

**vtServer<sup>®</sup>**  
vtAlpha/vtVAX Bare Metal  
V4.0.0

---

**X Server Configuration**  
Technical Note

BN-0008-03

X Server Configuration

BN-0008-03 (March 7, 2019)

© 2019 Vere Technologies LLC

*vtAlpha and vtVAX are marketed jointly by AVT and Vere Technologies LLC*



## Table of Contents

<b>1. Introduction .....</b>	<b>1</b>
1.1. Document Overview .....	1
1.2. Prerequisites.....	1
<b>2. X Server, DECwindows/CDE .....</b>	<b>2</b>
2.1. General Setup OpenVMS .....	3
2.1.1. Starting the DECwindows/CDE Desktop .....	4
2.1.2. Restarting the Desktop .....	5
2.2. General Setup Tru64 .....	6
2.2.1. Starting the DECwindows/CDE Desktop .....	7
2.2.2. Restarting the Desktop .....	8
<b>3. X Server Parameters and Settings.....</b>	<b>9</b>
3.1. Important Parameters .....	9
3.2. Monitor Setting .....	11
3.3. Video Drivers .....	12
3.4. Security Setting.....	14
3.5. Autostart Setting .....	15
3.6. Fonts.....	17
3.7. Keyboard .....	17
<b>4. Re-directing Display Output from OpenVMS and Tru64 .....</b>	<b>18</b>
4.1. OpenVMS .....	18
4.2. Tru64 .....	19
<b>5. Virtual Machine with One Network Interface.....</b>	<b>20</b>
5.1. Network Switch Setup.....	20
<b>6. XDISP .....</b>	<b>25</b>
<b>7. Appendix.....</b>	<b>28</b>
7.1. X Server Parameters .....	28
7.1. Server Dependent Parameters .....	32
7.2. Xdmcp Parameters .....	33

# 1. Introduction

## 1.1. Document Overview

This document describes the setup and configuration of the X Server in a virtual environment, and it explains some of the settings and a number of command-line options for X Server (X Window System display server).

This document is meant to be an extension to the vtServer Reference Manual. In the reference manual, some of the basics of the X Server configuration are explained. You can use this document for a deeper understanding and more detail of how to setup your configuration.

The DECwindows GUI (which is based on the Common Desktop Environment i.e. CDE) of OpenVMS and Tru64 runs on the X Server.

The different settings and options as mentioned in this document will not always apply to your specific situation/hardware. But you can use it to test/experiment with the available options to best fit your environment.

## 1.2. Prerequisites

If you want to setup a workstation, a few conditions have to be met.

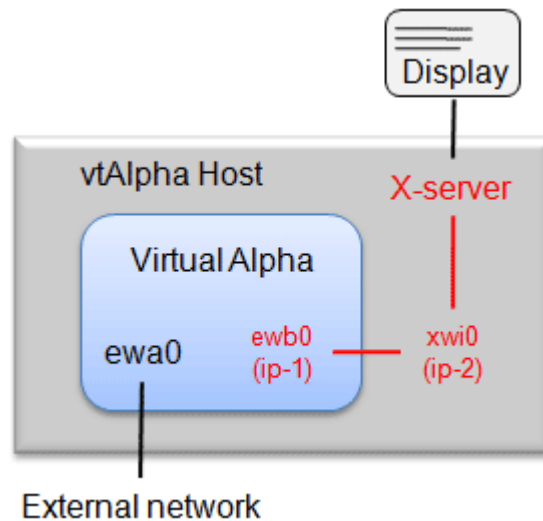
1. A working vtServer installation
2. A workstation/laptop with a graphics screen and graphics adapter. Any graphics card will (probably) work, but NVidia and ATI cards (for which we supply the drivers) will have additional features.

## 2. X Server, DECwindows/CDE

vtAlpha and vtVAX can be used to replace DEC AlphaStation and VAXstation workstations.

Instead of emulating the workstation video adapter, vtServer implements an X Windows server that interfaces to the Virtual Server using an internal network interface (xwi0). The host system's video display is used as the workstation monitor.

One notable difference between the physical and virtual workstation configurations is that the host keyboard and monitor are not used as the guest system console interface.



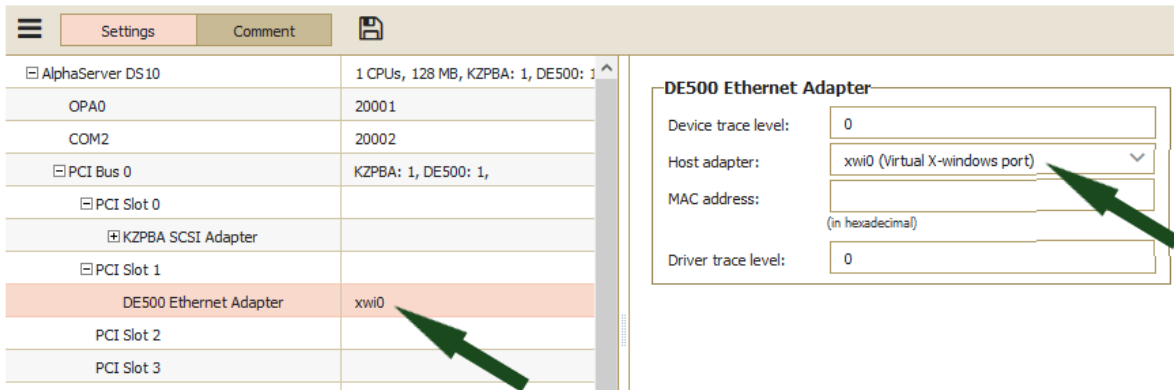
The default IP address for xwi0 is 10.143.0.1, but this can be configured to another address when so desired. The adapter on the OpenVMS or Tru64 system (in the examples used here) should have an address in the same subnet (e.g. 10.143.0.2).

## 2.1. General Setup OpenVMS

In the following example of configuring a workstation, the setup/configuration is simple. It is important to first create a *working* environment. From there you can expand and tailor the configuration to your needs.

For a simple setup follow these steps:

- Create a new Virtual Server configuration in vtServer (connect an OpenVMS disk to boot from)
- Connect the xwi0 adapter to the new Virtual Server (leave the default IP address of xwi0 at 10.143.0.1).



- Boot the new Virtual Server.
- Log in and (re)configure the network-adapter
  - If there is an existing TCP/IP configuration you do not use, remove it by deleting the “tcpip\*.dat” files from the sys\$system directory with the next command

```
$ Backup/log/delete sys$system:tcpip*.dat;* sys$manager:tcpip_dat.bck/save
```

- Start the TCP/IP configuration procedure

```
$ @sys$manager:tcpip$config
```

- Go to “Core Environment” and define the domain and interfaces (domain can be anything of your choosing)
- When defining the interface (WE0 for example) in OpenVMS, be sure to give it the IP address 10.143.0.2 and subnet 255.255.255.0.
- Now shutdown and startup TCP/IP from the top-menu

This ends all of the actions needed for the general setup for OpenVMS.

### 2.1.1. Starting the DECwindows/CDE Desktop

From the vtServer main menu, choose “Graphics environment”, then “Start workstation environment” (this starts the X Server on the vtServer). Press OK in the next screen; the screen will turn black and a mouse cursor should be visible. A login screen will eventually appear.



If you want to exit the graphics environment hold CTRL-ALT and press Backspace twice. You will return to the vtServer menu.

Starting the DECwindows/CDE desktop on your system.

1. The easiest option is to reboot the Virtual Server environment you (just) configured. And while the OpenVMS system is booting you can start the “Start workstation environment” and wait for the login screen.



If you haven't commented out the DECwindows startup from `systartup_vms.com` it will automatically start.

## 2.1.2. Restarting the Desktop

There might be occasions where you exited the DECwindows graphics environment and returned to the vtServer menu (for testing parameters for example). If you try to start the Workstation environment, the screen will (in most cases) stay black, i.e. the login-screen will not appear.

So, on an already running OpenVMS system where the login screen doesn't appear (after choosing "Start workstation environment"), follow these steps to restart the DECwindows environment:

- Open a terminal session (or connect to the primary console) of the OpenVMS system and log in.
- Write a small procedure (e.g. startdt.com) with a small delay in it (this delay isn't necessary in a Tru64 environment).

```
$ Wait 00:00:20 ! 20 sec. wait (you can vary the number of seconds)
$ @sys$manager:decw$startup
$ exit
```

- Start the DECwindows environment by running this procedure. Make sure this procedure runs after you choose "Start workstation environment".

```
$ @startdt
```

- Now exit the terminal/primary console (press CTRL-] and type quit) to the vtServer menu and start the "Start workstation option". The login screen should appear after a few seconds.

You should have a working X Server environment now. Use this environment to tailor the configuration and settings so it meets your workstation needs.



If you want to exit the graphics environment (and return to the vtServer menu), hold CTRL and ALT and press Backspace twice.  
Return to the graphics environment by starting the X Server again via "Graphics" – "X-server start".

## 2.2. General Setup Tru64

The previous chapter describes the general setup instructions and specifics for OpenVMS. The general setup also applies to a Tru64 environment. This chapter will explain some of the Tru64 specifics.

If this is a new installation, make sure you define your network interface card with the proper IP address. Log in your Tru64 system (by terminal or primary console).

Run “setup” from the prompt (#); choose “custom”, then “Network Setup Wizard”. Define the network adapter (e.g. tu0) with IP address 10.143.0.2. And specify a host name.

The following setup allows the system to start your system up with the DECwindows/CDE desktop.

- Add the following line to /usr/dt/config/Xservers
  - **10.143.0.1:0 foreign**
- The Xservers file is read-only, so change the permission first to write/modify
  - `chmod +w Xservers`
  - `chmod -w Xservers` (to restore the original permissions)
- Before you can log in to the DECwindows/CDE environment with the “root” user, you have to allow root to log in from that source. Add the following line to /etc/securettys
  - 10.143.0.1:0
- Add your system in the /etc/hosts file. Add the following line
  - 10.143.0.2 <system-name>

Where <system-name> is the hostname which you defined when you configured your network interface card

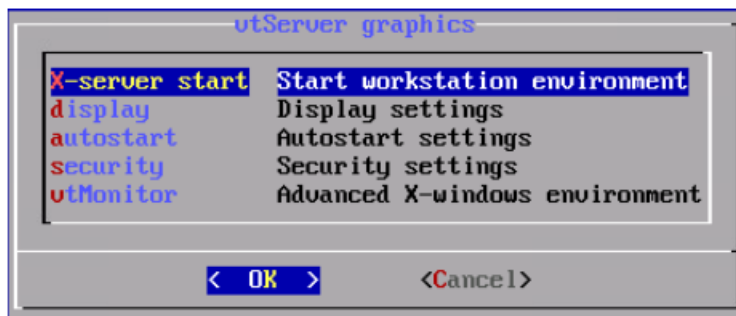


The display can also be referenced by setting the variable DISPLAY to 10.143.0.1:0.0 (when the IP address of the interface is customized, adjust the addresses above accordingly).



## 2.2.1. Starting the DECwindows/CDE Desktop

From the vtServer main menu, choose “Graphics environment”, then “Start workstation environment”.



Press OK in the next screen, a login screen will appear.

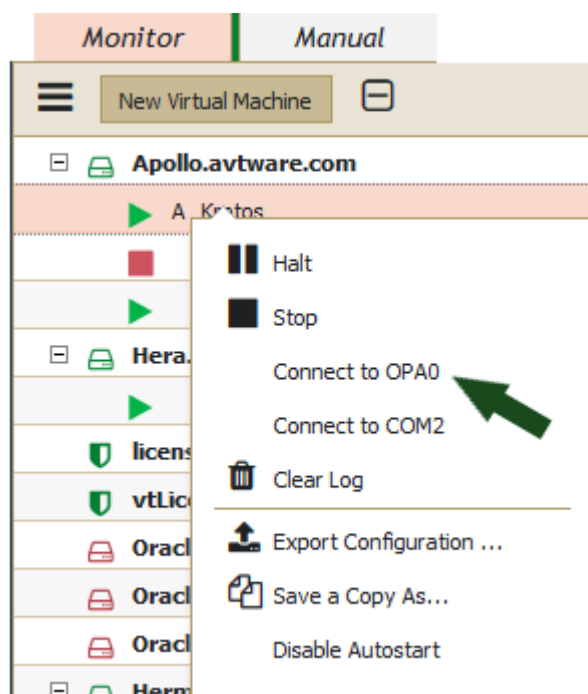
If you want to leave the graphics environment hold CTRL-ALT and press Backspace twice. You will return to the vtServer menu.

## 2.2.2. Restarting the Desktop

If for some reason the login screen does not appear, you can reset and restart the desktop login-sequence. Open a terminal session **to the Tru64** system (or connect to the “primary console”) and go to the directory `/usr/dt/bin` and type:

```
./dtconfig -reset  
./dtconfig -start (not always necessary, the “-reset” may suffice)  
./dtconfig (to see additional options of what dtconfig can do)
```

You can connect to the “primary console” via the “vtNavigation” window.



Press CTRL-] to exit the console. Next type “quit” to return to the menu.

## 3. X Server Parameters and Settings

### 3.1. Important Parameters

Getting the CDE environment to work is no guarantee your application will work. Maybe you need to redirect the “display” output to the “new” IP address of the X Server. Or you have to use the X Server parameters which your application requires.

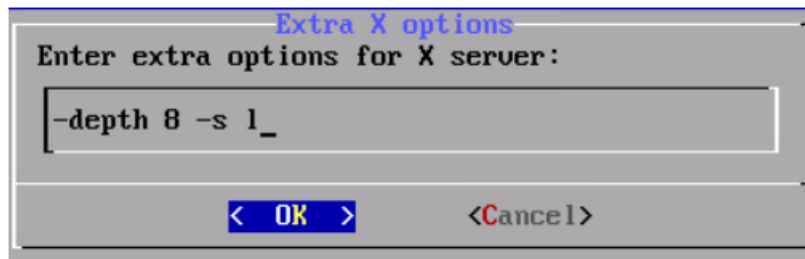
In this chapter we will cover how to use the parameter entry field and some of the parameters. In the next chapter we will discuss how to redirect display methods.

Many X Servers have device specific command line options. In this chapter, we’ll cover the parameters that proved to be important and essential in the installations and configurations of vtServer that we handled ourselves.

There are many more parameters that can influence the behavior of an X Server, but they haven’t proven to be important for a vtServer installation so far. An overview of these other parameters are to be found in the appendix.

The parameters are used by entering them in the field “Extra X options”. To get there: From the vtServer menu → Graphics environment → Display settings → Extra X Server parameters

Example X Server parameters:



Some of the options can have unexpected results. So, when you start to configure an X Server environment, do not use any extra parameters. Use and apply them one at a time until the configuration is to your liking.

- `-cc <class>` (class is an integer ranging from 0 to 5)
  - sets the visual class for the root window of color screens (default color visual class). The class numbers are as specified in the X protocol (not obeyed by all servers).
    - 0=StaticGray
    - 1=GrayScale
    - 2=StaticColor
    - 3=PseudoColor
    - 4=TrueColor
    - 5=DirectColor
- `+bs`
  - Enable any backing store support (on all screens)
- `-bs`
  - Disable any backing store support (on all screens). Backing store is disabled by default
- `-depth <depth>` (depth is an integer, default: 8)
  - Set color depth. This entry specifies what color depth the Display subsection is to be used for. This entry is usually specified, but it may be omitted to create a match-all Display subsection or when wishing to match only against the `FbBpp` parameter. The range of depth values that are allowed depends on the driver. Most drivers support 8, 15, 16 and 24. Some also support 1 and/or 4, and some may support other values (like 30). Note: Depth means the number of bits in a pixel that are actually used to determine the pixel color. 32 is not a valid depth value. Most hardware that uses 32 bits per pixel only uses 24 of them to hold the color information, which means that the color depth is 24, not 32.
- `+extension <extension name>` (name options: RENDER, COMPOSITE, SHAPE, SYNC Xinput)
  - Enable extension
- `-extension <extension name>` (name options: RENDER, COMPOSITE, SHAPE, SYNC Xinput)
  - Disable extension

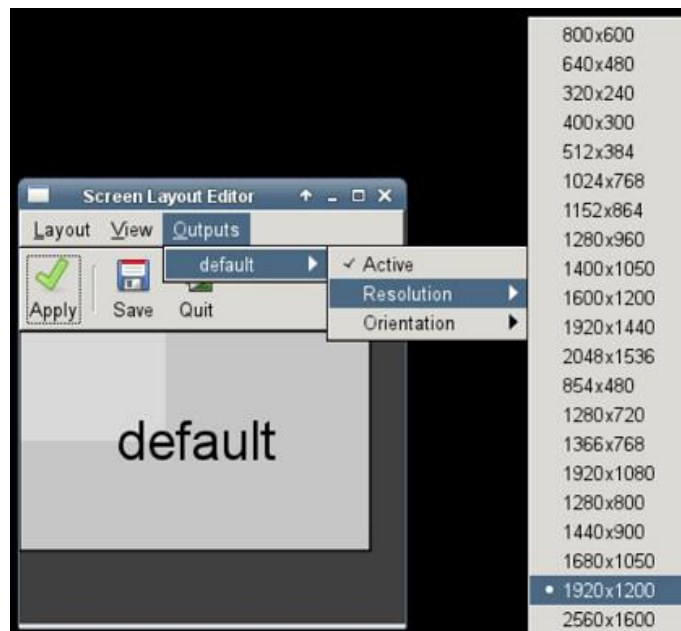
### 3.2. Monitor Setting

Configuration of the host video display devices is performed using the vtServer configuration menu (Main menu → Graphics Environment → Display settings → Monitor settings).

The position of the monitor(s) can be changed by clicking on one of the monitor window icons and dragging it to the correct position. Right click on a monitor window icon to display options to enable/disable the selected monitor or to change the display resolution.

“Apply” applies the settings without permanently saving them. When you are satisfied, select the “Save” option, then click on the ‘X’ button in the upper right corner of the window to close it.

The picture below shows the default “Screen Layout Editor”. When you are using a supported Nvidia or ATI card (and have installed the appropriate drivers) the layout will be different and will have more options.



### 3.3. Video Drivers

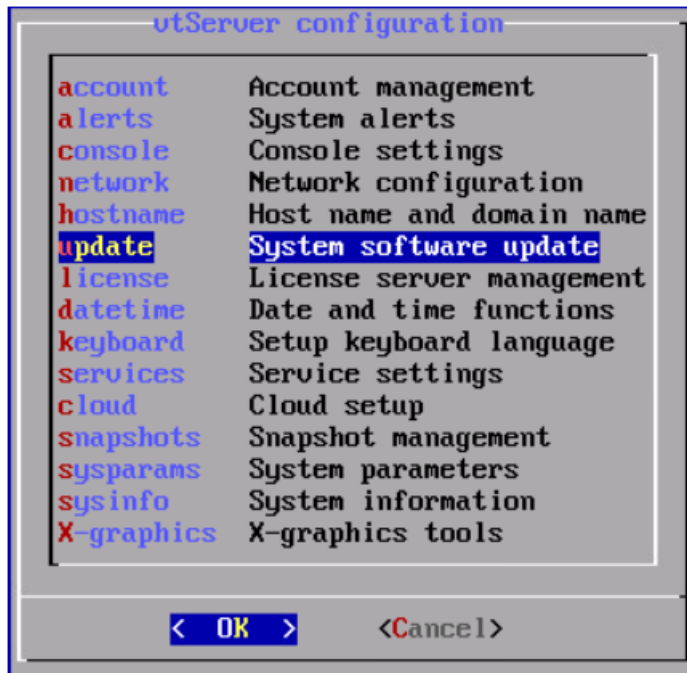
If you have a workstation with an Nvidia or ATI graphics card, we recommend installing the Nvidia or ATI drivers. These can be found on the vtServer installation DVD and USB.

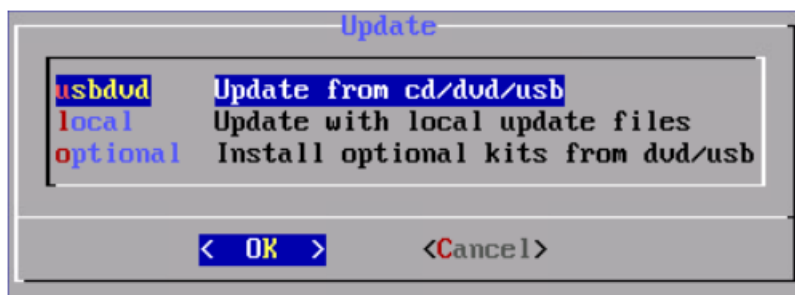
Before installing the driver, check if your card is supported by the supplied driver. Each driver kit has an accompanying text file with a list of supported adapters. Or go to our website where you can find the most recent update of the text files:

<https://www.avtware.com/downloads>

Follow these steps for installing (or removing) the driver:

1. Insert the vtServer installation DVD or USB in the vtServer system
2. Start the Console menu
3. Go to the vtServer configuration and select update





4. Select the source where the drivers are located.

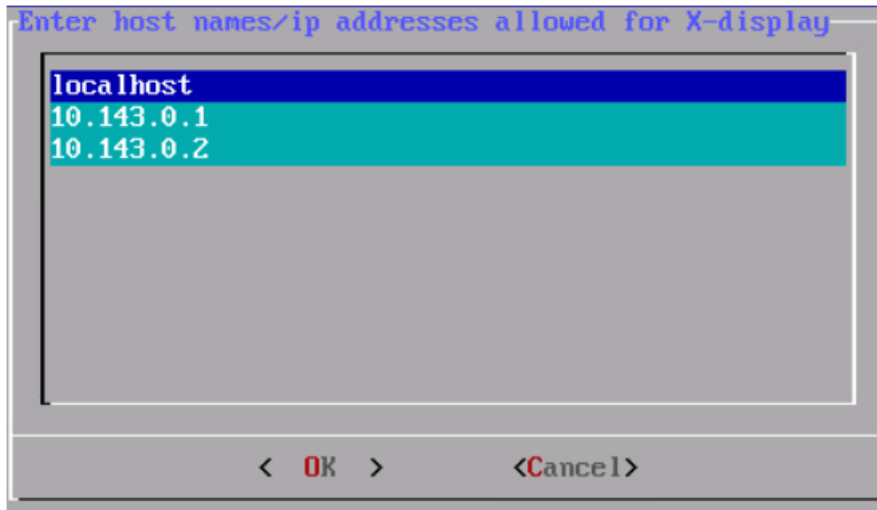
Now choose to install the ATI or NVidia driver (or "remove\_graphics" if you want to uninstall the previous installation of the driver). The installation will commence and notify you if the installation succeeded.



If you update vtServer, make sure you install the drivers that accompany the installation DVD and USB.

### 3.4. Security Setting

The default X Server security settings restrict access to the server to localhost (127.0.0.1), 10.143.0.1 and 10.143.0.2; the vtServer configuration menu (main menu → graphics → security) may be used to add access from other IP addresses, as needed.

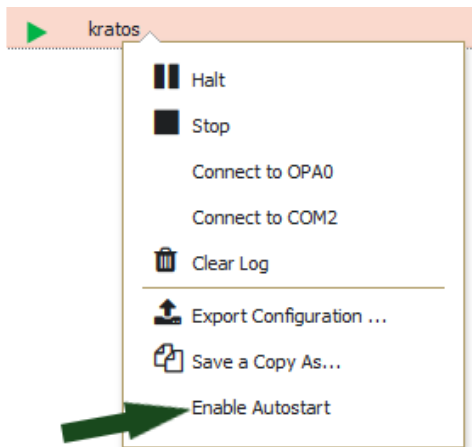




### 3.5. Autostart Setting

If you want to automatically start the system as a workstation you have to take the following three steps (the order in which you take them isn't important):

1. Enable the emulator "Auto-start upon host boot" for the instance in the user interface. This can be done via the vtMonitor web interface. Right-click on the system/instance on which you want to enable the auto-start function, and choose "Auto-start upon host boot". The capital "A" should appear in front of the system's configuration name.

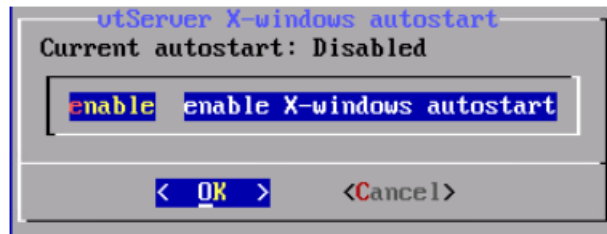


Starting a configuration does not automatically mean that the configured Virtual Server will boot its operating system. The setting of the default boot device and the auto\_action switch will boot the system.

2. Enable autoboot for the OpenVMS or Tru64 operating system in the SRM-console by entering a default boot device and setting AUTO\_ACTION to RESTART.

```
P00>>>
P00>>>
P00>>> set bootdef_dev dka0
P00>>> set auto_action RESTART
P00>>>
```

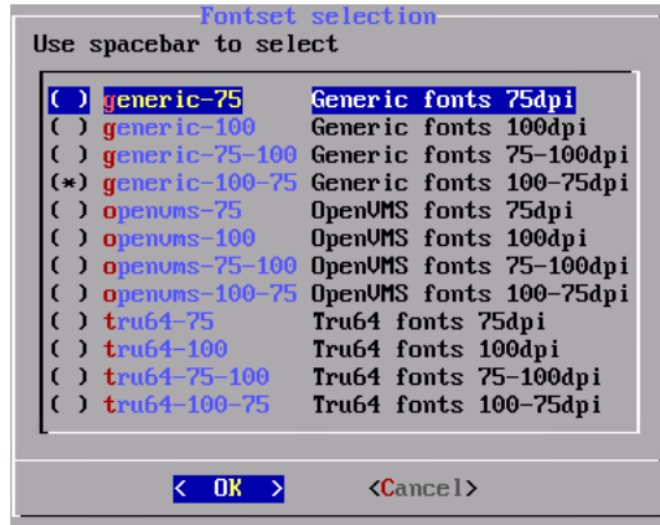
3. Enable the X-windows for autostart in the vtServer console (main menu → graphics → autostart).



Enabling the X-windows autostart feature will automatically start the X Server. It will result in displaying a graphical CDE/DECwindows login-panel at the end of the boot cycle, if the setup is correctly done on OpenVMS or Tru64.

### 3.6. Fonts

The font option (from the configuration menu → X-graphics → display → font) allows you to choose between **Generic**, **OpenVMS** or **Tru64** fonts.



### 3.7. Keyboard

The keyboard option (from the configuration menu → X-graphics → display → keyboard) is used to select the desired keyboard. **None** means use the default; the options **lk4xx\_ps2** or **lk4xx\_usb** selects that type of keyboard, and **ps2dec** selects a keyboard with all 20 function keys enabled, allowing use of F13 – F20 when pressing **right-Alt +** (F3 - F10).



## 4. Re-directing Display Output from OpenVMS and Tru64

### 4.1. OpenVMS

Once you've got DECwindows running, you can start DECwindows programs from your terminal. The output will be shown on your workstation screen. These "exe"-programs can be found in `sys$system` and start with `decw*`, and in `decw$examples:*.exe`

Take the following steps (TCP/IP must be running):

1. Create the file `SYSSPECIFIC:[SYSMGR]DECW$DEVICE.COM` with the following content:

```
$ decw$device == "vtserver"
```

(Replace `vtserver` with the IP host name or address of the vtServer host)

2. Create the file `SYSSPECIFIC:[SYSMGR]DECW$PRIVATE_APPS_SETUP.COM` with the following content:

```
$ define/nolog/system decw$default_transport tcpip  
$ decw$appnode == "10.143.0.1"
```

(if the default IP address for `xwi0` is changed, replace `10.143.0.1` with the `xwi0` address)

The following example shows how to create a display for a single X Windows application and start a DECwindows application from the OpenVMS command-prompt (the program and output will run on the graphics-screen of your station):

1. Create a display:  

```
$ set display/create/transport=tcpip/node=10.143.0.1
```
2. Start a program (find more in the `sys$system` and `decw$examples` directories):  

```
$ run sys$system:decw$clock  
$ run decw$examples:ico.exe (this program creates more output when tracing for errors)
```

## 4.2. Tru64

In Tru64 you can redirect/start programs from the terminal to your graphical DECwindows/CDE session. The output will be shown on your DECwindows/CDE workstation screen. Some of the programs can be found in **/usr/dt/bin**.

Take the following steps in a terminal session (TCP/IP must be running):

1. First create a DISPLAY variable

- a. `DISPLAY=10.143.0.1:0.0;export DISPLAY`

2. Start a program

- a. `cd /usr/dt/bin`

- b. `./dticon`

(the icon program will now run in your CDE session)

## 5. Virtual Machine with One Network Interface

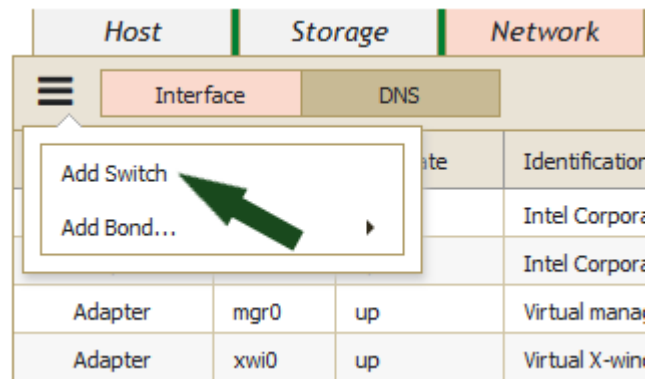
Some virtual machines support only one network interface.

The typical graphics configuration requires the use of two virtual network interfaces on the VAX/Alpha system: one for the general network traffic, and one dedicated to the X Windows server. The VAXstation 4000 model 90 supports only one network adapter. To configure the graphics on this system a virtual switch is used to allow both the primary network and the X Windows server to use the single virtual adapter.

### 5.1. Network Switch Setup

The configuration instructions below are performed using the vtMonitor interface.

- (1) Create a network virtual switch (avt0) for the X-Windows server. This is done using the Add Switch function on the Network → Interface tab.



The switch should appear in the Virtual Network Configuration list, as shown below:

Host		Storage		Network	
Interface		DNS			
Type	Name	Link state	Identification		
Switch	avt0	down	Virtual network		
Adapter	eth0		Intel Corpora		
Adapter	eth1	up	Intel Corpora		
Adapter	mgr0	up	Virtual mana		
Adapter	xwi0	up	Virtual X-win		

(2) Add interface eth1 and xwi0 to switch avt0, as shown below:

Host		Storage		Network	
Interface		DNS			
Type	Name	Link state	Identification		
Switch	avt0	down	Virtual network		
Adapter			Intel Corpora		
Adapter			Intel Corpora		
Adapter			Virtual mana		
Adapter			Virtual X-win		

- + Connect new Management Port
- ⊗ Connect eth0
- ⊗ Connect eth1
- ⊗ Connect mgr0
- ⊗ Connect xwi0
- Delete Switch

- (3) Configure xwi0 with an IP network address and mask in the same subnet as the virtual VAX system's IP address. This is done by selecting xwi0 in the Interface list, then changing the configuration information.

The screenshot shows the 'Network' configuration page in the X Server Configuration tool. The 'Interface' tab is selected, and the 'xwi0' interface is highlighted in the table. The configuration details for 'xwi0' are shown on the right, including IP address, subnet mask, and link state. A green arrow points to the IP address field.

Type	Name	Link state	Identification
Switch	avt0	up	Virtual network
Port	eth1	up	Intel Corporati
Port	xwi0	up	Virtual X-windo
Adapter	eth0	up	Intel Corporati
Adapter	mgr0	up	Virtual manage

**xwi0 adapter details**

DHCP:

IP-address: 10.10.10.250

Subnet mask: 255.255.255.0

Gateway:

Default Gateway

**xwi0 line details**

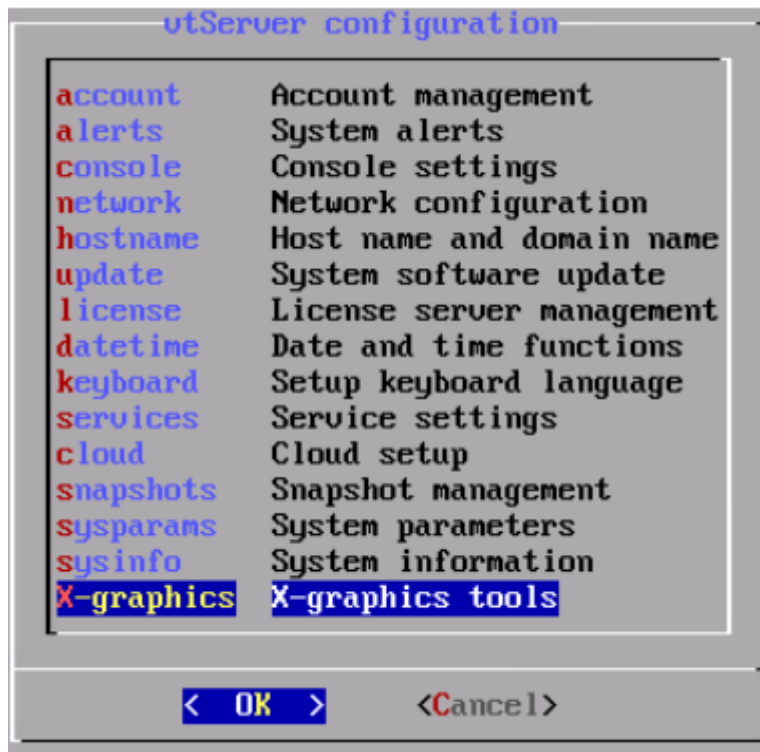
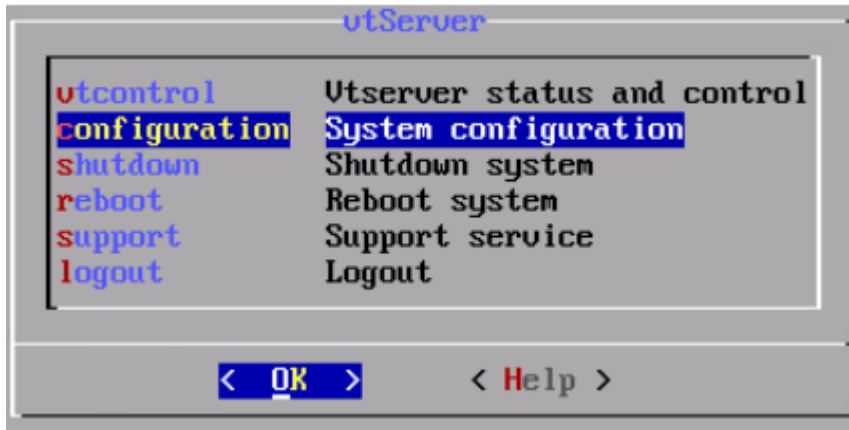
Link state: up

MTU size: 1500

Update Cancel



- (4) Use the vtServer console interface (blue screen) or a SSH connection to log in to vtServer, and move the cursor to “Graphics environment” and press OK.

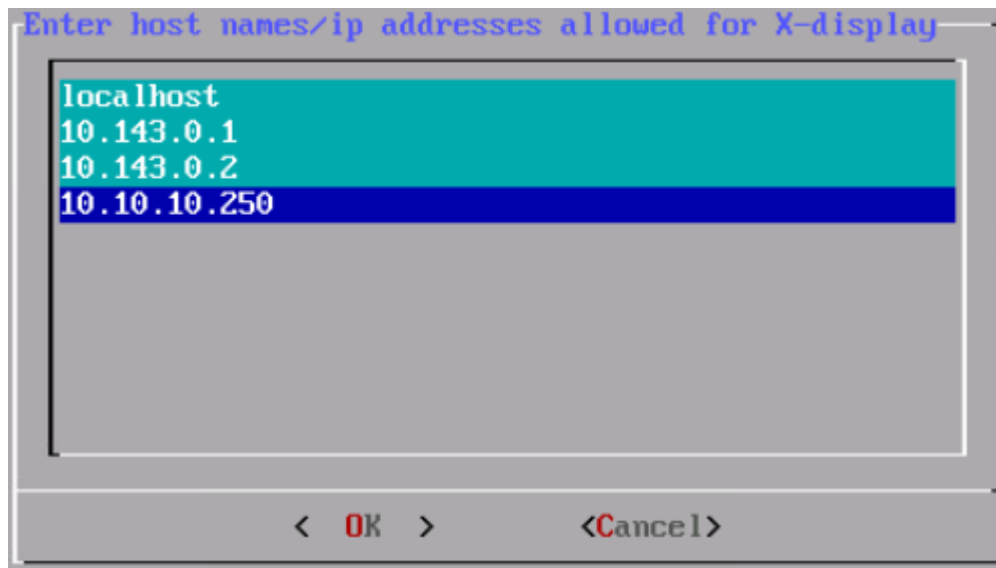


(5) Select "Security settings" and press OK.



(6) Add the IP addresses assigned to xwi0 and the virtual VAX to the authorization list.

Use the arrow keys to position the cursor at the end of the last line of the existing entries in the list, and press the Enter key. Enter the IP addresses to be added to the list. Press TAB to highlight OK, then press Enter.



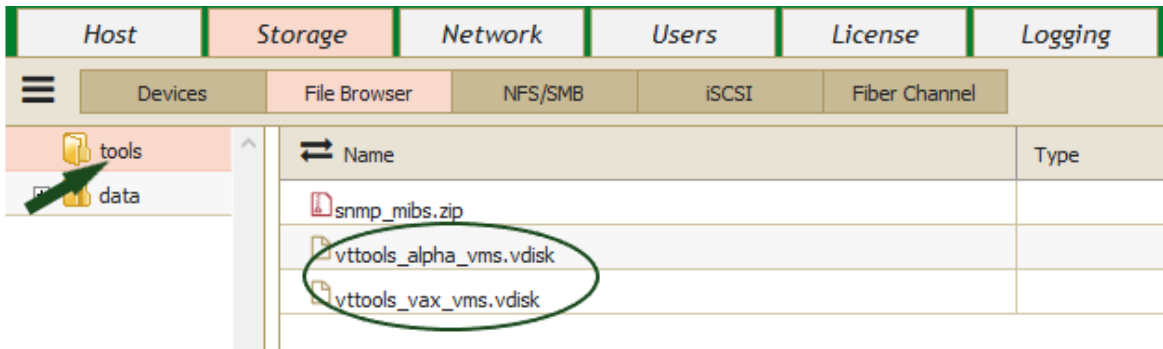
Connect the avt0 adapter to the new Virtual Server and follow the General Setup to complete the configuration.

## 6. XDISP

To make optimal use of a virtual Alpha or VAX, several added-value applications and tools are available for your use. One off these tools is XDISP.

XDISP is an extended X Windows display configuration utility for OpenVMS.

XDISP is found on the vttools logical disk:



Some applications have hard-coded assumptions about setting up a display device on OpenVMS. This may not always work in a virtual environment where we have explicit requirements, like redirecting the display to a server located on the network.

For this the 'set display' command has been modified to allow the setup of default settings that can be protected as well, thereby dictating the required parameters to an application.

The following switches have been added to the 'set display' command:

- /DEFAULT This will alter the default settings.
- /PROTECT This will force the created display device to use the parameters that have been setup as the default.

The following switch has been added to the 'show display' command:

- /DEFAULT This will show the default settings.

Example:

An application may setup the default display like this:

```
$ show display
```

```
Device:    WSA1: [super]
Node:      0
Transport: LOCAL
Server:    0
Screen:    0
```

An application that forces the display to a local display server makes it impossible to reach the desired display. For this we can setup the default characteristics of the display device like this:

```
$ set display/default/transport=tcpip/node=10.143.0.1/protect
$ show display/default
```

```
Device:    WSA0: [user]
Node:      10.143.0.1
Transport: TCPIP
Server:    0
Screen:    0
Protected: yes
```

If we create a display after that, we will force our parameters to the display device:

```
$ set display/create/transport=local/server=1
$ show display
```

```
Device:    WSA5: [user]
Node:      10.143.0.1
Transport: TCPIP
Server:    0
Screen:    0
Protected: yes
```

You can see that the parameters at display creation time are ignored because the unit is protected.

If the parameters need to be changed we first have to unprotect the display:

```
$ set display/noprotect
$ set display/node=1.2.3.4
$ show display
```

```
Device:      WSA5: [user]
Node:        1.2.3.4
Transport:   TCPIP
Server:      0
Screen:      0
Protected:   no
```

The defaults will also work if an application is directly talking to the display device since the parameters are enforced on the device driver level.

This application is supported on VAX/VMS V6.2 and up, as well as OpenVMS Alpha V6.2 and up. VMS Software Inc.'s (VSI) Alpha VMS versions are supported as well.

It is mandatory to install the VMS62TO71\_PCSI kit before installation on VMS V6.2 or V7.1.

This OpenVMS patch kit is located in the directory [VMS\_PATCHES] on the tools disk.

## 7. Appendix

The X Server options listed below are extracted from sites on the Internet. Many of the parameters in this appendix we haven't used or tested, but they could be of use for your configuration.

### 7.1. X Server Parameters

#### ***:displaynumber***

The X Server runs as the given *displaynumber*, which by default is 0. If multiple X Servers are to run simultaneously on a host, each must have a unique display number. See the DISPLAY NAMES section of the manual page to learn how to specify which display number clients should try to use.

#### ***-a number***

sets pointer acceleration (i.e. the ratio of how much is reported to how much the user actually moved the pointer).

#### ***-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.

#### ***-audit level***

sets the audit trail level. The default level is 1, meaning only connection rejections are reported. Level 2 additionally reports all successful connections and disconnects. Level 4 enables messages from the SECURITY extension, if present, including generation and revocation of authorizations and violations of the security policy. Level 0 turns off the audit trail. Audit lines are sent as standard error output.

#### ***-auth authorization-file***

specifies a file which contains a collection of authorization records used to authenticate access. See also the [\*xdm\(1\)\*](#) and [\*Xsecurity\(7\)\*](#) manual pages.

#### ***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.

#### ***-bs***

disables backing store support on all screens.

**-br**

sets the default root window to solid black instead of the standard root weave pattern.

**-c**

turns off key-click.

**c volume**

sets key-click volume (allowable range: 0-100).

**-cc class**

sets the visual class for the root window of color screens. The class numbers are as specified in the X protocol. Not obeyed by all servers.

**-co filename**

sets name of RGB color database. The default is */tmp/Xorg-KEM/lib/X11/rgb*.

**-core**

causes the server to generate a core dump on fatal errors.

**-deferglyphs whichfonts**

specifies the types of fonts for which the server should attempt to use deferred glyph loading. *whichfonts* can be all (all fonts), none (no fonts), or 16 (16 bit fonts only).

**-dpi resolution**

sets the resolution for all screens, in dots per inch. To be used when the server cannot determine the screen size(s) from the hardware.

**dpms**

enables DPMS (display power management services), where supported. The default state is platform and configuration specific.

**-dpms**

disables DPMS (display power management services). The default state is platform and configuration specific.

**-f volume**

sets feep (bell) volume (allowable range: 0-100).

**-fc cursorFont**

sets default cursor font.

**-fn font**

sets the default font.

**-fp *fontPath***

sets the search path for fonts. This path is a comma-separated list of directories which the X Server searches for font databases. See the FONTS section of this manual page for more information and the default list.

**-help**

prints a usage message.

**-l**

causes all remaining command line arguments to be ignored.

**-maxbigreqsize *size***

sets the maximum big request to *size* MB.

**-nolisten *trans-type***

disables a transport type. For example, TCP/IP connections can be disabled with **-nolisten tcp**. This option may be issued multiple times to disable listening to different transport types.

**-noreset**

prevents a server reset when the last client connection is closed. This overrides a previous **-terminate** command line option.

**-p *minutes***

sets screen-saver pattern cycle time in minutes.

**-pn**

permits the server to continue running if it fails to establish all of its well-known sockets (connection points for clients), but establishes at least one. This option is set by default.

**-nopn**

causes the server to exit if it fails to establish all of its well-known sockets (connection points for clients).

**-r**

turns off auto-repeat.

**r**

turns on auto-repeat.

**-s *minutes***

sets screen-saver timeout time in minutes.

**-su**

disables save under support on all screens.



**-t *number***

sets pointer acceleration threshold in pixels (i.e. after how many pixels pointer acceleration should take effect).

**-terminate**

causes the server to terminate at server reset, instead of continuing to run. This overrides a previous **-noreset** command line option.

**-to *seconds***

sets default connection timeout in seconds.

**-tst**

disables all testing extensions (e.g., XTEST, XTrap, XTestExtension1, RECORD).

**ttyxx**

ignored, for servers started the ancient way (from init).

**v**

sets video-off screen-saver preference.

**-v**

sets video-on screen-saver preference.

**-wm**

forces the default backing-store of all windows to be WhenMapped. This is a backdoor way of getting backing-store to apply to all windows. Although all mapped windows will have backing store, the backing store attribute value reported by the server for a window will be the last value established by a client. If it has never been set by a client, the server will report the default value, NotUseful. This behavior is required by the X protocol, which allows the server to exceed the client's backing store expectations but does not provide a way to tell the client that it is doing so.

**-x *extension***

loads the specified extension at init. This is a no-op for most implementations.

**[+*-*]xinerama**

enables(+) or disables(-) the XINERAMA extension. The default state is platform and configuration specific.

## 7.1. Server Dependent Parameters

Some X Servers accept the following options:

**-ld *kilobytes***

sets the data space limit of the server to the specified number of kilobytes. A value of zero makes the data size as large as possible. The default value of -1 leaves the data space limit unchanged.

**-lf *files***

sets the number-of-open-files limit of the server to the specified number. A value of zero makes the limit as large as possible. The default value of -1 leaves the limit unchanged.

**-ls *kilobytes***

sets the stack space limit of the server to the specified number of kilobytes. A value of zero makes the stack size as large as possible. The default value of -1 leaves the stack space limit unchanged.

**-logo**

turns on the X Window System logo display in the screen-saver. There is currently no way to change this from a client.

**nologo**

turns off the X Window System logo display in the screen-saver. There is currently no way to change this from a client.

**-render**

**default|mono|gray|color** sets the color allocation policy that will be used by the render extension.

*default*

selects the default policy defined for the display depth of the X Server.

*mono*

don't use any color cell.

*gray*

use a gray map of 13 color cells for the X render extension.

*color*

use a color cube of at most 4\*4\*4 colors (that is 64 color cells).

**-dumbSched**

disables smart scheduling on platforms that support the smart scheduler.

**-schedInterval *interval***

sets the smart scheduler's scheduling interval to *interval* milliseconds.

## 7.2. Xdmcp Parameters

X Servers that support XDMCP have the following options. See the X Display Manager Control Protocol specification for more information.

**-query *hostname***

enables XDMCP and sends Query packets to the specified *hostname*.

**-broadcast**

enable XDMCP and broadcasts BroadcastQuery packets to the network. The first responding display manager will be chosen for the session.

**-multicast [*address* [*hop count*]]**

Enable XDMCP and multicast BroadcastQuery packets to the network. The first responding display manager is chosen for the session. If an address is specified, the multicast is sent to that address. If no address is specified, the multicast is sent to the default XDMCP IPv6 multicast group. If a hop count is specified, it is used as the maximum hop count for the multicast. If no hop count is specified, the multicast is set to a maximum of 1 hop, to prevent the multicast from being routed beyond the local network.

**-indirect *hostname***

enables XDMCP and send IndirectQuery packets to the specified *hostname*.

**-port *port-number***

uses the specified *port-number* for XDMCP packets, instead of the default. This option must be specified before any -query, -broadcast, -multicast, or -indirect options.

**-from *local-address***

specifies the local address to connect from (useful if the connecting host has multiple network interfaces). The *local-address* may be expressed in any form acceptable to the host platform's [\*gethostbyname\(3\)\*](#) implementation.

**-once**

causes the server to terminate (rather than reset) when the XDMCP session ends.

**-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).

**-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!).

**-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.