CTRL+ALT+BackSpace sequence for halting the server.

HotKeyZoom enables or disables the CTRL+ALT+KP_+ and CTRL+ALT+KP_- sequences for changing monitor resolution.

ServerNumLock forces the server to handle the NumLock Key internally. This is provided for systems where the Xlib cannot handle the NumLock Key correctly.

Keymap file name	Description
da_DK.xkm	Danish
de_AT.xkm	Austrian
de_CH.xkm	Swiss German
de_DE-sni.xkm	SNI 97801 compatible
de_DE.xkm	German
el_GR.xkm	Greek
en_CA.xkm	Canadian English
en_GB.xkm	UK English
en_US.xkm	US American
en_US-dvk.xkm	US Dvorak
es_ES.xkm	Spanish
es_LA.xkm	Latin American Spanish
fi_FI.xkm	Finnish
fr_BE.xkm	Belgian
fr_CA.xkm	Canadian French
fr_CH.xkm	Swiss French
fr_FR.xkm	French
is_IS.xkm	Icelandic
it_CH.xkm	Swiss Italian
it_IT.xkm	Italian
ja_JP.xkm	Japanese
nl_NL.xkm	Dutch
no_NO.xkm	Norwegian
pt_PT.xkm	Portuguese
sv_SE.xkm	Swedish
tr_TR.xkm	Turkish

Table 3-3 Keymap file names

[MOUSE]

```
Device          = <string>;
Protocol        = MICROSOFT |
                MOUSEMAN |
                MOUSESYSTEMS |
                LOGITECH |
                BUSMOUSE |
                PS/2;
BaudRate        = <integer>;
Emulate3Buttons = YES | NO;
EmulateTimeout  = <integer>;
```

Device describes the operating system device file for use with the mouse.

Protocol selects the type of mouse (i.e. its protocol) that is connected to the system.

BaudRate set the baud rate, either on the mouse or the computer's side. For mice that allow dynamic speed adjustments (like Logitech) the baud rate is changed in the mouse. Otherwise the rate is set on the computer's side to allow mice with non-standard rates.

Emulate3Buttons enables or disables the emulation of a third button for mice which only have two physical buttons. The third button is emulated by pressing both buttons simultaneously.

EmulateTimeout is the timeout in milliseconds for the 3 button emulation. Since a single button press could mean that either really only a single button is pressed, or a second press will follow (which means a virtual middle button event), there is a timeout controlling when to post the ButtonPress event.

```
[TABLET]
Device           = <string>;
Protocol         = MMSERIES |
                 BITPADONE;
BaudRate        = <integer>;
Parity          = DEFAULT |
                 NONE |
                 EVEN |
                 ODD;
StopBits        = DEFAULT | 1 | 2;
LinesPerInch    = <integer>;
ScaleX          = <integer>;
ScaleY          = <integer>;
Transducer      = STYLUS | CURSOR;
Size            = <integer>x<integer>;
```

Device, **Protocol** and **BaudRate** have the same semantics as in the mouse section.

LinesPerInch is the accuracy of the Tablet.

ScaleX and **ScaleY** scale factors to postscale the coordinates coming from the tablet before they are posted as an X-Event.

Transducer is the type of the Transducer.

Size is the width and height of the tablet in tablet coordinates. Tablets have a different coordinate system than X itself has. To transform the tablet coordinates into X-coordinates the X-Server has to know the size of the tablet in terms of reported coordinates. For BITPADONE type tablets this information cannot be queried from the tablet itself and has to be specified explicitly.

```
[TOUCHSCREEN]
Device           = <string>;
Protocol         = DURATOUCH280;
BaudRate        = <integer>;
OriginX         = <integer>;
OriginY         = <integer>;
Size            = <integer>x<integer>;
```

Device, **Protocol** and **BaudRate** have the same semantics as in the mouse section. **OriginX**, **OriginY** and **Size** are parameters for adjusting a touchscreen. The **Size** parameter specifies the total maximum values the touchscreen returns in device dependent units. **OriginX** and **OriginY** specify the minimum values adjusted to the screen origin.

```
[SETTINGS]
  SnfFormat      = ISC | SCO | SUN | HP | IBM | DEC;
  FileLimit      = <integer>;
  DataLimit      = <integer>;
  StackLimit     = <integer>;
  TerminateAtReset = YES | NO;
  SharedMemory   = YES | NO;
  TestExtensions = YES | NO;
  BackingStore   = YES | NO;
  SaveUnders     = YES | NO;
  NoBanner       = YES | NO;
```

SnfFormat specifies the format the X-Server assumes for *.snf font files. Those files contain no information about bitorder of padding sizes. Hence it is necessary to force them into a specific format. The default is always the format that **Xvga** uses internally.

FileLimit limits the number of open files (and therefore the number of client connections as well).

DataLimit limits the data space to the given size in Kb.

StackLimit limits the stack space to the given size in Kb.

TerminateAtReset changes behavior when the last client of a session disconnects. Normally when this happens the X-Server goes through a reset-cycle. However sometimes it is useful that the server terminate instead of going through this reset cycle.

SharedMemory enables or disables the shared memory related X-extensions. This might be useful if the shared memory facilities are not present.

TestExtensions enables or disables the playback/record related X-extensions. This is useful if strict security is to be enforced. By default all those extensions (XTEST, XTestExtension1, DEC-XTRAP) are enabled.

BackingStore enables or disables the use of backingstore X-Server wide.

SaveUnders enables or disables the use of saveunders X-Server wide.

NoBanner enables or disables the initial startupbanner **\$(XVGAHOME)/etc/Xlogo**.

```
[SCREEN]
  Visual      = DEFAULT |
               STATICGRAY |
               GRAYSCALE |
               STATICCOLOR |
               PSEUDOCOLOR |
               TRUECOLOR |
               DIRECTCOLOR;
  SoftwareCursor = YES | NO;
  FontCache      = YES | NO;
  StippleCache   = YES | NO;
  TileCache      = YES | NO;
```

```
PatternCache      = YES | NO;
PixmapCache      = YES | NO;
FbOnly           = YES | NO;
GrayVisOnly      = YES | NO;
EnergyStar       = YES | NO;
StandByTime      = <integer>;
SuspendTime      = <integer>;
OffTime          = <integer>;
ScreenSize       = <integer>x<integer>;
Desktop          = <integer>x<integer>;
Depth            = <integer>;
Monitor          = <string>;
Board            = <string>;
```

Visual is the default visual for X-Clients. If set to **Default** then the default visual for that depth is used. For depth 4 this is **StaticColor**, for 8 it is **PseudoColor** and for 15, 16 and 24 it is **TrueColor**. Note that in depth 4, the default **StaticColor** mapping is setup in a way so that popular Desktops will run best. If **StaticColor** is specified explicitly a 1/2/1 RGB mapping is used, which is very useful for running PEX clients.

SoftwareCursor enables the forced use of a software cursor. This comes in handy in cases, where the hardware should support a hardware cursor, but for some reason (like timing-problems) it doesn't work correctly.

GrayVisOnly forces the X-Server to expose only StaticGray or GrayScale visuals to the clients. This is very useful for running PEX in 4bpp or 8bpp. In those depths there are not enough colors per channel for reasonable rendering quality. If a GrayVisual is enforced PEX clients will use a Color to Monochrome mapping.

EnergyStar, **StandByTime**, **SuspendTime** and **OffTime** control the DPMS screen saving feature. Basically the DPMS standard defines multiple states: ON (Normal display), STAND_BY (no HSync pulse), SUSPEND (no VSync pulse) and OFF (no HSync and VSync pulses). The normal screen-saver timeout is the time to go from the ON to the STAND_BY state (note that for some non 100% DPMS graphics chips this state is only screen blanking). SuspendTime is the time from STAND BY to SUSPEND, and OffTime is the time from SUSPEND to OFF. The times are given in seconds. By default SuspendTime is 900 and OffTime is 1800.

ScreenSize is the width and height of the physical screen in millimeters. Note, that not using that entry causes the server to assume a 75dpi display.

Desktop is the size of the virtual desktop. This can be much larger than the physical screen.

Depth is log 2 of the number of colors to be used. This can be thought of as the number of bitplanes used to represent color, or the number of bits used by the RAMDAC. Raising 2 to the power of this number yields the maximum number of colors that the Server will show with this board.

Board is the path to the xqa-file relative to **\$XVGPHOME/boards**. For the Peritek VFX-M, the string is "**number9/vfx-m.xqa**".

Monitor is the path to the vda-file relative to **\$XVGPHOME/monitors**. If you use an IBM VGA monitor you would supply the string "ibm/vga.vda".

[RESOLUTIONS]

```
<integer>x<integer> list;
```

A list of width by height pixel resolutions (e.g., 640 x 480, 1280 x 1024), one per line, which should match those in the board description file. The X server starts up with the first resolution on the list that is supported by the board and the monitor. Other supported resolutions may be switched to while the X server is running using the **HotKeyZoom** sequence.

3.4 Monitor description files

The PX Windows distribution includes a variety of monitor description files in the **PeritekX/monitors** directory tree. If your monitor is not included, please contact Peritek for a recommended file or for assistance creating a new monitor description file.

Monitor description (**.vda**) files are VESA Display Information Format files. VDIF files contain a common set of parameters that define a display sufficiently for a video controller to configure itself for a optimal monitor timing. VDIF also provides information about the display geometry and color characteristics. A VDIF file consists of a general description section and a number of Operational Limit Sections, each of which may cover several Preadjusted Timing Sections. The structure of a sample VDIF file is shown in Figure 3–2.

```
[VERSION]
[MONITOR_DESCRIPTION]

[OPERATIONAL_LIMITS]
  [PREADJUSTED_TIMING]
  [PREADJUSTED_TIMING]

[OPERATIONAL_LIMITS]
  [PREADJUSTED_TIMING]

[OPERATIONAL_LIMITS]
  [PREADJUSTED_TIMING]
  [PREADJUSTED_TIMING]
  [PREADJUSTED_TIMING]

[GAMMA_TABLE_ENTRIES]
[GAMMA_TABLE]
```

Figure 3–2 Structure of a monitor description (.vda) file.

The syntax for each section of the file is shown in Appendix A.

3.5 Board description files

The PX Windows distribution includes board description (**.xqa**) files for Peritek boards in the **PeritekX/boards** directory tree. These Extended Query Architecture files are designed to describe the capabilities of the graphics hardware at a very low level. Hardware limitations are explicitly given in this file as well as the different modes of operation (depths) the hardware can use. The overall structure of these files is shown in Figure 3–3.

```
[ADAPTER]
[ADAPTER_INFO]
    [PCLK_TABLE] (optional)

[VISUAL]
    [RESOLUTIONS]
    [DESKTOPS]

[VISUAL]
    [RESOLUTIONS]
    [DESKTOPS]
...

```

Figure 3-3 Overall structure of board description (.xqa) files

You should not need to alter these files. If you have questions, please contact Peritek for assistance.

3.6 Command-line configuration

Most configuration of the **Xvgp** server should be done using the **Xvgp.ini** file rather than with command-line options. The **Xvgp** man page in Chapter 5 lists all the command-line options; most of them are redundant with either the **Xvgp.ini** file or with the **xset** client. See the man page for **xset** in your X Windows System installation. You may wish to use command-line options to disable access control with **-ac**, for testing the server from remote clients before you have access control set-up (see Section 4.4). Or you may wish to control the screen saver, using the **-s**, **v** and **-v** options.

For **VxWorks**, “command-line flags” are not given on the command line but are placed in a special file, **Xvgp.config**. A sample file that turns off access control and sets the screen saver timeout to 10 minutes is included in the software distribution in the directory **PeritekX/arch/VxWorks/Xvgp.config**.

3.7 Kernel configuration (VxWorks)

At the present time, **VxWorks BSPs** for PowerPC processors boards vary considerably in the support they provide for PMC graphics. Some do not read the PCI header and program the base address registers for the request, or do not map sufficient PCI memory for the frame buffer. If your system required changes to kernel files (other than **config.h**) to allow the **Xvgp** server to run, the changed files will be placed in a directory with the BSP name (e.g. **mv2604**) that is a subdirectory of **os_specific/vxworks** on the Peritek distribution. A **README** file in that directory will explain what changes were made and why. Use these files as a guide in changing your own BSP before recompiling your kernel.

For reference, Table 3-4 shows the extra definitions that Peritek adds to **config.h** in all its **VxWorks** test systems.

Definition added to config.h	Description
-------------------------------------	-------------

INCLUDE_DEBUG	native debugging
INCLUDE_LOADER	object module loading
INCLUDE_NET_SYM_TBL	load symbol table from network
INCLUDE_NFS	nfs package
INCLUDE_NFS_SERVER	nfs server
INCLUDE_NFS_MOUNT_ALL	automatically mount all NFS file systems
INCLUDE_PING	ping() utility
INCLUDE_RLOGIN	remote login
INCLUDE_SHELL	interactive c-expression interpreter
INCLUDE_SHOW_ROUTINES	show routines for system facilities
INCLUDE_SPY	spyLib for task monitoring
INCLUDE_STARTUP_SCRIPT	execute start-up script
INCLUDE_SYM_TBLM_TBL	symbol table package error status
INCLUDE_TELNET	telnet-style remote login
INCLUDE_UNLOADER	object module unloading

Table 3-4 Extra INCLUDEs from configAll.h used by Peritek in config.h (VxWorks)

3.8 Xdec server and ws configuration (Tru64 Unix only)

The preliminary release of the VFX driver for Tru64 Unix, **PERVFX1.0.0**, can be configured to a variety of graphics modes when the kit is built at Peritek, but does not yet include the correct configuration interface to allow different screen sizes and depths to be set by specifying flags to the server. This deficiency will be corrected in the next minor release. The default graphics mode as shipped is 1280 x 1024 screen size, 8 bits per pixel, 60 Hz. Please contact Peritek if you need a different graphics mode or the ability to change graphics modes for your application.

The mouse and keyboard ports on the VFX-M are not supported by the Tru64 UNIX driver. Currently you need to use the mouse and keyboard ports on the processor board when you run X. Support for the VFX-M PS/2 ports will be added in a future release of the driver.

The driver enables the Tru64 X Server to access the VFX-M's color map, display controller registers, and frame buffer. Note that at the time of this writing, the driver relies on the X Server's CFB (Color Frame Buffer) software to perform all drawing operations. It is, therefore, not a hardware-optimized server, and does not use any I128S2 accelerated graphics functions.

Current software for the VFX-M Graphics Board under Tru64 UNIX consists of a functional but as yet unoptimized driver that operates through Tru64 UNIX's Workstation (WS) environment for the X Windows server. The installation instructions in section 2.5 configure the kernel so that the VFX-M will be accessed by the **ws** workstation driver. For information about configuring the X server, type **man Xdec** on the command line and see the *X Window System Environment Programming Guide* on your **Tru64 Unix** system's Software Documentation CD-ROM.

4 Running the Xvgp server

The **Xvgp** server can either be started by individual users or started by the operating system at boot-up and controlled by a login process. The details of doing this are system dependent, and should be documented in your overall X Window System installation. If the descriptions for your operating system that follow, combined with your system documentation, are not enough to get you started, please contact Peritek for assistance.

4.1 Command line start-up (LynxOS and Unix)

The Peritek VGP-series boards do not currently function as a system text-only console on all systems. In such cases, command line start-up is an option only from a serial console, another X display or from a remote login. The command

```
% Xvgp :0
```

executed as root will start executing the server as X display number 0 on the system, providing **Xvgp** is in your path. If another X display is already started, use :1 to start the **Xvgp** server as display number 1.

The above command will cause the server to start, display the X logo and an X cursor. It will not start any clients or window managers, and therefore is not a very useful way to start the server except as a test of server functioning. Your X Window System installation should provide **startx** or **xinit** scripts that start the server with a window manager and terminal client.

For users other than root to start the X server, the **Xvgp** file must be installed setuid root. The command to accomplish this, for an **Xvgp** file installed in **/usr/bin/X11** is

```
# chmod 4755 /usr/bin/X11/Xvgp
```

4.2 X Display Manager start-up (LynxOS and Unix)

You can have your workstation start the X server through the X Display Manager, **xdm**, to provide a login prompt on the graphics display when the workstation boots. Check the documentation for your X Windows System for the information on starting **xdm** and editing the Xservers configuration file, since the details are system-specific.

xdm is invoked by one of the system startup **rc** files. On LynxOS, this is typically done by uncommenting the invocation of **rc.local** in **/net/rc.network**, since the Xserver must not be started until the network protocols have been initialized, and providing an **rc.local** that invokes **xdm**. A sample **rc.local** file that loads the device drivers for the **VFX-M** and starts **xdm** is included in the Peritek software distribution in the directory **os_specific/lynx**.

When started, **xdm** will consult **/usr/lib/X11/xdm/xdm-config**, its configuration file. This file contains the names of files which contain information to specify various X resources, and typically contains several lines, such as:

```

DisplayManager.servers:      /usr/lib/X11/xdm/Xservers
DisplayManager.errorLogFile: /usr/lib/X11/xdm/xdm-errors
DisplayManager*resources:   /usr/lib/X11/xdm/Xresources
DisplayManager*startup:     /usr/lib/X11/xdm/Xstartup
DisplayManager*reset:       /usr/lib/X11/xdm/Xreset
DisplayManager*session:     /usr/lib/X11/xdm/Xsession

```

The name of the file containing the list of displays is referenced as **DisplayManager.servers**. The file **Xservers** will contain one line for each X server to be started. Sample **Xservers**, **Xresources** and **Xsession** files are also provided in the Peritek **os_specific/lynx** directory. Use these files to replace or as guides to modify the corresponding files already installed as part of your X Windows System.

Starting **xdm** under Linux is similar, except that the **lib** and **bin** directories are located at **/usr/X11R6/lib** and **/usr/X11R6/bin**. Sample files may be found in **os_specific/linux**.

4.3 Start-up scripts (VxWorks)

Starting the **Xvga** server under **VxWorks** is rather different than for LynxOS or Unix. Sample scripts, which may be used either on the target shell or on the Tornado host shell, are provided on the Peritek distribution in the directory **os_specific/vxworks**. The first script below is the basic script, **vfx_startx.scr**, that should be run on power-up to load the **VFX-M** device drivers and spawn the X server process.

```

# Start-up script for loading drivers and starting the Xvga server.
# Assumes BARS in VFX-M PCI header have already been programmed.
# Change bus and slot number to match the location of your Peritek
# board.
"
# These values are correct for the PMC slot on the Cetia CVME604

bus = 1
slot = 0

# Load the correct driver for your system.
ld <vfxdriver.o_cvme604

# Install the drivers. This should be done only once, on power-up

vfxDrv()
taskDelay(100)
# Graphics driver
vfxDevCreate("/vfx0", bus, slot, 0)
taskDelay(100)
# Keyboard driver
vfxDevCreate("/vfx1", bus, slot, 1)
taskDelay(100)
# Mouse driver
vfxDevCreate("/vfx2", bus, slot, 2)
taskDelay(100)

```

```
# Load the X server executable
ld <Xvgp

# Set environment variables the X server reads to find its drivers
# and configuration files. You will need to change /mypath to the
# parent directory of PeritekX on your system.

putenv("XVGPHOME=/mypath/PeritekX")
putenv("XVGPINI=/mypath/PeritekX/arch/VxWorks/Xvgp.ini")
putenv("XVGPGGRAPHICS=/vfx0")
putenv("XVGPKKEYBOARD=/vfx1")
putenv("XVGPCONFIG=/mypath/PeritekX/arch/VxWorks/Xvgp.config")
envShow()

# Increase the spawned task stack size from the PowerPC default of
# 20000 to 100000; in Peritek tests, netscape required a stack of
# 50000 to load and start executing, and x11perf required a stack of
# 100000 to complete.

spTaskStackSize=100000

# Spawn the X server task.
sp xvgp
```

If local window managers or clients are to be run, they may be added to the start-up script after the X server process is spawned.

If the X server process exits due to task deletion or the CTRL+ALT+BackSpace sequence, it may sometimes be impossible to reinitialize the drivers and restart the server without cycling the power.

On systems where neither the firmware nor the OS programs PCI base address registers and allocates PCI memory space (see Section 3.7) additional commands, found in the script `os_specific/vxworks/vfx_config.scr` and shown below, must be run *before* `vfx_startx.scr`. The correct values of `pcimembase` and `pciibase` depend on the memory mapping of your VxWorks BSP — see `sysPhysMemDesc[]` in the BSP file `sysLib.c`. The correct values for bus and slot depend on where your card is installed. The values below were tested on the Motorola MVME2604.

```
ld <vfxdriver.o_mv2604

bus = 0
slot = 16
pcimembase = 0x03000000
pciibase = 0x01080000
interrupt_line = 0x19

vfx_config(0, bus, slot, pcimembase, pciibase, interrupt_line)
```

4.4 Security

While the **Xvgp** server may be run with access control disabled (`-ac` command line option) for testing purposes, in most environments more security is desirable. The X Window System provides both client-based and host-based access control.

For client-based access control, specify an authorization file to the **Xvga** server when it is started using the **-auth** command-line flag. Consult the **xdm** and **Xsecurity** manual pages for your X Windows System installation for the file format and behavior of this authorization method.

For host-based access control, you must create one file in the **/etc** directory on the local system for each running X server (this may be an NFS-mounted directory, as for **VxWorks**). The file name is of the form **/etc/X<server>.hosts**, where **<server>** is the server number, for example, **/etc/X0.hosts**, or **/etc/X1.hosts**. This number corresponds to the display number used when the X server is started, such as **:0**, **:1**, etc.

The server hosts files contain the list of hosts allowed to connect to the particular server. Each line in each server hosts file should contain one valid hostname, for example:

```
localhost
aardvark
ferrari
morgan
alfa
maserati
```

Each of these named hosts must be capable of running clients on the display. For other operating systems, the specification of allowed hosts will be different.

4.5 Using xdpinfo to test the server

To see if your Peritek board is working, you can query it from another machine which is capable of running X clients (where **Target-machine-IP** is the IP address or name of the system with the Peritek board):

```
% xdpinfo -display Target-machine-IP:0
```

If the server is running you will get several screens of output such as.

```
% xdpinfo -display idaho:0
name of display:    idaho:0.0
version number:    11.0
vendor string:     Peritek Corporation
vendor release number: 5000
maximum request size: 262140 bytes
motion buffer size: 256
bitmap unit, bit order, padding: 32, LSBFirst, 32
image byte order:  LSBFirst
number of supported pixmap formats: 14
supported pixmap formats:
    depth 1, bits_per_pixel 1, scanline_pad 32
    depth 2, bits_per_pixel 4, scanline_pad 32
    depth 3, bits_per_pixel 4, scanline_pad 32
    ....
    depth 30, bits_per_pixel 32, scanline_pad 32
    depth 32, bits_per_pixel 32, scanline_pad 32
keycode range:    minimum 86, maximum 251
focus: PointerRoot
number of extensions: 3
```

```
SHAPE
Multi-Buffering
MIT-SUNDRY-NONSTANDARD
default screen number: 0
number of screens: 1
screen #0:
  dimensions: 1280x1024 pixels (325x260 millimeters)
  resolution: 100x100 dots per inch
  depths (14): 1, 2, 3, 4, 6, 8, 9, 12, 15, 16, 18,
24, 30, 32
  root window id: 0x28
  depth of root window: 8 planes
  number of colormaps: minimum 1, maximum 1
  default colormap: 0x26
  default number of colormap cells: 256
  preallocated pixels: black 0, white 1
  options: backing-store YES, save-unders YES
  current input event mask: 0x0
  number of visuals: 6
  default visual id: 0x23
  visual:
    visual id: 0x20
    class: StaticGray
    depth: 8 planes
    size of colormap: 256 entries
    red, green, blue masks: 0x0, 0x0, 0x0
    significant bits in color specification: 8 bits
  .....
  visual:
    visual id: 0x25
    class: DirectColor
    depth: 8 planes
    size of colormap: 8 entries
    red, green, blue masks: 0x7, 0x38, 0xc0
    significant bits in color specification: 8 bits
  number of mono multibuffer types: 6
    visual id, max buffers, depth: 0x20, 0, 8
    visual id, max buffers, depth: 0x21, 0, 8
    visual id, max buffers, depth: 0x22, 0, 8
    visual id, max buffers, depth: 0x23, 0, 8
    visual id, max buffers, depth: 0x24, 0, 8
    visual id, max buffers, depth: 0x25, 0, 8
  number of stereo multibuffer types: 0
```

If running **xdpyinfo** does not produce output similar to this, you are not talking to the board or the server is not properly loaded and running. If **xdpyinfo** works, try running an X client such as **xterm** :

```
% xterm -display Target-machine-IP:0
```

This should put up an **xterm** window on the display which is connected to the Peritek board, and the keyboard and mouse connected to the board should be active.

5 Man pages

Xvgp(1)

NAME

PX Windows **Xvgp** Server for Peritek VGP series PMC Graphics Boards

SYNOPSIS

Xvgp [:displaynumber] [option] ...

DESCRIPTION

Xvgp is a highly optimized X Window Display server for Peritek VGP series PMC graphics boards. The software is optimized for these boards and does not support other graphics adapters.

The server configuration file specifies the use of several further files or directories, as well as server options. The most significant files for the server, are those that specify the board configuration(s) and the monitor. Apart from those, the server also needs to know the mapping of colors, where to find fonts and how to connect to the mouse.

NETWORK CONNECTIONS

Xvgp currently supports connections with TCP/IP sockets. Other connections made using reliable byte-streams may be supported in the future. **Xvgp** listens on port (6000+*n*), where *n* is the display number,

OPTIONS

There are many options you can use with the **Xvgp** server. Most of them are also present in the **Xvgp.ini** file and we recommend specifying them in **Xvgp.ini** rather than using the command line option. All versions of the **Xvgp** server accept the following command line options:

-a *number*

sets pointer acceleration (i.e. the ratio of how much motion is reported to how much the user actually moved the mouse). [Default: 2]

-ac

disables host-based access control mechanisms. Enables access by any host, and permits any host to modify the access control list. Use with extreme caution. This option exists primarily for running test suites remotely. [Default: access control enabled; use **xhost**, **xauth** to change]

-audit level

sets the audit trail level. At level 1, only connection rejections are reported. Level 2 additionally reports all successful connections and disconnects. Level 0 turns off the audit trail. Audit lines are sent as standard error output. [Default: 1; cannot be dynamically configured]

-auth authorization-file

specifies a file which contains a collection of authorization records used to authenticate access. See also the **xdm** and **Xsecurity** manual pages. [Default: none]

bc

disables certain kinds of error checking, for bug compatibility with previous releases (e.g., to work around bugs in R2 and R3 xterms and toolkits). Deprecated. [Default: no bug compatibility; set **xset**]

-bs

disables backing store support on all screens. [Default: backing store enabled]

-co filename

sets name of RGB color database. [Default: **\$XVGPHOME/etc/Xrgb**; set in **Xvgp.ini**; cannot be dynamically configured]

-core

causes the server to generate a core dump on fatal errors. [Default: do not generate core; cannot be dynamically configured]

-dpi resolution

sets the resolution of the screen, in dots per inch. To be used when the server cannot determine the screen size from the hardware.

-f volume

sets feep (bell) volume (allowable range: 0-100). [Default: 50; set with **xset**]

-fc cursorFont

sets default cursor font. [Default: "cursor"; set with **xset**]

-fn font

sets the default font. [Default: "fixed"; set with **xset**]

-fp font path

sets the search path for fonts. This path is a comma separated list of directories which the X server searches for font databases. [Default: configured by **Xvgp.ini**; set with **xset**]

-help

-
- prints a usage message. [Default: `helpless`]
- I**
causes all remaining command line arguments to be ignored. [Default: `off`]
- p** *minutes*
sets screen-saver pattern cycle time in minutes. [Default: `10`; set with **xset**]
- pn**
permits the server to continue running if it fails to establish all of its well-known sockets, but establishes at least one. [Default: `requires all sockets`]
- r**
turns off auto-repeat. [Default: `auto-repeat enabled`; set with **xset**]
- r**
turns on auto-repeat. [Default: `auto-repeat enabled`; set with **xset**]
- s** *minutes*
sets screen-saver timeout time in minutes. [Default: `10 minutes`; set with **xset**]
- su**
disables save under support on all screens. [Default: `save-unders enabled`; not dynamically configurable]
- t** *number*
sets pointer acceleration threshold in pixels (i.e. after how many pixels pointer acceleration should take effect). [Default: `4`; set with **xset**]
- terminate**
causes the server to terminate at server reset, instead of continuing to run. [Default: `server continues`; cannot be dynamically configured]
- to** *seconds*
sets default connection timeout in seconds. [Default: `60 seconds`]
- tst**
disables all testing extensions (e.g., `XTEST`, `XTestExtension1`). Default: `test extensions enabled`]
- v**
sets blank-screen screen-saver. [Default: `enabled`]
- v**
sets drawing-screen screen-saver
- wm**

forces the default backing-store of all windows to be `WhenMapped`; a cheap trick way of getting backing-store to apply to all windows. [Default: `off`]

You can also have the X server connect to **xdm** using the X Display Manager Control Protocol (XDMCP). Although this is not typically useful as it does not allow **xdm** to manage the server process, it can be used to debug XDMCP implementations, and serves as a sample implementation of the server side of XDMCP. See your X Window System documentation for more information on XDMCP. The following options control the behavior of XDMCP.

-query *hostname*

Enable XDMCP and send Query packets to the specified host. [Default: `none`]

-broadcast

Enable XDMCP and broadcast BroadcastQuery packets to the network. The first responding display manager will be chosen for the session. [Default: `none`]

-indirect *hostname*

Enable XDMCP and send IndirectQuery packets to the specified host. [Default: `none`]

-port *port-num*

Use an alternate port number for XDMCP packets. Must be specified before any `-query`, `-broadcast` or `-indirect` options. [Default: `177`]

-class *display class*

XDMCP has an additional display qualifier used in resource lookup for display-specific options. This option sets that value, by default it is "MIT-Unspecified" (not a very useful value). [Default: `MIT-Unspecified`]

-cookie *xdm-auth-bits*

When testing XDM-AUTHENTICATION-1, a private key is shared between the server and the manager. This option sets the value of that private data (not that it is very private, being on the command line!). [Default: `none`]

-displayID *display-id*

Yet another XDMCP specific value, this one allows the display manager to identify each display so that it can locate the shared key. [Default: `none`]

SVR4 COMPATIBILITY OPTIONS

In order to be compatible with other SVR4 X-Servers **Xvgp** provides the following additional command line options:

-cmap

Ignored.

-config

Ignored. **Xvgp** always uses **\$XVGPHOME/Xvgp.ini** as the configuration file for security reasons.

-renderer

Ignored. All possible font renderers are present all the time.

-xnetaccess on

-xnetaccess off tells the server to disable host based access control (on) or to enable it (off). This is basically the same as the **-ac** option.

FONT RENDERERS

The following font file formats are supported (listed by extension):

Bitmap Fonts	*.bdf, *.snf, *.pcf
Type1 Fonts	*.pfa, *.pfb
Speedo Fonts	*.spd

EXTENSIONS

The following extensions are included by default:

- BIG-REQUESTS
- DOUBLE-BUFFER
- MIT-SCREEN-SAVER
- MIT-SHM
- MIT-SUNDRY-NONSTANDARD
- RECORD
- SHAPE
- SYNC
- X3D-PEX
- XC-MISC
- XIE
- XIdle
- XTEST
- XTestExtension1
- XsightExtension

HOT KEYS

Xvgp provides some HotKeys for controlling various features. Each of these HotKeys consists of pressing the Left ALT key together with the CONTROL key (left or right) along with a third key:

CONTROL-ALT-Numberpad+ and CONTROL-ALT-Numberpad-

These hot key sequences use the plus and minus keys on the numberpad to cycle through all the choices for resolution (screen width and height in pixels) in **Xvgp.ini**.

CONTROL-ALT-BACKSPACE

This keystroke sequence may be disabled in **Xvgp.ini**. This will kill the server immediately and terminate all clients with extreme prejudice.

SIGNALS

The X server attaches special meaning to the following signals:

SIGHUP

This signal causes the server to close all existing connections, free all resources, and restore all defaults. It is sent by the display manager whenever the main user's main application (usually an **xterm** or window manager) exits to force the server to clean up and prepare for the next user.

SIGTERM

This signal causes the server to exit cleanly.

SIGUSR1

This signal is used quite differently from either of the above. When the server starts, it checks to see if it has inherited SIGUSR1 as SIG_IGN instead of the usual SIG_DFL. In this case, the server sends a SIGUSR1 to its parent process after it has set up the various connection schemes. **xdm** uses this feature to recognize when connecting to the server is possible. Note that after startup the server will reuse this signal internally.

SEE ALSO

General information: X(1)

Protocols: *X Window System Protocol*, *The X Font Service Protocol*, *X Display Manager Control Protocol*

Fonts: *bdfpcf(1)*, *mkfontdir(1)*, *xf(1)*, *xlsfonts(1)*, *xfontsel(1)*, *xfd(1)*, *X Logical Font Description Conventions*

Security: *Xsecurity(1)*, *xauth(1)*, *Xau(1)*, *xdm(1)*, *xhost(1)*

Starting the server: *xdm(1)*, *xinit(1)*

Controlling the server once started: *xset(1)*, *xsetroot(1)*, *xhost(1)*

6 Bibliography

X Window System References

Books published by O'Reilly & Associates:

URL: <http://www.ora.com/>

X Window System Series

by various authors:

This authoritative, multi-volume set covers virtually every aspect of X Windows.

X Window System : Core and Extension Protocols : X Version 11, Releases 6 and 6.1

by Robert W. Scheifler, James Gettys (Contributor), Al Mento (Editor), Donna Converse
Paperback (February 1997)

Digital Press; ISBN: 155558148X

X Window System : Extension Library : X Version 11, Release 6 and 6.1

by Robert W. Scheifler, James Gettys (Contributor), Al Mento, Donna Converse
Paperback (March 1997)

Digital Press; ISBN: 1555581463

Technical X Window System and Motif WWW Sites (Kenton Lee)

URL: <http://www.rahul.net/kenton/xsites.html>

Peritek VGP-series Hardware and Software Manuals

URL: <http://www.peritek.com/documents.html>

VFX-M Graphics Controller Hardware Manual

VCQ-M Graphics Controller Hardware Manual

Graphics Library Package for VGP-series Graphics Boards Software Manual

7 Appendix A

Syntax of VESA Display Information Format files (.vda files)

Below is a complete description of the file format used for monitor description by the **Xvga** server (see Section 3.4) For a complete description of the parameters, see the VESA VDI standard document, or contact Peritek for assistance.

```
[VERSION]
    <integer>.<integer>;

[MONITOR_DESCRIPTION]
    Date                = <date>;
    FileRevision        = <string>;
    Manufacturer        = <string>;
    ModelNumber         = <string>;
    MinVDIFIndex        = <string>;
    Version             = <string>;
    SerialNumber        = <string>;
    DateManufactured    = <date>;
    MonitorType         = COLOR |
                        COLOUR |
                        MONOCHROME;
    CRTSize             = <integer>;           // inches

// If Color Monitor:

    RedPhosphorDecay    = <scaled>;           // msec
    GreenPhosphorDecay  = <scaled>;           // msec
    BluePhosphorDecay   = <scaled>;           // msec
    BorderRed           = <integer>;          // percent
    BorderGreen         = <integer>;          // percent
    BorderBlue          = <integer>;          // percent
    WhitePointx         = <scaled>;           // in CIExyY
    WhitePointy         = <scaled>;
    WhitePointCapY      = <scaled>;
    RedChromaticityx    = <scaled>;
    RedChromaticityy    = <scaled>;
    GreenChromaticityx  = <scaled>;
    GreenChromaticityy  = <scaled>;
    BlueChromaticityx   = <scaled>;
    BlueChromaticityy   = <scaled>;
    RedGamma            = <scaled>;
    GreenGamma          = <scaled>;
    BlueGamma           = <scaled>;

// If Monochrome Monitor:

    Phosphor1Decay      = <scaled>;           // msec
    Phosphor2Decay      = <scaled>;           // optional
    Phosphor3Decay      = <scaled>;           // optional
    Border              = <integer>;          // percent
    WhitePointx         = <scaled>;           // in CIExyY
```

```

WhitePointy          = <scaled>;
WhitePointCapY       = <scaled>;
Phosp1Chromaticityx = <scaled>;          // optional
Phosp1Chromaticityy = <scaled>;          // optional
Phosp2Chromaticityx = <scaled>;          // optional
Phosp2Chromaticityy = <scaled>;          // optional
Phosp3Chromaticityx = <scaled>;          // optional
Phosp3Chromaticityy = <scaled>;          // optional
Gamma                = <scaled>;          // optional

[ OPERATIONAL_LIMITS ]
MaxHorPixel          = <integer>;          // pixels
MaxVerPixel          = <integer>;          // lines
MaxHorAddrLength     = <integer>;          // mm
MaxVerAddrHeight     = <integer>;          // mm
VideoType            = ANALOG |
                    TTL |
                    ECL |
                    DECL |
                    OTHER ;
TerminationResistance = <integer>
WhiteLevel           = <scaled>;          // Volts
BlackLevel            = <scaled>;          // Volts
BlankLevel           = <scaled>;          // Volts
SyncType             = ANALOG |
                    TTL |
                    ECL |
                    DECL |
                    OTHER;
SyncConfiguration    = SEPARATE |
                    C |
                    CP |
                    G |
                    GP |
                    OTHER;
SyncLevel            = <scaled>;          // Volts
MaxPixelClock        = <scaled>;          // MHz
MinHorFrequency      = <scaled>;          // kHz
MaxHorFrequency      = <scaled>;          // kHz
MinVerFrequency      = <scaled>;          // Hz
MaxVerFrequency      = <scaled>;          // Hz
MinHorRetrace        = <scaled>;          // usec
MinVerRetrace        = <scaled>;          // msec

[ PREADJUSTED_TIMING ]
Preadjustedtimingname = <string>;
HorPixel             = <integer>;          // pixels
VerPixel             = <integer>;          // lines
HorFrequency         = <scaled>;          // kHz
VerFrequency         = <scaled>;          // Hz
PixelClock           = <scaled>;          // MHz
CharacterWidth       = <integer>;          // pixels
HorAddrLength        = <integer>;          // mm
VerAddrHeight        = <integer>;          // mm
PixelWidthRatio      = <integer>;          // gives H:V
PixelHeightRatio     = <integer>;

```

```

ScanType          = INTERLACED |
                  NONINTERLACED |
                  OTHER;
HorSyncPolarity   = NEGATIVE |
                  POSITIVE;
VerSyncPolarity   = NEGATIVE |
                  POSITIVE;
HorTotalTime      = <scaled>;           // usec
HorAddrTime       = <scaled>;           // usec
HorBlankStart     = <scaled>;           // usec
HorBlankTime      = <scaled>;           // usec
HorSyncStart      = <scaled>;           // usec
HorSyncTime       = <scaled>;           // usec
VerTotalTime      = <scaled>;           // msec
VerAddrTime       = <scaled>;           // msec
VerBlankStart     = <scaled>;           // msec
VerBlankTime      = <scaled>;           // msec
VerSyncStart      = <scaled>;           // msec
VerSyncTime       = <scaled>;           // msec

[GAMMA_TABLE_ENTRIES]
<integer>;

[GAMMA_TABLE]
<integers>

```

8 Index

A

access control..... 30, 34
 authorization..... 31

C

client 8, 32
 client-server architecture..... 8
 command line flags..... 26
 Compaq..... 16, 27
 contact information..... 9

F

font path..... 20
 framegrabber..... 13

H

hosts file..... 31

K

keyboard 21

L

Linux..... 9, 12
 LynxOS..... 14, 28

M

monitor..... 25
 monitor resolution..... 21, 25
 mouse..... 22

P

PeritekX..... 11, 15
 PERVFX100..... 16
 platforms..... 9

R

RGB path 20

S

screen saver..... 35
 security..... 30
 signals 38

T

tablet 22
 technical support..... 9
 touchscreen 23
 Tru64 Unix..... 16, 27
 TV video 13

V

vda files..... 19, 25
 visual..... 24
 VxWorks..... 15, 26, 29

W

ws..... 16

X

X Window System 8
 xdm 28
 xdm-config..... 28
 xdpinfo 31
 xqa files..... 19, 25
 Xservers 29
 xterm..... 32
 Xvgp.ini 18, 19