ORACLE | xsigo systems

XgOS Remote Boot Guide

Release 3.8.0

Xsigo Systems
70 West Plumeria Drive
San Jose, CA 95134
USA

http://www.xsigo.com
Tel: +1.408.329.5600
Part number: 650-20029-08 Rev A
Published: Oct 2012

EMI Statement, United States of America (Class A)

“NOTE: This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.”

EMI Statement, Canada (Class A)

This Class A digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de la classe A est conforme à la norme NMB-003 du Canada.

EMI Statement, Europe and Australia (Class A)

“Warning - This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.”

EMI Statement, Japan (Class A)

この装置は、情報処理装置等電波障害自主規制協議会（ＶＣＣＩ）の基準に基づくクラスＡ情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

“This is a Class A product based on the standard of the Voluntary Control Council For Interference by Information Technology Equipment (VCCI). If this equipment is used in a domestic environment, radio disturbance may arise. When such trouble occurs, the user may be required to take corrective actions.”

Lithium Battery - Replacement and Disposal

CAUTION!

Danger of explosion if the lithium battery is incorrectly replaced. Replace only with the same or equivalent type recommended by the manufacturer. Dispose of used batteries according to the manufacturer's instructions.

Laser Caution for I/O Cards (CDRH-US)

USE OF CONTROLS OR ADJUSTMENTS OR PERFORMANCE OF PROCEDURES OTHER THAN THOSE SPECIFIED HEREIN MAY RESULT IN HAZARDOUS RADIATION EXPOSURE.

Complies with 21 CFR Chapter 1, Subchapter J, Part 1040.10.

IEC 60825-1: 1993, A1: 1997, A2: 2001; IEC 60825-2: 2000

Replacement Laser Transceiver Modules

For continued compliance with the above laser safety Standards, only approved Class 1 modules from our approved vendors should be installed in the product. Contact Xsigo Customer Support (see Technical Support Contact Information) for approved-vendor contact information.

Power Cord Set Requirements – General

The requirements listed below are applicable to all countries:

The length of the power cord set must be at least 6.00 feet (1.8 m) and a maximum of 9.75 feet (3.0 m).

All power cord sets must be approved by an acceptable accredited agency responsible for evaluation in the country where the power cord set will be used.

The power cord set must have a minimum current capacity of 13A and a nominal voltage rating of 125 or 250 V ac~, as required by each country's power system.

The appliance coupler on the power cord must meet the mechanical configuration of an EN 60320 / IEC 60320 Standard Sheet C20 connector, which is the connector on the Fabric Director. The C20 connector supports a C19 plug as the mating part on the power cord that connects to the Fabric Director.

Power Cord Set Requirements – Specifics By Country

United States (UL), Canada (CSA)

The flexible power cord set must be UL Listed and CSA Certified, minimum Type SVT or equivalent, minimum No. 18 AWG, with 3-conductors that includes a ground conductor. The wall plug must be a three-pin grounding type, such as a NEMA Type 5-15P (rated 15A, 120V) or Type 6-15P (rated 15A, 250V).

Europe (Austria (OVE), Belgium (CEBEC), Denmark (DEMKO), Finland (SETI), France (UTE), Germany (VDE), Italy (IMQ), Netherlands (KEMA), Norway (NEMKO), Sweden (SEMKO), Switzerland (SEV), U.K. (BSI/ASTA)

The flexible power cord set must be <HAR> Type H03VV-F, 3-conductor, minimum 0.75mm^2 conductor size. Power cord set fittings, particularly the wall plug, must bear the certification mark of the agency responsible for evaluation in the country where it is being used, with examples listed above.

Australia (DFT/SAA)

Cord is as described under “Japan (PSE)” immediately below. Pins in the power plug must be with the sheathed, insulated type, in accordance with AS/NZS 3112:2000.

Japan (PSE)

The appliance coupler, flexible cord, and wall plug must bear a “PSE” Mark in accordance with the Japanese Denan Law. The flexible cord must be Type VCT or VCTF, 3-conductor, 0.75 mm^2 conductor size. The wall plug must be a grounding type with a Japanese Industrial Standard C8303 (15A, 125V) configuration.

Software Compliance – GPL (GPL v2) Licenses and Notices

Xsigo Systems, a wholly owned subsidiary of Oracle, uses certain elements of GNU Public License (GPLv2) code. Under the conditions of the GPL licensing agreement, you are entitled to request a copy of the open source/freeware code. For questions about Xsigo’s use of the GPL code, or to request a copy of the code, you can contact Xsigo by completing the web form at http://pages.xsigo.com/compliance.html. Afterward, Xsigo will contact you to assist you with your request.

Xsigo Systems, eine ganz besessene Tochtergesellschaft von Oracle, gewisse Elemente des GNU Public License (GPLv2) Code. Unter den Bedingungen von der GPL Lizenzvertrag werden Sie berechtigt, eine Kopie der offenen Quelle/ Freewarecodes zu erbitten. Für Fragen um den Gebrauch von Xsigo des GPL Codes oder eine Kopie des Codes zu erbitten, können Sie Xsigo durch Vollenden der Gewebesform an http://pages.xsigo.com/compliance.html kontaktieren. Nachher wird Xsigo Sie kontaktieren, Ihnen mit Ihrer Bitte zu helfen.

Xsigo Systens, une filiale entièrement possédée d'Oracle, utilisent de certains éléments de GNU Public License (GPLv2) le code. Sous les conditions du GPL autorise l'accord, vous êtes autorisé à demander une copie du code de code source libre/graticiel. Pour les questions de l'usage de Xsigo du code de GPL, ou demander une copie du code, vous pouvez contacter Xsigo en complétant la forme Web à http://pages.xsigo.com/compliance.html. Après, Xsigo vous contactera pour vous aider avec votre demande.

Preface

Purpose

This guide describes how to configure Oracle's Xsigo Fabric Director, your storage devices and your host server to support booting from a remote disk.

Audience

This guide is intended for data center network administrators and system administrators. It assumes that its readers have knowledge and familiarity with common configuration and management tasks related to administering a data center. Although this guide does present some conceptual material about topics and technologies, it is not intended as a complete and exhaustive reference on those topics.

Conventions

Table 1 shows the typographical conventions used in this guide.

Table 1 Syntax Usage

Convention	Description	Example
courier bold	Commands and keywords that must be entered exactly as shown. It also highlights significant lines in the screen output display.	**show vnic**
courier plain	Actual display output that has been copied from the device. Also used for variable names shown in command syntax.	resourceUnavailable
" "	Quotes reference specific fields taken from the screen display on the device.	See the "state" field.
< >	Angle brackets indicate variables for user input. Replace the angle brackets and variable name with information that is indicative of your setup.	**add vnic** <vnic-name>.<server-profile> <slot>/<port>
{}	Curly braces indicate a choice of required keywords or variables. You must enter at least one of the enclosed parameters.	**set vnic** {* | <vnic-name>}
[ ]	Square brackets indicate a choice of optional keywords or variables.	**show system version** [**-all**]
|	A pipe operator indicates a choice. You can enter one of the parameters on either side of the pipe.	**set vnic** {* | <vnic-name>}

Related Documentation

This document is one part of the Xsigo Systems documentation set. Table 2 shows the other documents in the Fabric Director documentation set.

Table 2 Related Documentation for the Xsigo Systems Fabric Director

Document	Part Number	Revision Level and Date
Fabric Manager User Guide	650-30005-02	Rev A 10/2012
Fabric Director Quick Install Guide	650-20022-04	Rev A 10/2012
Fabric Director Hardware and Host Drivers Installation Guide	650-30008-03	Rev A 10/2012
Fabric Accelerator Quick Start Guide	650-20085-03	Rev A 10/2012
Fabric Performance Manager User Guide	650-20082-02	Rev A 10/2012
XgOS Software Upgrade Guide	650-20028-06	Rev A 10/2012
XgOS Command-Line Interface User Guide	650-30007-03	Rev A 10/2012
XgOS vNIC Switching Configuration Guide	650-20052-02	Rev A 10/2012
Installing Host Drivers on Windows 2008 Servers	650-20081-02	Rev A 10/2012
Hyper-V Setup Guide	650-20040-02	Rev A 10/2012
SAN Install for Windows 2008 Servers	650-20078-03	Rev A 10/2012

Release notes are also available with each major hardware or software release.

Revision History

Table 3 shows the revision history for this document.

Table 3 Revision History

Document Title	Document Number	Revision Level and Date
XgOS Remote Boot Guide	650-20029-08	Rev A 10/2012
XgOS Remote Boot Guide	650-20029-07	Rev A 05/2012
XgOS Remote Boot Guide	650-20029-06	Rev A 12/2011
XgOS Remote Boot Guide	650-20029-05	Rev A 11/2009
XgOS Remote Boot Guide	650-20029-04	Rev A 08/2009
XgOS Remote Boot Guide	650-20029-03	Rev A 05/2009
XgOS Remote Boot Guide	650-20029-02	Rev A 11/2008

Technical Support Contact Information

Xsigo Customer Support Services is willing to help solve any reported issues 24 hours a day, 7 days a week, 365 days a year. The Xsigo Technical Assistance Center (TAC) is open 9:00 a.m. to 6:00 p.m. PST Monday through Friday. If you need assistance, you can contact the Xsigo Technical Assistance Center (TAC) in any of the following ways:

- Email

  You can send an email to Xsigo at *support@xsigo.com* and we will respond within 24 hours (Monday through Friday).

- Web Access

  You can create a Service Request through the Support Web interface (http://www.xsigo.com/support) and we will respond within 24 hours (Monday through Friday). If you do not have a log-in we will be more than happy to provide you with access to create, view, update and close Service Requests. You can also open RMA cases through the Web.

- Phone Contact

  In the event that you are in need of a faster response for any reason, Xsigo provides response to all phone calls in a maximum of 30 minutes (24 hours a day, 7 days a week, 365 days a year).

  — You can reach us through the Xsigo switchboard by dialing +1 408-329-5600 and selecting option “2”

  — You can reach us through a direct line, by dialing +1 408-736-3013 (24 hours a day, 7 days a week).

  — For our US customers, you can call us through our toll-free number by dialing 866-974-4647

Contents

Remote Booting Overview 1

This chapter introduces the various ways to implement remote server booting when using Oracle Xsigo Fabric Director to manage your server I/O. It contains the following sections:

Remote Boot Sequence

All computers boot from boot devices, which might include a local SATA disk, a CD-ROM, a USB floppy device, or an Ethernet controller. Xsigo has implemented a ROM BIOS extension for its HCA cards. This extension, called XgBoot, enables you to use Xsigo virtual I/O resources as boot devices for the server. You configure the system BIOS to include XgBoot in the boot order and configure Oracle's Xsigo Fabric Director to make certain virtual resources bootable.

A general description of the boot process follows:

1. On power-up, the server's BIOS performs basic hardware initialization and a power-on self-test (POST).
2. BIOS reads the boot sequence, where the XgBoot for the Xsigo HCA should be the first bootable device in the list, as shown in Figure 1.

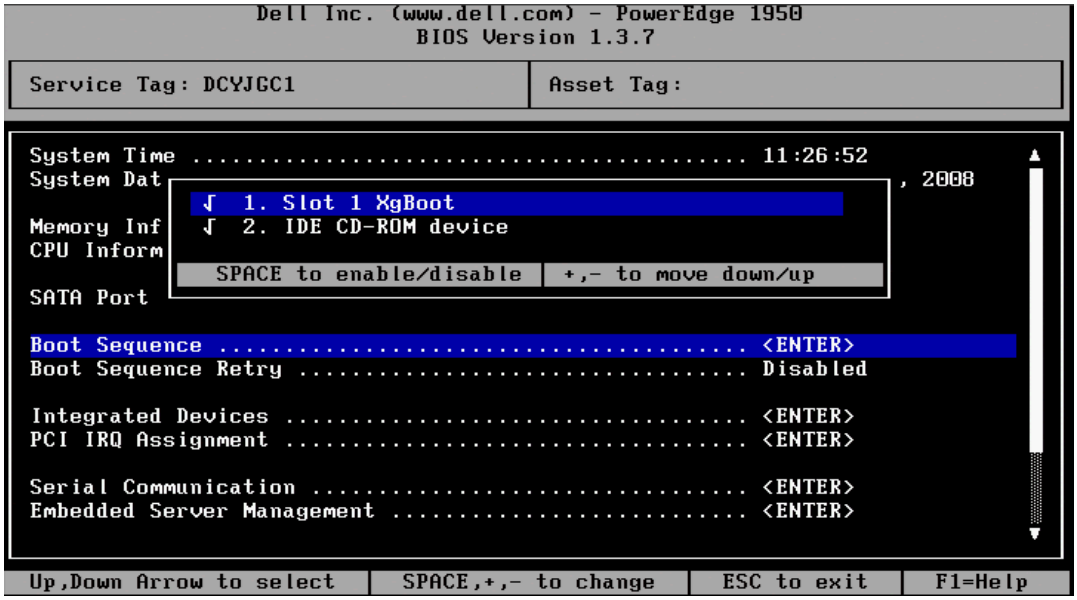


Figure 1 Dell 1950 BIOS, Slot 1 XgBoot HCA

3. The host server establishes a connection to the Fabric Director, where the Fabric Director determines the I/O resource to use and how to set up the communication path from the host server to the boot volume.
4. XgBoot interrogates the boot volume to determine if it is present and bootable. If it is bootable, the Xsigo HCA functions as a hard-disk controller. BIOS begins to treat the Xsigo HCA as the controller for the remote hard disk.

5. The OS loader is installed, which is specific to each OS type:
   - For Linux, the loader is GRand Unified Bootloader (GRUB). This loader resides in the boot sector of a bootable disk. The software responsible for loading the GRUB loader is held in the Option ROM of the HCA and runs in the context of BIOS.
   - For Windows, the loader is NTLoader and is loaded by the same option ROM in the HCA.

   When the loader runs, it typically reads a configuration file from the disk and allows the user to boot the OS in a number of configurations. For GRUB, this file is at **`/boot/grub/grub.conf`**.

   For Linux, GRUB will load the kernel and the initial RAM disk (initrd for Red Hat 4.x and 5.x hosts, and initramfs for Red Hat 6.x hosts)) into memory and begin running the kernel. The root file system (rootfs) given to the kernel is the initrd/initramfs. The kernel runs a program in the initrd/initramfs in the location /init. Typically this program is a shell script that loads the kernel modules and mounts the real root file system. See Xsigo initrd/initramfs on page 5 for more details.
6. The data (Linux kernel or Windows) is sent from the hard disk and loaded into memory. It begins to work and loads all the necessary drivers into the host server.

   Note the boot messages in Figure 2 that are issued as a server connects to a WWN SAN storage device.

```
XgBoot Version 1.3 Built: Fri Dec 14 12:47:53 PST 2007
HCA FW version: 5.2.0
HCA Node Guid: 0x2c90200221d68
Bringing up port 1..
Port 1 bringup successful, LID: 30, SM-LID: 1
Bringing up port 2..
Port 2 bringup FAILED
XSMP session to GUID 0x13970201000201, LID: 1 Successful
IOP connections Successful  GUID: 0x139703010003ca, LID: 6
Number of bootable targets (chassis): 1
WWN: 50:06:01:60:3a:20:1e:83, LUN: 0
```

Figure 2 Typical SAN Bootup

You should also look for the message “XgBoot detected PnP BIOS” that indicates the Xsigo option ROM is installed on the HCA and the proper device-failover mechanisms are supported. If there are two Xsigo HCAs in the system with two option ROMs, two “XgBoot detected PnP BIOS” entries will appear. Thereafter, the system boots up.

General Configuration Process

Whatever remote booting process you select, you will use the following high-level steps:

1. Set up the boot volume.

   You need the boot image available on the network.
2. Configure bootable I/O resources on the Fabric Director.

   For SAN Booting, this is a vHBA configured as bootable. For PXE boot, you use a bootable vNIC. Either resource must be assigned to a server profile that is bound to the server that will boot remotely.
3. Configure the server.

   HCAs on the server must have compatible firmware and their option ROM must be enabled. Refer to “Firmware and Option ROM Levels” on page 153 for instructions on updating firmware.

4. Connect the server-profile on the Fabric Director to the server.
5. Set the server profile to either SAN Boot (**sanboot=**) or iSCSI Bootable (**iscsiboot=**).

Chapters about different kinds of remote booting contain the specific instructions to accomplish this configuration.

Connecting the Server Profile to the Server

When a server boots locally, the Fabric Director becomes aware of its presence on one of its InfiniBand links. When a server cannot boot, the Fabric Director becomes aware of *something* on the link, but has no host name or other information about the server because no communication is possible. Because of this limitation, assigning a server profile with the remote booting configuration to a server works differently from the usual assignment.

To assign a server profile to a server that boots remotely, you need the GUID of the HCA port over which it will boot. You can get this GUID in the following ways:

- When first installing the server, note the HCA GUID ID on the HCA label. Adding one (1) to this number gives you the port GUID for the first HCA port. Adding two (2) to this number gives you the port GUID for the second HCA port.
- After completing the remote-booting configuration, power up the server. Wait until you see the power-up lights both on the server and on the Fabric Director port where the server connects. Then, at an Fabric Director CLI prompt, issue the following command:

  ```
  show physical-server
  name     guid         descr port             os     version  server-profile
  ------------------------------------------------------------------------------
  unknown                      2c90200204cbe
  ```

  The number listed under the port heading is the port GUID for the server.

After you have the port GUID, you can use it in assigning the server profile, as shown:

```
set server-profile <remote-boot-profile> connect 2c90200204cbe
set server-profile <remote-boot-profile> san-boot 2c90200204cbe
set server-profile esx50 iscsi-boot <vnic> -target-iqn=<IQN> -lun=<iscsi-LUN>
```

Types of Remote Booting

The remainder of this book provides specific information and instructions for configuring different types of remote booting.

Remote Booting for ESX Server

The following chapters explain remote booting remote boot functionality for ESX servers:

- Chapter 4, “ESX Server SAN Boot.”
- Chapter 6, “ESX Server iSCSI Boot.”

Remote Booting for Linux Servers

The following chapters explain remote booting for Linux servers:

- Chapter 3, “Linux Server SAN Boot.”
- Chapter 5, “Linux Server iSCSI Boot.”
- Chapter 8, “PXE Boot.”

Remote Booting for Windows Servers

For instructions about SAN Booting Windows servers, see Chapter 7, “Windows Server SAN Boot.”

Xsigo initrd/initramfs 5

When you configure a Linux server for remote booting, you provide it with an initrd/initramfs somewhere on the network. An initial RAM disk (initrd/initramfs) is a RAM-based file system provided to the kernel at boot time. The initrd/initramfs's purpose is to mount the root file system. The format of the initrd/initramfs is typically a compressed Copy Input Output (CPIO) archive but is dependent on operating system version.

In Red Hat Enterprise Linux 6.0 and later hosts, the initrd is actually an initramfs (init RAM file system). The initrd and initramfs are conceptually and functionally the same. However, you should be aware of the terminology change depending on which version of Linux OS will be SAN booted on your host.

This chapter describes the Xsigo initrd and explains how to configure its use. It contains the following sections:

What the initrd Does

The Xsigo initrd is a customized initrd for:

- Linux SAN Boot
- PXE boot
- iSCSI Boot

The initrd loads the virtual I/O drivers and mounts the root file system using a vHBA or a vNIC.

The initrd contains a bash script called "init" plus Aikido scripts that perform higher level functions. The Aikido interpreter is supplied as part of the initrd in order to run the high-level scripts. The init script performs the following duties:

1. Makes device nodes (/dev/tty, etc)
2. Loads kernel modules and Xsigo drivers
3. Waits for the Xsigo drivers to settle

   Settling means to wait for the IB link to come up, to wait for vNICs and vHBAs to appear.
4. Mounts the root file system (/sbin/init)

   Information on where to mount the root file system can be obtained, one at a time, from the following locations:

   — From kernel arguments (see Control the initrd through Kernel Command-Line Arguments on page 10)

   — From /proc/driver/xsvhba/san-info for SAN and PXE boot
5. If all else fails, drops to a text-based menu system that can be used for troubleshooting. See Boot Menu Troubleshooting on page 6.

Using the Xsigo initrd

The Xsigo initrd can be used to provide diskless operation of a server using a Xsigo vNIC or from a SAN disk using a vHBA. The file is a standard format Linux initrd that can be supplied by the PXE server or GRand Unified Bootloader (GRUB) along with a Linux kernel. When the kernel boots with the initrd, the Xsigo drivers will be loaded and the Linux root file system can be mounted from a remote SAN disk.

In order to specify which server to mount, the initrd reads the kernel command line and looks for Xsigo-specific options. Typically, the Xsigo configuration comes from the Fabric Director. These options are used to specify which vNIC to use as the network interface and where to mount the root file system from.

For SAN Boot the initrd can read the kernel command line, or it can use information supplied from the Xsigo Fabric Director.

To see the contents of an initrd:

```
zcat xsigo-initrd-<kernel>-<version>-i386.img | cpio -t
```

To extract the contents into a directory:

```
cd destdir
zcat xsigo-initrd-<kernel>-<version>-i386.img | cpio -uid
```

The Xsigo initrd does not include the utilities and other files that you need if you are using multipathing. If you use multipathing, you must add any multipathing requirements to the initrd before deploying it. For instructions about modifying the initrd for multipathing, see Modifying the initrd for Multipathing with RHEL 5.x Hosts on page 19.

Also see Installing the SAN Volume (Linux Servers) on page 9.

Boot Menu Troubleshooting

If the root file system cannot be mounted automatically, the initrd enters a mode called **bootmenu**. This mode is a screen-based menu used to troubleshoot problems and try alternatives.

Running bootmenu

You can force the boot menu to be run by adding the kernel argument **bootmenu** to command line.

The menu looks like this:

```
                        Xsigo Systems Virtual Boot (1.0.0)
-- Boot configuration ----------------------------------------------------
                              [No boot configuration]

-- Main menu -------------------------------------------------------------

d. Boot from SAN Disk n. Boot from network  t. Terminal settings
r. Refresh screen     c. Reset terminal     s. Spawn shell
E. Exit to shell      R. Reboot             P. Power off

Selection:
```

The top part of the screen display allows you to enter and display the current boot configuration. Press a key to invoke a menu option (no return is required). When entering information, TAB moves to the next field. Pressing ENTER moves to the next field and terminates the entry at the last field. You can also use the UP and DOWN arrow keys to move between fields.

Providing SAN Boot Configuration Through bootmenu

The **d** option allows you to enter SAN Boot information:

```
                        Xsigo Systems Virtual Boot (1.0.0)
-- Boot configuration ----------------------------------------------------

                              [No boot configuration]

-- SAN Boot menu ---------------------------------------------------------

v. Show VHBAs                    l. Enter LVM configuration
d. Enter SAN disk configuration  b. Boot operating system
r. Refresh screen                q. Return to previous menu

Selection:
```

Providing SAN LVM Information Through bootmenu

To enter LVM information, use the **l** option:

```
                        Xsigo Systems Virtual Boot (1.0.0)
-- SAN LVM configuration -------------------------------------------------

     Group          VolGroup00_______
     Volume         LogVol00_________
     Mount opts     _________________


-- SAN Boot menu ---------------------------------------------------------

v. Show VHBAs                    l. Enter LVM configuration
d. Enter SAN disk configuration  b. Boot operating system
r. Refresh screen                q. Return to previous menu

Selection:
```

Providing SAN Disk Information Through bootmenu

For SAN disk information (target and LUN), use the **d** option on the second-level menu:

```
                        Xsigo Systems Virtual Boot (1.0.0)
-- SAN disk configuration ------------------------------------------------

     VHBA           vhba1_____________
     Target         3_________________
     WWPN           00:33:13:0a:34:54
     LUN            42________________
     Mount opts     __________________

-- SAN Boot menu ---------------------------------------------------------

v. Show VHBAs                    l. Enter LVM configuration
d. Enter SAN disk configuration  b. Boot operating system
r. Refresh screen                q. Return to previous menu
Selection:
```

Linux Server SAN Boot 9

SAN Boot allows you to boot a Linux host server from a SAN volume accessed through a vHBA. The remote disk to boot from is identified by a target World Wide Port Name (WWPN) and Logical Unit Number (LUN) on a storage disk array device. SAN Boot allows Linux hosts with an InfiniBand HCA to connect to Oracle's Xsigo Fabric Director, and receive their boot information from a disk connected to the host by a Xsigo vHBA.

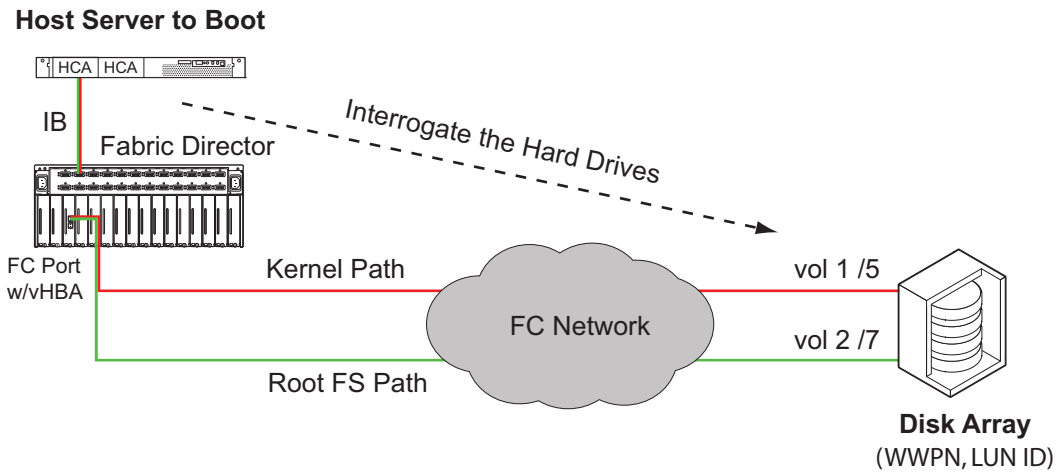


Figure 1 Linux SAN Boot Topology

This chapter explains how to set up SAN Boot for a Linux server using virtual HBAs. It contains the following sections:

- Control the initrd Through Fabric Director Commands
- Control the initrd through Kernel Command-Line Arguments
- CLI Support for SAN Boot
- Best Practices and Caveats
- Installing Linux on the SAN
- Modifying the initrd for Multipathing with RHEL 5.x Hosts
- SAN Boot for Red Hat Enterprise Linux 6.1 and Later Hosts

Control the initrd Through Fabric Director Commands

You can send instructions to the initrd through the Fabric Director, by using the information model in the Fabric Director (entered via the command-line interface), which communicates with the host's vHBA driver (/proc/driver/vhba/san-info). See CLI Support for SAN Boot on page 11.

For example:

```
set server-profile myserver san-boot myvhba 1:2:3:4:5:6:7:8 42
```

If you build a SAN Boot configuration for a **server-profile** on an Fabric Director, the vHBA driver will create a file called /proc/driver/xsvhba/san-info containing information similar to the kernel arguments. The initrd can read this file and determine the root file system location from it.

Control the initrd through Kernel Command-Line Arguments

Sometimes users have an existing system for SAN Boot and prefer not to use the Xsigo Fabric Director for SAN Boot configuration. Instead, kernel arguments in **grub.conf** are used to specify where devices should be mounted. The following information overrides the information provided by the vHBA driver.

Multiple devices are supported, where multiple **sanboot=** arguments are included on the command line. Some kernel arguments control aspects of the settle facility in the initrd. Settling is used to wait for Xsigo's drivers to initialize.

The options are:

sanwait[=<secs>]

This option waits for vHBAs for <secs> seconds. A value of 0 means no wait. If no value is specified, then the default is used (30 seconds).

netwait[=<secs>]

This option waits for vNIC for <secs> seconds. A value of 0 means no wait. If no value is specified, then the default is used (30 seconds).

If both of these arguments are not present, then both are deemed to be enabled by default (wait for net and SAN).

When booting from SAN, the default behavior is to wait for both vNICs and vHBAs to settle. Because there may be no vNICs present in the server profile, time is wasted by waiting for them. You can specify that the initrd waits for only the vHBA by adding **sanwait** to the kernel command line.

You can have multiple arguments on the kernel command line.

kernel Command-Line Arguments for Troubleshooting

bootdebug

Turns on debugging detail in the initrd.

bootmenu

Runs the bootmenu instead of automatically booting. See Boot Menu Troubleshooting on page 6 for more details.

emergency

Runs bash after mounting the root file system. Do not run the system init scripts.

`init=/bin/bash`

Runs bash after mounting the root file system.

`single`

Boots to single-user mode instead of the default run-level configured in /etc/inittab.

CLI Support for SAN Boot

Roles

SAN Boot operates in three different roles (phases):

- Load—Loads the operating system and initrd (in the case of Linux) from a SAN disk. The kernel and initrd are located and installed into memory.
- Mount—Mounts the root file system on a SAN disk. It may or may not be the same disk as the operating system was loaded from. The following mount types are available: logical volume management (LVM) and direct. In the mount role, initrd starts and mounts the device file system corresponding to the vHBA specified. The target disk and its root file system inside the OS are mounted.
- Loadmount—Both roles are performed, and all files come from the same location. That is, the same target disk contains the boot image and the root file system. This common use case loads the kernel and initrd into memory, where initrd starts and mounts the device file system corresponding to the vHBA specified.

Syntax for SAN Boot

```
set server-profile <name> san-boot [<vhba>|none] <wwpn> <lun>
set server-profile <name> san-boot <vhba> <wwpn> <lun>
show server-profile <server> san-boot
```

SAN Boot Parameter Descriptions

`set server-profile` `<name>`—Identifies a named server profile to boot from.

`san-boot` `<vhba>|none` — Creates a boot object and associates it with a named vHBA. The SAN Boot object can use only the vHBAs that are available to the server profile. You must have a previously created vHBA and scanned for available LUNs.

`<wwpn>`—Available target WWPN, as used for a physical hard disk.

`<lun>`—Available LUN ID on the target, as used for a logical hard disk.

none—Removes the SAN Boot association from a server profile.

Note

For more information on LUNs and targets, refer to the *XgOS Command-Line Interface User Guide*.

show server-profile <server> **san-boot**—Displays SAN Boot information for a server profile.

Example: Mount Type

To load using the Fabric Director's information model:

```
set server-profile myserver san-boot myvhba 1:2:3:4:5:6:7:8 42
```

Example: show server-profile san-boot

To display the SAN Boot facilities on an Fabric Director:

```
show server-profile myserver san-boot
server    role     vhba   mnt-type  lvm-grp      lvm-vol   dev            mnt-opts   disks
-------------------------------------------------------------------------------
myserver loadmount vhba1                                   1:2:3:4:5:6:7:8(42/LM)
1 record displayed
```

Note the following information:

- The server name is myserver.
- The vHBA used for SAN Booting is vhba1.
- The disk to be mounted is LUN 42 on 1:2:3:4:5:6:7:8

Best Practices and Caveats

SAN Boot is complex. This section presents some issues to keep in mind.

Upgrading The Xsigo Drivers On A SAN-Booted Server

The environment on a SAN Booted server is no different from a standard server with respect to the Xsigo drivers. On a SAN-booted server, the drivers are installed or upgraded by using **rpm -Uvh** <driver-name>.

Reduce the Time SAN Boot Takes

When you are using SAN boot for your Linux server, the boot process might pause for several minutes at the **starting udev** message. You can reduce the duration of this pause by setting the **udevtimeout** kernel parameter to a lower value.

To reset the **udevtimeout** parameter when using GRUB, modify the entry as follows:

```
// initial entry
kernel /vmlinuz-2.6.18-53.el5 ro root=/dev/VolGroup00/LogVol00 rhgb quiet

// modified entry
kernel /vmlinuz-2.6.18-53.el5 ro root=/dev/VolGroup00/LogVol00 rhgb quiet udevtimeout=5
```

Installing Linux on the SAN

You can use any of the following approaches to install the SAN volume:

- SAN Installation Using Linux Installers
- Example 1: CD Installation for 5.0.x Host Drivers on RHEL 5 Hosts
- Example 2: RHEL 5 SAN Boot Installation and Configuration through xg-insert-dd for PXE Boot
- Example 3: Installing over Multipath vHBAs Using XgBoot

SAN Installation Using Linux Installers

The Anaconda installer is supported for Red Hat servers. Xsigo does not modify the installer. Instead Xsigo uses the version supplied by the OS provider and adds in Xsigo's driver disks. A Xsigo driver disk is created for a specific kernel for a specific OS.

The **`xsigo-boot`** tar archive contains img files and scripts used to support booting the Xsigo drivers.

Table 1 Files and Scripts Used for the Xsigo Initrd

File Name	Purpose
`xsigo-boot-``<kver>-<version>-<arch>`**`.tar`**	Boot file compiled for a specific kernel version.
`xsigo-initrd-``<kver>-<version>-<arch>`**`.img`**	A Linux initrd with Linux that uses the Xsigo drivers. The initrd is used with Red Hat Enterprise Linux distributions earlier that version 6.X
`xsigo-initramfs-``<kver>-<version>-<arch>`**`.img`**	A Linux initramfs with Linux that uses the Xsigo drivers. The initramfs is conceptually similar to an initrd, but the initramfs is used with Red Hat Enterprise Linux distributions equal to or later that version 6.X
`xsigo-rhdd-``<kver>-<version>-<arch>`**`.img`**	A Red Hat Anaconda installer driver disk.
`xg-insert-dd`	For RHEL5, if you want to avoid using a CD to load the driver disk, you can insert the disk into the Anaconda initrd using the script **`xg-insert-dd`**. It creates a new initrd prefixed with the string "xsigo-". The `xg-insert-dd` script is used on Red Hat Enterprise Linux hosts earlier than RHEL 6 Update 1, which is the first RHEL 6 platform supported. (RHEL 6 Update 0 is not supported.)
`xg-insert-dd-ext2`	For RHEL 4, inserts the driver disk into the Anaconda initrd.

You can add the Xsigo driver disk into the installer in the following ways:

1. Boot from a CD-ROM. Create an ISO image that provides the required drivers to the installer during runtime. This is supported on RHEL 4.x and 5.x and earlier hosts, but is not supported on RHEL 6.X and higher hosts.

2. Boot from the initrd itself. For RHEL 4 and 5 servers, Xsigo provides a script that takes a standard initrd (for example the Anaconda initrd from the Red Hat installation disk) and then inserts the Xsigo drivers (as a driver disk) into that initrd. Thereafter, enter the Xsigo Anaconda initrd into the PXE boot system as the initrd. As the kernel boots, it will load the Xsigo drivers. For RHEL 6.x server, Xsigo you use the append command to add the Xsigo drivers to the initrid.

On the Fabric Director, set a vHBA to load using a specific WWPN and LUN ID. The simplest setup is to use the same disk for loading the operating system and mounting the root file system—the loadmount role. For example:

```
set server-profile <name> san-boot <vhba-name> <wwpn> <lun-id>
```

The initrd loads from the mounted SAN Boot object. Once installed, the SAN disk will appear as any other disk in the system.

Example 1: CD Installation for 5.0.x Host Drivers on RHEL 5 Hosts

This example illustrates how to perform the SAN installation from CD-ROM using Linux installers. To complete this procedure, you will do the following general phases:

- create a temporary workspace (a tmp directory)
- get the extracted xsigo-boot image
- use the xsigo-insert-dd script to inject the Xsigo host drivers into the RHEL initrd
- create an ISO image of the initrd with the Xsigo host drivers in it. Afterwards, you can move the new ISO image to the SAN where it can be used to SAN or iSCSI boot a connected host.

To install RHEL5 over a vHBA to SAN storage:

Step 1 Download the corresponding RHEL iso and the xsigo-boot tar package to a server.

Step 2 Create a directory /tmp/remaster for doing the remaster of the RHEL 5 iso to inject the xsigo modules.

```
#mkdir /tmp/remaster
```

Step 3 Create subdirectories in this directory

```
#mkdir /tmp/remaster/iso
#mkdir /tmp/remaster/extracted
#mkdir /tmp/remaster/xsigo
```

Step 4 Change to the directory /tmp/remaster/xsigo and extract the xsigo-boot tar package

```
#cd /tmp/remaster/xsigo
#tar -xvf /builds/drivers/5.0.1.LX3D/redhat/xsigo-boot-2.6.18-238.el5-
5.0.1.LX3D-x86_64.tar
```

Step 5 Mount the ISO to extract the contents

```
#mount -o loop /export/isos/testing/rhel-server-5.6-x86_64-dvd.iso /tmp/
remaster/iso
```

Step 6 Copy the contents to the extracted subdirectory

```
# rsync -av /tmp/remaster/iso/ /tmp/remaster/extracted/.
```

Step 7 Unmount the iso directory

```
# umount /tmp/remaster/iso
```

Step 8 Change directory to “isolinux” in the “extracted” folder

```
# cd /tmp/remaster/extracted/isolinux
```

Step 9 Use the `xg-insert-dd` script to insert the xsigo modules into the initrd.img.

```
#/tmp/remaster/xsigo/xg-insert-dd /tmp/remaster/xsigo/xsigo-rhdd-2.6.18-
238.el5-5.0.1.LX3D-x86_64.img initrd.img
```

This will generate a xsigo-initrd.img file

Step 10 Move the original initrd.img to initrd.img.orig

```
# mv /tmp/remaster/extracted/isolinux/initrd.img /tmp/remaster/
extracted/isolinux/initrd.img.orig
```

Step 11 Move the xsigo-initrd.img to initrd.img

```
# mv /tmp/remaster/extracted/isolinux/xsigo-initrd.img /tmp/remaster/
extracted/isolinux/initrd.img
```

Step 12 Come out of the isolinux directory.

```
# cd /tmp/remaster/extracted
```

Step 13 Now the xsigo modules are loaded in to the initrd.img,use the “mkisofs” to make a iso from this contents.

```
#/usr/bin/mkisofs -R -J -T -o /tmp/xsigo-rhel-server-5.6-x86_64-dvd.iso
-b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot
-boot-load-size 4 -boot-info-table -R -m TRANS.TBL .
```

The remastered ISO will be available in /tmp/xsigo-rhel-server-5.6-x86_64-dvd.iso

Step 14 Burn this ISO and use this for Installing on vHBA LUN or vNIC provided iSCSI LUN.

Example 2: RHEL 5 SAN Boot Installation and Configuration through xg-insert-dd for PXE Boot

For Red Hat to install the Xsigo virtual I/O, you will need to build a custom initrd from the default initrd that Red Hat provides. To support creating the custom initrd, Xsigo provides `xg-insert-dd` which is a script that injects the Xsigo host drivers into the default initrd provided by Red Hat. The `xg-insert-dd` script is required to allow the Red Hat server to recognize the Xsigo devices and attach the correct host drivers. Without the custom initrd, the Xsigo devices, including the bootable vNIC and vHBA, are marked as unknown devices, and eventually get marked “not bootable” and are bypassed while the Red Hat initrd is loading. If the bootable vNIC and vHBA are marked “not bootable,” the server will not be able to SAN boot.

To use the `xg-insert-dd` script, you will need the following software:

- Xsigo boot drivers (the rhdd image)
- Stock initrd (provided by Red Hat on your installation media)
- `xg-insert-dd` script

You will want to copy them all to a temporary working directory to create the custom initrd.

The `xg-insert-dd` script has the following syntax:

`sh xg-insert-dd {-iscsiboot[=boot.tgz]]` `<dd.img> <anaconda>`

where:

- `dd.img` is the Xsigo boot driver that you want to insert into the default initrd
- `anaconda` is the default Red Hat initrd

Note

Disregard the **`-iscsiboot=boot.tgz`** option.

When `xg-insert-dd` is run, it opens the default initrd for editing, places the Xsigo host drivers inside, then repacks the initrd as a custom initrd with the `xsigo-` prefix. When the server uses this custom initrd, the Xsigo drivers will be loaded at boot time.

To install RHEL5 over a vHBA to SAN storage:

Step 1 On the host server, install an HCA containing Xsigo's SAN Boot option ROM.

If the HCA does not have the correct option ROM, see "Firmware and Option ROM Levels" on page 153 for instructions about how to install it.

Note

Many SAN Boot-capable servers exist. However only certain BIOSes support a certain number of HCAs with the Xsigo option ROM. Consult Xsigo Support for the hardware compatibility matrix.

Step 2 Download the appropriate `xsigo-boot-<kernel>.tar` file unique to your distribution and architecture. RHEL5 32-bit is used in this example.

Untar **`xsigo-boot-`**`<kernel>`**`.tar`** file and create a PXE Boot image by using the `xsigo-insert-dd` tool and following the prompts presented while running `xsigo-insert-dd`.

From Linux:

`tar xvf xsigo-boot-``<kernel>`**`.tar`**

Step 3 From the Fabric Director, create a Server Profile for the server performing the RHEL5 install to SAN storage. For the physical connection, use the host's GUID.

You should have the GUID address available. The GUID is commonly displayed on the HCA packaging. The GUID is also displayed during the server's POST, so you can get the GUID during the server's POST.

```
add server-profile pokemon 2c02123a5303
```

For more about assigning a Server Profile to a server that cannot boot yet, see "Connecting the Server Profile to the Server" on page 3.

Step 4 Add the vHBA to the server profile:

```
add vhba vhba0.pokemon 3/1
```

Step 5 Confirm that LUNs are visible to the server-profile by running a rescan and showing target luns. At this point you will need to have already configured the storage side (such as, zoning, disk allocation, and so on).

```
set vhba vhba0.pokemon prescan
show vhba vhba0.pokemon targets
```

Step 6 As an option, remove all physical hard drives from server. This step is not mandatory.

Step 7 Set up the server profile to do SAN Boot. The **set server-profile** <name> **san-boot** provides the server with LUN that the host needs to connect to boot up.

```
set server-profile pokemon san-boot vhba0 50:06:01:60:3A:20:1E:83
```

Step 8 Log in to your virtual terminal and use the initrd found in the TAR ball.

Step 9 Install the IB stack first:

```
rpm -ivh kernel-ib.rpm
```

Step 10 After the `kernel-ib` rpm is successfully installed, install the Xsigo host driver RPM:

```
rpm -ivh xsigo-host-drivers.rpm
```

Step 11 After installation is complete, you will need to copy over the xsigo-initrd-<kernel>-<version>-i386.img file into the SAN storage /mnt/sysinstall/boot directory and modify the GRUP configuration for this initd file.

Then, edit the initrd line to read the following:

```
initrd /xsigo-initrd-<kernel>-<version>-i386.img
```

Step 12 After adding the preceding line to the initrd, write the changes (**:w**) and quit the editor (**:q**).

Step 13 Reboot the server.

Example 3: Installing over Multipath vHBAs Using XgBoot

This example shows how to SAN install a Red Hat Enterprise Linux on the SAN. For this example, RHEL 6u3 is used.

To complete this procedure, you will need the following:

- Two Fabric Directors connected to an RHEL server
- A fully configured and functional PXE server and TFTP server
- A fully configured and functional storage target, which at least one LUN available. The LUN must be large enough to contain the RHEL image on it completely.
- A RHEL 6u3 server with an HCA installed and running the latest version of XgBoot. This requirement allows the server to detect the Fabric Director

To install over multipath images, follow this procedure:

Step 1 If you have not already created a Server Profile, do so now. For illustrative purposes, this procedure uses the Server Profile named `sar`. For example:

```
add server-profile sar
```

Step 2 Repeat the previous step on the second Fabric Director.

Step 3 Add a bootable vNIC from one of the Fabric Directors:

```
admin@lanai[xsigo] add vnic pxevnic.sar 2/1 -boot-capable=true
```

As an option, you can display the vNIC tor

```
admin@lanai[xsigo] show vnic pxevnic.sar
name              state       mac-addr               ipaddr       if        if-
state       ha-state       local-id       type       vlans       qos       flags
-------------------------------------------------------------------------------
```

```
pxevnic.sar       up/up       00:13:97:30:E0:1B                    2/1
up                          0                       none       --        -b-----
```

Step 4 Add two vHBAs (a multipath vHBA) from each Fabric Director and make sure each multipath vHBA points to the same LUN. For example to add the primary vHBA:

```
admin@lanai[xsigo] add vhba vhba1.sar 1/1 -wwn-id=601
```

As an option, you can display the vHBA to verify that it is correctly created:

```
admin@lanai[xsigo] show vhba vhba1.sar target
vhba                    name                  wwnn
wwpn                                         lun-ids
------------------------------------------------------------------------------
vhba1.sar                                  50:0A:09:80:88:2A:39:9F
50:0A:09:81:88:2A:39:9F                      99
1 record displayed
```

Step 5 Set the Server Profile to support SAN Boot functionality.

```
admin@lanai[xsigo] set server-profile sar san-boot vhba1
50:0A:09:81:88:2A:39:9F 99 mount lvm
```

As an option, you can display the Server Profile to verify that it was correctly created.

```
admin@lanai[xsigo] show server-profile sar san-boot
server         role            vhba          mnt-type         lvm-grp
lvm-vol          dev          mnt-opts          disks
------------------------------------------------------------------------------
sar       loadmount     vhba1     lvm
50:0A:09:81:88:2A:39:9F(99/LM)
1 record displayed
```

Step 6 On the second Oracle Fabric Director, repeat the previous step to create the secondary vHBA. For example, to create the second vHBA:

```
admin@maui[xsigo] add vhba vhba2.sar 1/1 -wwn-id=601
```

As an option, you can display the vHBA to verify that it is correctly created.

```
admin@maui[xsigo] show vhba vhba2.sar target
vhba                    name                wwnn
wwpn                                         lun-ids
------------------------------------------------------------------------------
vhba2.sar                                  50:0A:09:80:88:2A:39:9F
50:0A:09:81:88:2A:39:9F                      99
1 record displayed
```

Step 7 Set the Server Profile to support SAN Boot functionality.

```
admin@maui[xsigo] set server-profile sar san-boot vhba2 50:0A:09:81:88:2A:39:9F
                  99 mount lvm

admin@maui[xsigo] show server-profile sar san-boot
server         role             vhba          mnt-type         lvm-grp
lvm-vol          dev          mnt-opts          disks
-------------------------------------------------------------------------------
sar            loadmount        vhba2        lvm

50:0A:09:81:88:2A:39:9F(99/LM)
1 record displayed
```

Step 8 Log in to the PXE server and make sure that the label contains the multipath option (mpath) so that the installer looks at the LUNs from the two vHBAs as a Multipath Disk. If needed, edit the label to make it as follows:

```
label RHEL6U3-505LX
kernel vmlinuz-rhel6u3-x86_64
append initrd=initrd-rhel6u3-x86_64.img,xsigo-rhdd-2.6.32-
279.el6.x86_64-5.0.5.LX-x86_64.img  mpath  network
```

Step 9 Log into the server, and enter the BIOS.

Step 10 In BIOS, change the boot sequence so that XgBoot is at the top of the boot devices list. By doing so, you ensure that XgBoot boots through the bootable vNIC.

Step 11 Boot from the label and continue the installation.

Step 12 When selecting the hard disk on which to install, make sure it shows the LUNs as the multipath disk (for example, either `/dev/mapper/mpath0` or `/dev/dm-0`).

Step 13 Continue with the manual installation and before the reboot install the kernel-ib and Xsigo host drivers RPMs.

Step 14 Reboot the server, and this time it boots from the Multipath LUN by checking “multipath –l” command.

Modifying the initrd for Multipathing with RHEL 5.x Hosts

Xsigo provides a multipath-capable RPM file (multipath.rpm). If your Red Hat servers are running in a multipath environment, you will need to install multipath.rpm. Previously, you were required to edit the initrd to allow multipathing, but this is no longer required. Instead, just make sure to install multipath.rpm. You no longer need to make changes to Device Mapper.

To support multipathing, follow this procedure:

Step 1 Install multipath.rpm before rebooting the server.

Step 2 Make sure the Xsigo devices that the server will be booting from are not in the Blacklist file.

Red Hat configurations black list every device by default. As a result, all devices are ignored. Depending on your network, you might need to comment out the entire Blacklist, or edit the Blacklist to create a custom stanza that causes only some boot devices to be ignored. For information about editing or customizing the Blacklist file, consult the documentation that accompanied your multipath software.

SAN Boot for Red Hat Enterprise Linux 6.1 and Later Hosts

The procedure for configuring SAN Boot on a Red Hat Enterprise Linux host is the same as for Red Hat 5.x hosts, with the exception of how the Xsigo host drivers are included in the RHEL 6.1 initramfs. (The initramfs is conceptually and functionally the same as the initrd in RHEL 5.X distributions and earlier.)

With RHEL 6.1 hosts, the process of adding the Xsigo host drivers to make a modified initrd has been made much simpler through the use of the **append** command. This command allows you to simply add the Red Hat disk drivers to the 6.1 OS image when the server is PXE or SAN Booting. The **append** command is issued on the PXE Boot Server that booting hosts connect to during their PXE or SAN Boot phase.

Using the append Command

The **append** command takes the place of the xg-insert-dd tool that you use with Red Hat 5.x hosts. As a result, the xg-insert-dd tool is not used with RHEL 6.x hosts, but still is required with RHEL 5.x hosts. The **append** command performs the same function as the xg-insert-dd does for RHEL 5, but in a more streamlined way. To inject the Xsigo host drivers into the RHDD, use the **append initrd** option:

```
append initrd=<initramfs-string>,<xsigo-rhdd-image> ksdevice=<device-name>
network
```

where:

- `<initramfs-string>` is the Red Hat OS image and the Xsigo boot driver (RHDD) that you want to insert into the Red Hat OS image separated by a comma only--no blank.
- `<device-name>` specifies the particular interface that you want the installer (for example, eth2) to use for the kickstart process.

To use the **append** command, you will use the base RHEL 6.1 initramfs and the Xsigo RHDD image. For example:

```
append initrd=initrd-rhel6u1-x86_64.img,xsigo-rhdd-2.6.32-
131.0.15.el6.x86_64-3.6.9.LX3-x86_64.img ksdevice=eth2 network
```

Pay close attention to the syntax, especially the comma (,) separator between the RHEL image and the Xsigo RHDD. There should be no blank spaces between the RHEL image and the Xsigo RHDD--only a comma. If spaces exist, or the comma is not present, the command will fail and the Xsigo host drivers will not be appended to the RHEL image. As a result, the server will not SAN Boot.

Example: RHEL 6.1 SAN Boot Installation and Configuration for PXE Boot

For Red Hat to install the Xsigo virtual I/O, you will need to add the Xsigo modules into the default initramfs that Red Hat provides by using the **append initrd** command to append the Xsigo host drivers to the default initramfs. The Xsigo host drivers must be added to the initramfs to allow the Red Hat server to recognize the Xsigo devices and attach the correct host drivers.

If the Xsigo devices bootable vNIC and vHBA are not in the initramfs, the Linux OS temporarily marks them unknown devices, and they eventually are marked “not bootable.” When the Xsigo vNIC and vHBAs are marked not bootable, they are bypassed while the Red Hat initramfs is loading, and the server will not be able to SAN boot from the Xsigo devices.

To use the **append** command, you will need the following software:

- Xsigo boot drivers (the RHDD image)
- Stock initrd (provided by Red Hat on your installation media)

To install RHEL6.x over a vHBA to SAN storage:

Step 1 On the host server, install an HCA containing Xsigo's SAN Boot option ROM.

If the HCA does not have the correct option ROM, see "Firmware and Option ROM Levels" on page 153 for instructions about how to install it.


Note

Many SAN-boot-capable servers exist. However only certain BIOSes support a certain number of HCAs with the Xsigo option ROM. Consult Xsigo Support for the hardware compatibility matrix.

Step 2 Download the appropriate `xsigo-boot-<kernel>.tar` file unique to your distribution and architecture.

Step 3 Create the modified initramfs by running the **`append initrd`** command.

Step 4 From the Fabric Director, create a server profile for the server performing the RHEL6 install to SAN storage. For the physical connection, use the host's HCA GUID.

You can get the server's GUID from the HCA packaging. Or, the GUID is also displayed during the server's power on self-test (POST), so you can watch for it during the server's POST. You should have the server's GUID available for this part of the procedure.

```
add server-profile pokemon 2c02123a5303
```

For more about assigning a server profile to a server that cannot boot yet, see "Connecting the Server Profile to the Server" on page 3.

Step 5 Add a vHBA to the server profile:

```
add vhba vhba0.pokemon 3/1
```

Step 6 Confirm that LUNs are visible to the server profile by running a rescan and showing target LUNs. At this point you will need to have already configured the storage side (such as, zoning, disk allocation, and so on).

```
set vhba vhba0.pokemon rescan
show vhba vhba0.pokemon targets
```

Step 7 As an option, remove all physical hard drives from server. This step is not mandatory.

Step 8 Set up the server profile to do SAN Boot. The **`set server-profile`** `<name>` **`san-boot`** command provides the server with all the information that it needs to connect to SAN storage upon boot up (such as, the vHBA to use, WWPN and LUN to connect to, volume group, logical volume to use, and so on).

```
set server-profile pokemon san-boot vhba0 50:06:01:60:3A:20:1E:83 0
```

Step 9 After installation is complete but before rebooting the server, you will need to install the kernel IB.rpm file:

```
rpm -ivh kernel-ib.rpm
```

Step 10 After the `kernel-ib` rpm is successfully installed, install the Xsigo host driver RPM:

```
rpm -ivh xsigo-host-drivers.rpm
```

Step 11 Reboot the server.

SAN Boot allows you to boot a VMware ESX Server from a SAN volume accessed through a vHBA. The remote disk to boot from is identified by a target World Wide Port Name (WWPN) and Logical Unit Number (LUN) on a storage disk array device.

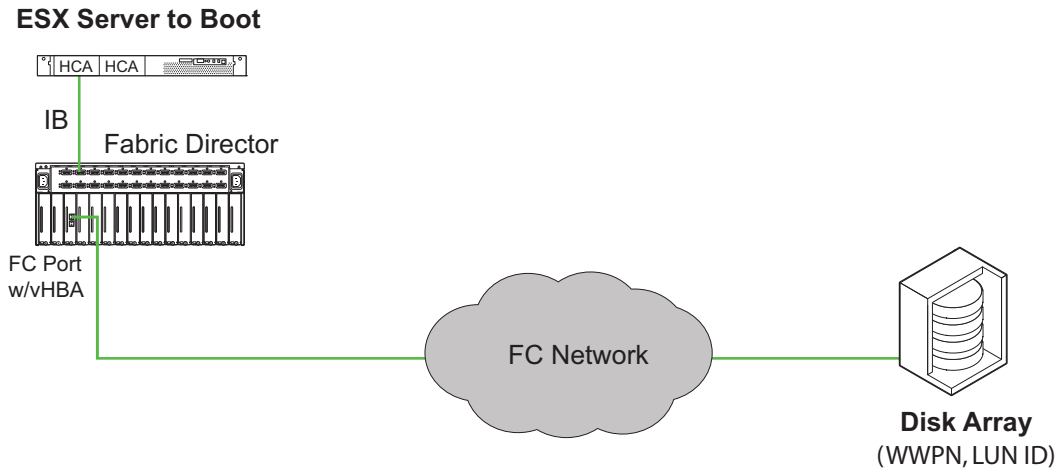


Figure 1 ESX Server SAN Boot Topology

This chapter describes SAN Boot in the context of the following ESX Server versions:

- ESXi Server 5.0 (and updates).
- ESX Server Classic 4.1 (and updates). In the following documentation, ESX Classic 4.1 is referred to as “ESX 4.1.” Some screen caputures might show ESX 4.0, but the procedure is applicable to ESX 4.1.
- ESXi Server 4.1 (and updates). In the following documentation, ESXi 4.1 is referred to as “ESXi 4.1.” Some screen caputures might show ESX 4.0, but the procedure is applicable to ESX 4.1.

This chapter contains the following sections:

Autodeploying ESXi 5.1 Host Drivers

Through auto deploy, ESXi 5.1 hosts can be set up to receive host drivers through a PXE boot server. Auto deployments are useful in large deployments of ESXi 5.1 hosts.

This section documents how to configure auto deploy for Xsigo Host drivers.

Install the Required Software

Perform the following procedure in the steps shown:

Step 1 Download vCenter 5.1

Step 2 Download ESXi 5.1 offline bundle.

Step 3 Install vCenter Server 5.1

Step 4 Install vSphere Client 5.1

Step 5 Install PowerShell 2.0 on vCenter Server

Step 6 Install PowerCLI 5.1 on vCenter Server

Step 7 Install auto deploy (which is located in the vCenter server ISO).

Step 8 Install a TFTP Server on the vCenter Servers

Configure TFTP and vCenter Servers and DHCP

After installing all of the required software, following this procedure to set up the TFTP server, the vCenter server, and the DHCP address allocation.


Note

In this section you will be configuring multiple servers so that IP addresses can be allocated to host interfaces. Make sure that the DHCP Server, TFTP Server, and vCenter Server are all on the same subnet to avoid connectivity problems.

Configure the TFTP Server

Step 1 Start the TFTP server by clicking *File-> Configure->Start*.

When this step is complete, a folder called `c:\TFTP-Root` is created. The folder name might be different depending on which vendor or version of TFTP is used.


Note

If you are running the TFTP Server on Windows, make sure that the firewall is open for TFTP communication.

Configure the vCenter Server

Step 2 On the vSphere client Home screen, verify that the Auto Deploy icon is present.

If the icon is not present, verify that auto deploy is actually installed on the vSphere server.

If auto deploy is not installed, download and install it now.

If auto deploy is installed but not running, check that the auto deploy listener service is running, then re-install.

Step 3 Click the Auto deploy icon to display the auto deploy screen.

Step 4 On the Auto deploy screen, find the BIOS DHCP location, and copy the file name indicated. You will use this information soon. An example of the file name is `undionly.kxpe.vmw.hardwired` or something similar.

Step 5 Click download `TFTP Boot.zip` and download the file to `c:\TFTP-Root` folder.

Step 6 Unzip the filed in `c:\TFTP-Root`.

Step 7 Click the folder to select it, then copy the location of the BIOS DHCP file name.

Step 8 Go to the DHCP Server and add the file name and IP address of the TFTP Server.

Step 9 Start-> Administrative Tools->DHCP

Step 10 Expand the IPv4->Scope->Address Pool and edit to update with the IP address range available to you.

Step 11 Expand the IPv4->Scope Options, then right-click Configure->Options and set the following options:

- number 066 Boot Server Host name with the IP address of the TFTP Server
- number 067 Bootfile Name with the BIOS DHCP file name (for example, `undionly.kxpe.vmw.hardwired).`

At the completion of this section, you should be able to boot your host and can verify that the interface is getting an IP address through the DHCP server.

Create the Boot Image for the ESXi 5.1 Hosts

After the IP address is assigned to the interface in the auto deployment server, auto deploy will fail due to an error that no ESXi images is associated. You will need to create a boot image with the Xsigo host drivers and the ESXi 5.1 ISO present by using the PowerShell CLI.

Follow this procedure:

Step 1 Open PowerCLI and connect to your vCenter Server.

Step 2 Run the following commands:

```
Add-EsxSoftwareDepot –DepotUrl c:\VMware-ESX-5.1.0-799733-depot.zip

Add-EsxSoftwareDepot http://<vcenter server ipaddr>/vShpere-HA-depot

Add-EsxSoftwareDepot –DepotUrl c:\<xsigo-hostdriver>.zip

New-EsxImageProfile –CloneProfile “ESXi-5.1.0-799733-standard” –name
“ESXiStatelessImage-Xsigo”
```

```
Add-EsxSoftwarePackage –ImageProfile “ESXiStatelessImage-Xsigo” –
SoftwarePackage vmware-fdm

Add-EsxSoftwarePackage –ImageProfile “ESXiStatelessImage-Xsigo” –
SoftwarePackage net-ib-core

Add-EsxSoftwarePackage –ImageProfile “ESXiStatelessImage-Xsigo” –
SoftwarePackage net-ib-mad

Add-EsxSoftwarePackage –ImageProfile “ESXiStatelessImage-Xsigo” –
SoftwarePackage net-ib-sa

Add-EsxSoftwarePackage –ImageProfile “ESXiStatelessImage-Xsigo” –
SoftwarePackage net-mlx4-core

Add-EsxSoftwarePackage –ImageProfile “ESXiStatelessImage-Xsigo” –
SoftwarePackage net-mlx4-ib

Add-EsxSoftwarePackage –ImageProfile “ESXiStatelessImage-Xsigo” –
SoftwarePackage net-ib-ipoib

Add-EsxSoftwarePackage –ImageProfile “ESXiStatelessImage-Xsigo” –
SoftwarePackage net-xscore

Add-EsxSoftwarePackage –ImageProfile “ESXiStatelessImage-Xsigo” –
SoftwarePackage net-xsvnic

Add-EsxSoftwarePackage –ImageProfile “ESXiStatelessImage-Xsigo” –
SoftwarePackage net-xve

Add-EsxSoftwarePackage –ImageProfile “ESXiStatelessImage-Xsigo” –
SoftwarePackage scsi-xsvhba

New-DeployRule –Name “FirstBoot” –Item “ESXiStatelessImage-Xsigo” –
AllHosts

Add-deployRule –DeployRule “FirstBoot”

Esxport-ESXimageProfile –ImageProfile “ESXiStatelessImage-Xsigo”
–EsxportToBundle –FilePath c:\ESXiStatelessImage-Xsigo.zip
```

At the completion of this procedure, you will have a bootable ISO with Xsigo host drivers included.

You can try doing autpdeploy as the rule is created.

```
Export-ESXimageProfile –ImageProfile “ESXiStatelessImage-Xsigo”
–ExportToIso –FilePath c:\ESXiStatelessImage-Xsigo.iso
```

Create the Host Profile for Auto Deploy

A Server Profile will be required to support auto deployment. After the boot image is created, perform the following procedure:

Step 1 Boot the auto deployed host to verify that auto deployment occurs successfully.

Step 2 Open the vSphere client and log in to the vCenter Server.

Step 3 Ins the Server list, you should find the auto-deployed host.

Step 4 Create a cluster and assign it a name—for example, `DEVCLUSTER`

Step 5 Add the auto-deployed host to the cluster.

Step 6 Right-click on the auto deployed host, and click `Enter Maintenance Mode`.

Step 7 Right-click on the host, and click *Host Proflie->Create Profiel From Host*.

Step 8 Enter a host name.

Step 9 Expand the menu bar by clicking View->Management->Host Profiles.

Step 10 Right click the new host profile you just created, then select Edit Profile.

Step 11 Edit the host prof8ile as needed—for example, specify network interface, login details, and so on).

Step 12 Right-click the host, then select *Host Proflie->Manage Profile*.

Step 13 Select the profile you just created.

Step 14 Expand the menu bar view->Management->Host Profiles->select the host profile created

The auto deployed host should be listed in the “Hosts and Clusters” option.

Step 15 In the “Hosts and Clusters” option, right-click the host and select Update answer file.

Step 16 Change any settings (if required).

Step 17 In the “Hosts and Clusters” option, right-click the host and select `Check Profile Compliance`.

Step 18 In the “Hosts and Clusters” option, right click on host and select `Apply Profile`.

Step 19 Start PowerCLI and update the auto-deploy rule with the host profile created:

```
New-DeployRule –name “ProductionBoot-Xsigo” –Item “ESXiStatelessImage-
Xsigo”, <host profile created>, <cluster created in vCenter Server>
-Pattern “vendor=xsigo”
```

Step 20 Issue the following commands:

```
Add-Deployrule –DeployRule “ProductionBoot-Xsigo”
Remove-DeployRule –DeployRule FirstBoot -delete
```

Auto Deploying through the Host Profile

After configuring the Server Profile, follow this procedure:

Step 1 Reboot the auto deploying server. It should get deployed with new rule created `ProductionBoot-Xsigo`

Step 2 After reboot you can check the network interface are similar to the one saved in host profile

Step 3 Further testing can be done by creating vnic on auto deploying server and creating a vSwitch,


Note

While creating the vNIC, you should follow the naming convention of: `vmnicX`, where `X` is a number.

Step 4 Create the Host Profile

Step 5 Save the profile and apply it on the auto deploying host. When the auto deploying host reboots, it should retain the configuration from host profile.

Configuring ESXi 5.0 Server SAN Boot

Configuring SAN Boot for VMware ESXi Server 5.0 systems is conceptually similar to configuring an ESXi 4.1 system, with the exception of creating a modified ISO image that includes the Xsigo host drivers. SAN Boot is supported for ESXi 5.0 servers through a procedure that has the following steps:

- Creating a modified ISO image for booting.
- Creating the SAN Boot server profile with a vHBA that can reach the target/LUN where the ISO image is installed.
- Installing the image on the ESXi 5.0 server.


Note

To complete this procedure, you will need some additional utilities (for example, mkisofs and tar/gzip/etc) for a typical default install.

Prerequisites

Before installing the new Xsigo host drivers, make sure that any previous version of host driver is removed. For example, if you are upgrading your ESX server from 4.1 to 5.0, you will need to uninstall the 4.1 host drivers.

```
esxupdate -b <bundle-id> remove
```

Creating a Modified ISO Image

To have the Xsigo vNICs and vHBAs available to the ESXi 5.0 OS for PXE or SAN Booting, you will need to inject the Xsigo host drivers into the native ESX OS. This procedure documents how to inject the Xsigo devices into the ESXi 5.0

bundle. You will basically use an existing ESXi 5.0 bundle, use a downloadable tool to inject the Xsigo host drivers into the ESX OS, then use another downloadable tool to repackage the modified ISO.

Before creating a modified ISO image for ESXi 5.0 hosts, be aware of the following:

- Creating the custom ISO is accomplished through Microsoft Windows PowerShell—and specifically the VMware vSphere PowerCLI plug-in for PowerShell. The Windows server will need this tool installed. Make sure the Windows server has the correct requirements to run PowerShell and PowerCLI.
- Creating the custom ISO is supported on a Windows host server only. The server requirements are determined by the PowerShell application.
- You use a pre-configured ESXi bundle as a baseline, then inject the Xsigo bits into it:
  - For ESXi 5.0 Update 0 (GA), the ESXi bundle is `VMware-ESXi-5.0.0-469512-depot.zip` and is available from VMware's website.
  - For ESXi 5.0 Update 1, the ESXi bundle is `VMware-ESXi-5.0.0-623860-depot.zip`
- You will need full administrative rights on the Windows server where you will be creating the custom ISO.

The following procedure assumes the working directory is `\images\New` for the user "adminA". The procedure also uses the VMware 5.0 GA bundle (`469512`) for illustrative purposes. If your hosts are running ESXi 5.0 Update 1, you will need to use the bundle with build number `623860`.

To create the modified ISO for ESXi 5.0 hosts, follow this procedure:

Step 1 Install PowerShell on the Windows server if you have not done so already.

Step 2 Install the PowerCLI plug-in if you have not done so already.

Step 3 Download the `VMware-ESXi-5.0.0-469512-depot.zip` file to the Windows server.

Step 4 Start PowerCLI.

Step 5 In PowerCLI, run the following commands to import the ESXi 5.0 bundle and the Xsigo host drivers into PowerCLI:

```
Add-EsxSoftwareDepot -DepotUrl C:\Users\adminA\Desktop\images\New\VMware-ESXi-5.0.0-469512-depot.zip

Add-EsxSoftwareDepot -DepotUrl C:\Users\adminA\Desktop\images\New\xsigo_5.0.1ESX.1-1vmw.500.0.0.406165.zip
```

Step 6 Run the following commands to specify the file that you want to use when creating the output ISO. The profile determines metadata about the output ISO, such as formatting, compression method, and so on. In this example, the profile is named `ESXi-5.0.0-469512-standard-xsigo` for illustrative purposes.

```
New-EsxImageProfile -CloneProfile ESXi-5.0.0-469512-standard -name "ESXi-5.0.0-469512-standard-xsigo"
```


Note

Notice that the **`-name`** string supplied for the profile is enclosed in quotation marks. The quotation marks are required syntax for the profile's name.

Step 7 Run the following commands to add the IB stack and other dependencies to the depot.

```
Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage net-ib-core

Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage net-mlx4-core

Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage net-ib-mad

Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage net-ib-sa

Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage net-ib-ipoib

Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage net-mlx4-ib

Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage net-xscore

Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage net-xsvnic

Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage net-xve

Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage scsi-xsvhba
```

Step 8 Run the following command to create single output ISO containing all required files from the depot. The following example assumes unsigned drivers to provide the most complete example.

```
Export-EsxImageProfile -ImageProfile ESXi-5.0.0-469512-standard-xsigo -ExportToIso
-FilePath C:\Users\adminA\Desktop\images\New\VMware-VMvisor-Installer-5.0.0-
469512_Xsigo.x86_64.iso -NoSignatureCheck
```

Note

Xsigo makes every effort to release signed, certified host drivers. However, on some occasions, unsigned drivers might be released. If you receive unsigned Xsigo host drivers, the **`Export-EsxImageProfile`** command has the **`-NoSignatureCheck`** option which will bypass signature checking.

Use the `-NoSignatureCheck` for unsigned drivers.

Omit the `-NoSignatureCheck` option if the drivers are signed.

Creating a SAN Boot Server Profile

If you have not already created a SAN Boot server profile, you must do so. The SAN Boot Server profile must have at least one vHBA that is connected to the storage where the ESXi 5.0 boot image will reside.

Step 1 Create the server profile. For example, to create the server profile “esx50” for “server10” which is connected through IB port 23 on Oracle Fabric Director “tuffy:

```
add server profile esx50 server10@tuffy:ServerPort23
```

Step 2 Create the vHBA for the SAN Boot Server Profile. For example, to create a vHBA named “vhba1” in server profile “esx50” and have the vHBA terminated on the fibre channel port 1 in slot 8:

```
add vhba vhba1.esx50 8/1
```

Step 3 Set the Server Profile for SAN Booting and connected to the WWN of the target where the boot image will reside:

```
set server-profile esx50 san-boot vhba1 22:00:00:50:CC:20:0E:6E 203
```

Step 4 Verify that the server profile is up and connected as shown.

```
show server-profile esx50 san-boot
```

```
server    role          vhba    mnt-type   lvm-grp    lvm-vol         dev        mnt-opts disks
-------------------------------------------------------------------------------------------
esx50     loadmount     vhba1                                    22:00:00:50:CC:20:0E:6E
(203/LM)
```


Note

Make a note of the size of the LUN on which the boot image will reside. You will be prompted to select the correct LUN when you load the host drivers onto the ESXi 5.0 server, and knowing the LUN’s size will help you select it from the list of connected storage targets.

With the modified ISO ready and the SAN Boot server profile created, you will need to load the ISO onto the ESXi 5.0 server and reboot the server so that the SAN Boot vHBA is recognized as the primary boot option for the ESXi 5.0 server.

Loading the Image into the ESXi 5.0 Server

When the modified ISO image has been created and put on a network reachable device, and a SAN Boot server profile has been created, you can now load the ISO image into the ESXi 5.0 server, and boot the server. When it boots, you will have an option to select the LUN that contains the SAN Boot ISO.


Note

The following procedure assumes the installation of the SAN Boot image and configuration of the SAN Boot feature through an ILO or DRAC.

To load the SAN Boot image into the ESXi 5.0 server, follow this procedure:

Step 1 From the ESXi 5.0 server, locate and mount the modified ISO (`ESXi-5.0.0-Standard-Xsigo Installer`) as shown in Figure 2 on page 32.

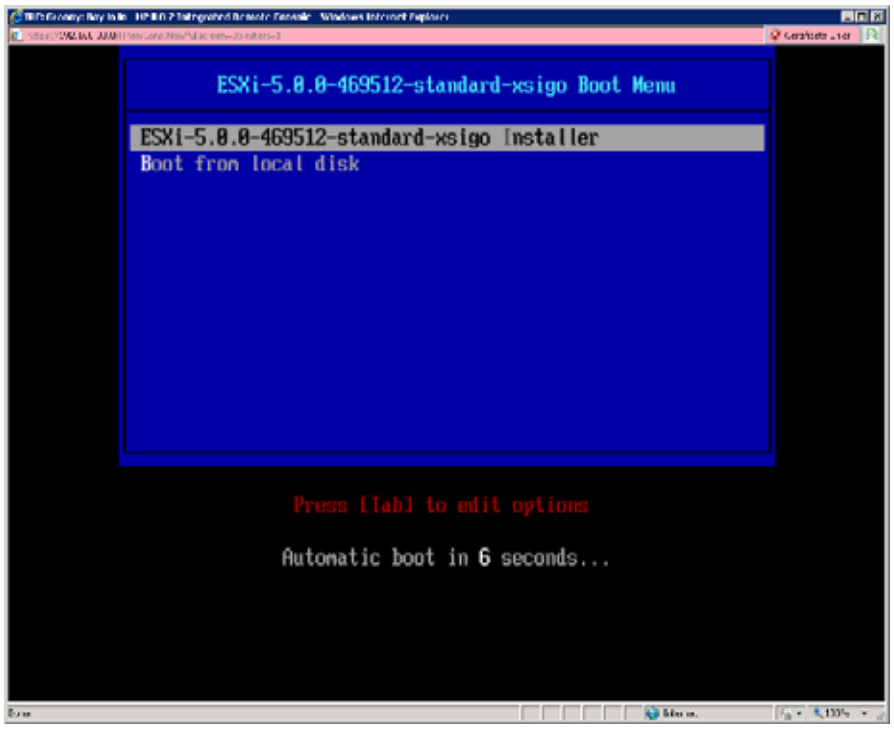


Figure 2 Selecting the Xsigo ESXi 5.0.0 Installer

When the `ESXi-5.0.0-standard-xsigo Installer` is mounted, files are loaded onto the server. You will see output similar to Figure 3 on page 33.

Note

Be aware that loading files onto the server can take a long time. You must wait for it to complete.

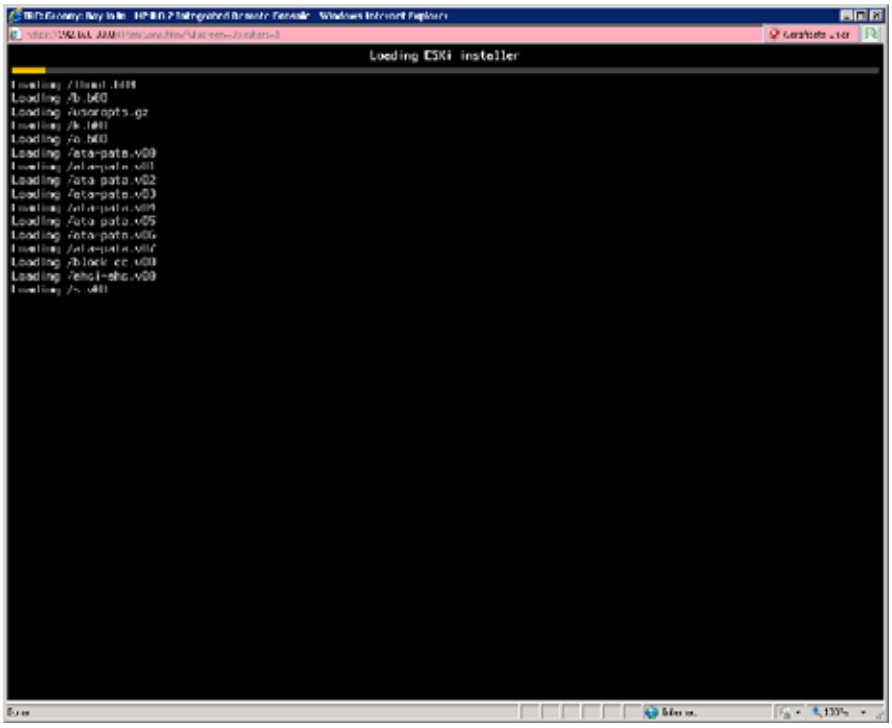


Figure 3 Loading OS Files from Modified ISO Image

Step 2 After the OS files have been loaded, the server boots to the ESXi 5.0 native operating system. See Figure 4 on page 34.

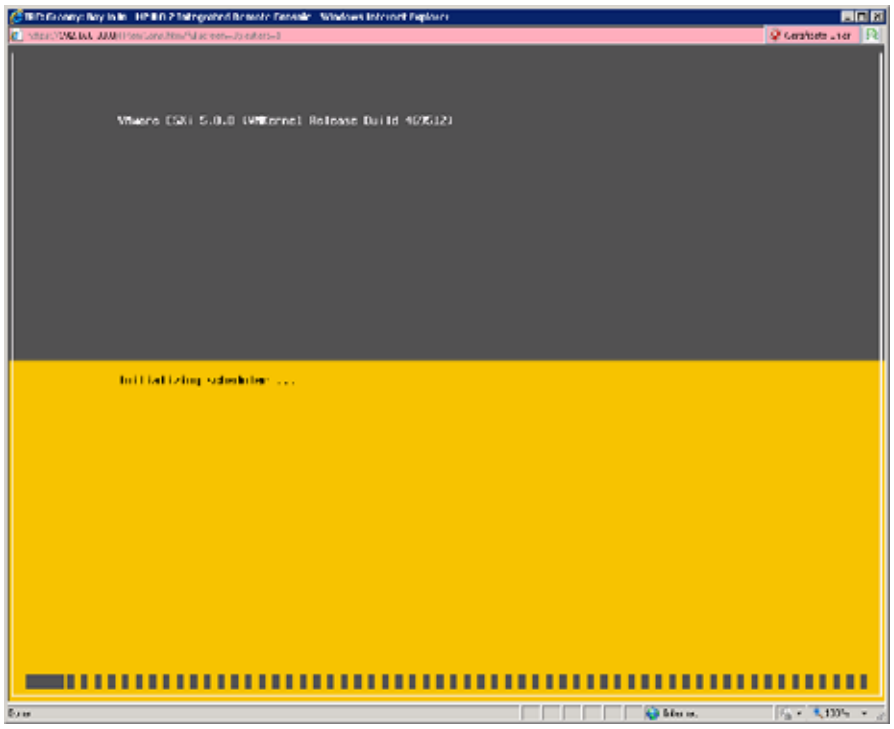


Figure 4 Boot Up

Step 3 As part of the boot sequence, you must acknowledge license agreement. See Figure 5 on page 35.

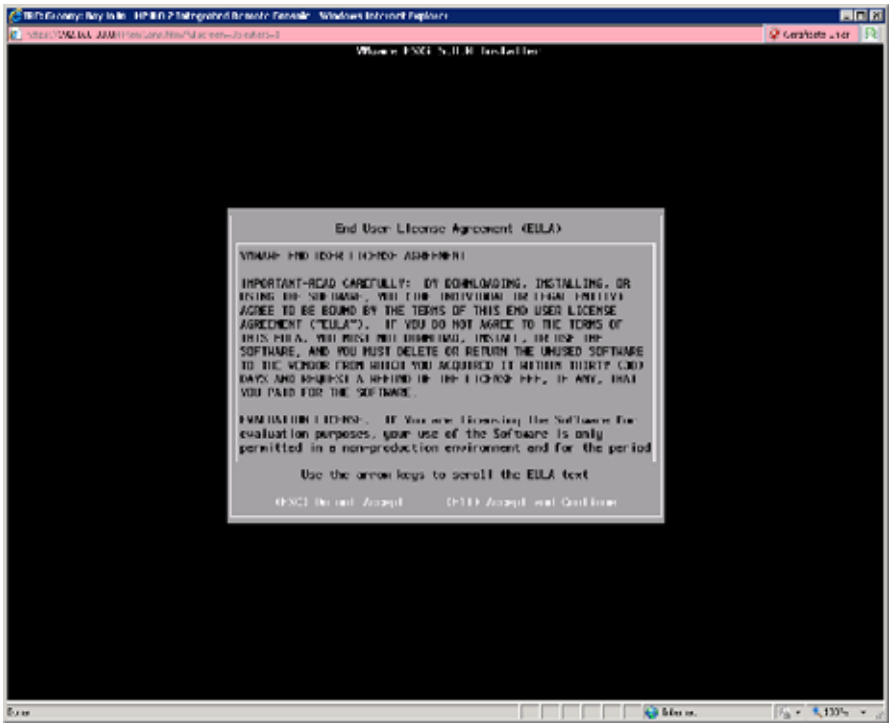


Figure 5 End User License Agreement for ESXi 5.0

Step 4 Accept (***F11)*** the license agreement (or decline by pressing ***Esc***) as needed. To continue the installation procedure, accept the license agreement and start the ESXi 5.0 installer. Figure 6 on page 36 shows the ESXi 5.0 installer Welcome screen.

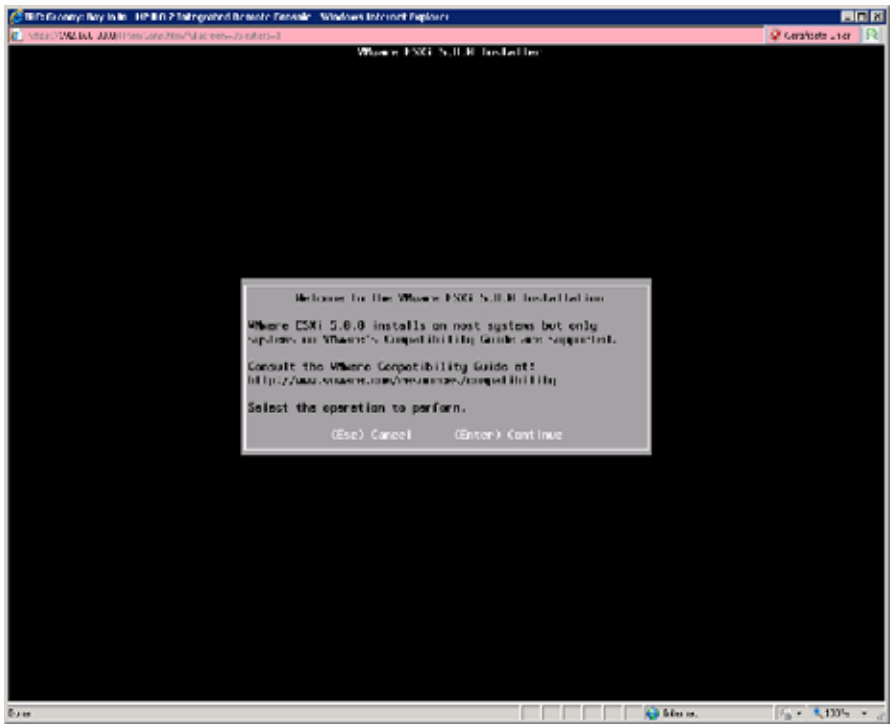


Figure 6 ESXi 5.0 Installer

Step 5 Press ***Enter*** to continue the installer.

Follow the installer until you are prompted to specify a boot disk on the Select a Disk to Install or Upgrade dialog.

Step 6 On the Select a Disk to Install or Upgrade dialog, select the LUN on which the SAN Boot vHBA is connected. Figure 7 on page 37 show an example of this dialog.

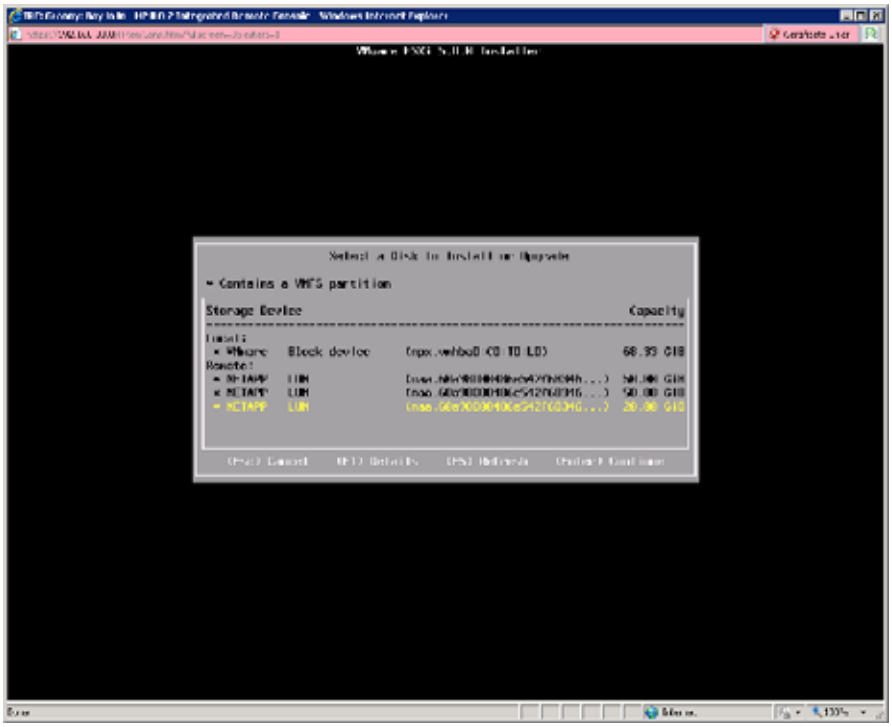


Figure 7 Select a Disk for Booting

Note

Notice that you can get details about a LUN (if desired) by pressing ***F1***.

Step 7 Press ***Enter*** to continue the installer until you see the Confirm Install dialog.

If the LUN has already had ESXi 5.0 installed, a VMware file system will be labelled on the LUN. The installer will prompt you to determine how you want to proceed, as shown in Figure 8 on page 38.

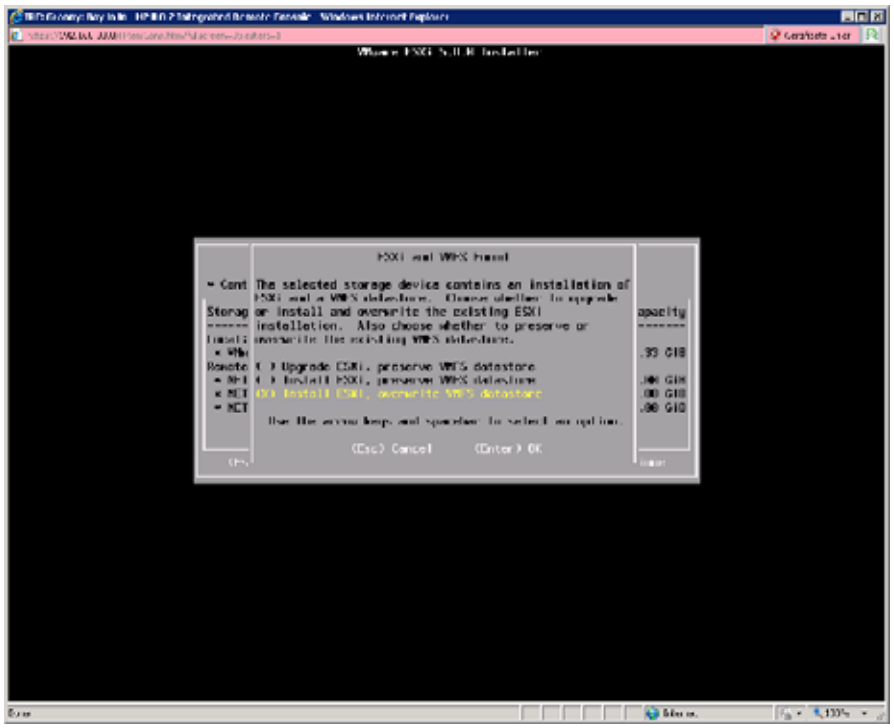


Figure 8 Format VMFS Dialog

Step 8 On the VMFS Format dialog, select the option as needed for your installation:

- Upgrade ESXi, preserve VMFS datastore.
- Install ESXi, preserve VMFS datastore
- Install ESXi, overwrite VMFS datastore. This is required for a fresh install.

Note

Additional dialogs for the keyboard type and language are displayed. Make sure to select the correct option for your ESX installation.

Step 9 When prompted, log in to the ESXi 5.0 server. You will need to log in to the server to complete the installation. Figure 9 on page 39 shows the log in prompt.

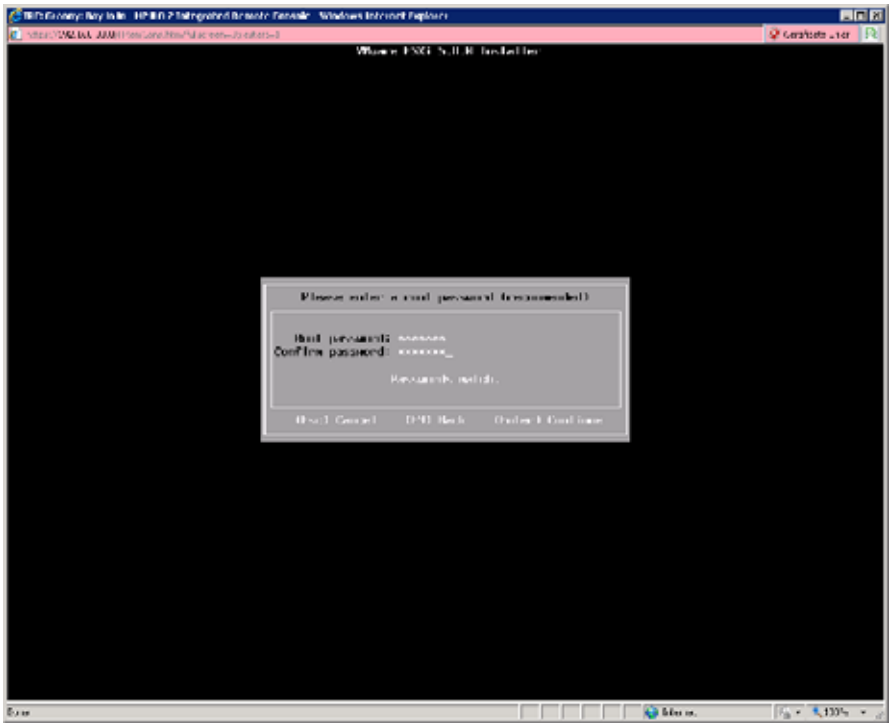


Figure 9 Log In to ESXi 5.0 Server

After you log in, the installation runs to completion. The Confirm Installation dialog is displayed. Figure 10 shows this dialog.

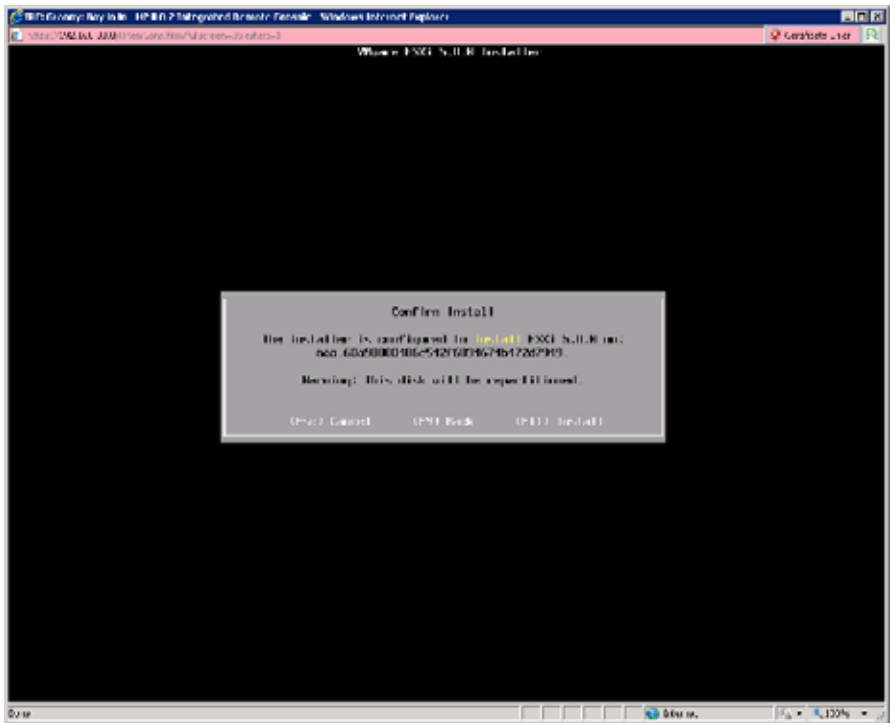


Figure 10 Confirm Install

Step 10 Press ***F11*** to confirm that you want to continue the install. When you press ***F11***, the OS and Xsigo host drivers are loaded onto the ESXi 5.0 Server, as shown in Figure 11 on page 41.

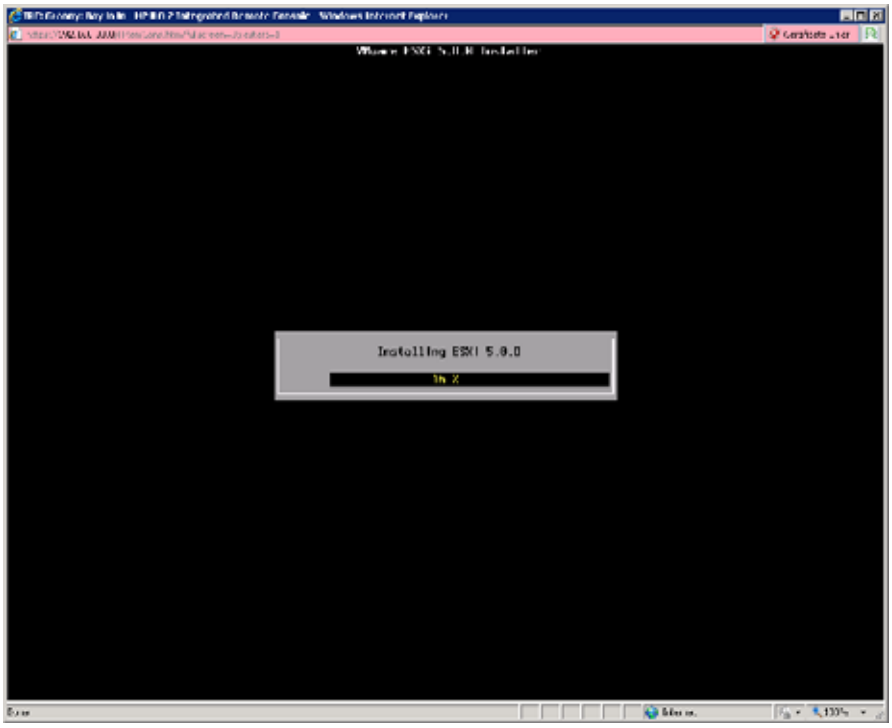


Figure 11 ESXi 5.0 OS and Drivers Loading

Step 11 When the OS and host drivers are loaded onto the ESXi 5.0 Server, you are prompted to finalize the installation, as shown in figure Figure 12 on page 42.

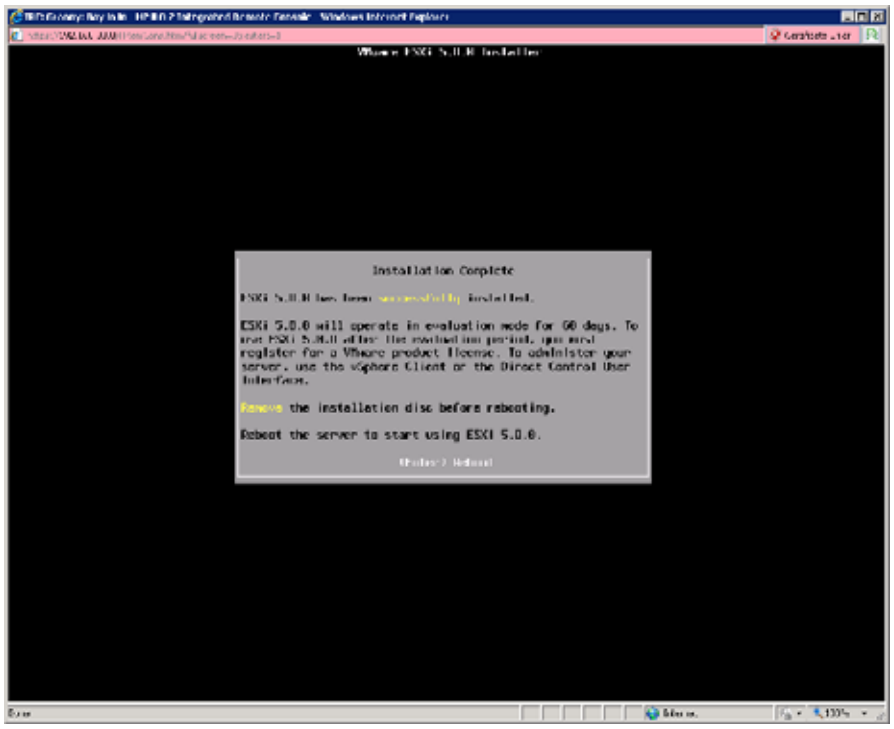


Figure 12 Finalizing the Install

As part of finalizing the OS and host driver installation, the server must be rebooted to load the new OS and host drivers into memory.

Step 12 Press ***Enter*** to reboot the server. When the server reboots, it progresses through the boot devices until it locates the SAN Boot vHBA from which it retrieves the SAN Boot image, as shown in Figure 13 on page 43.

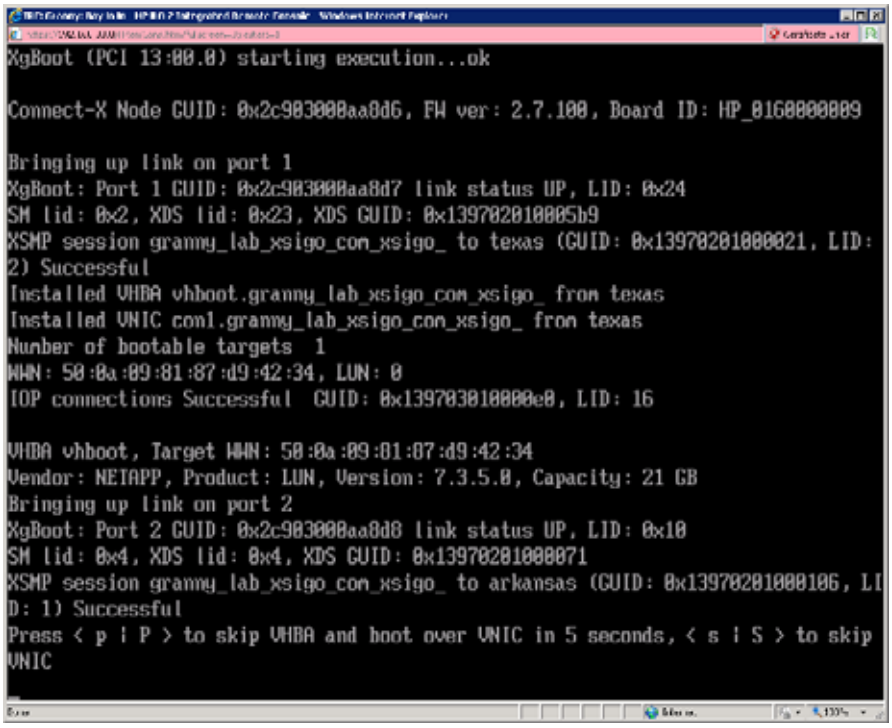


Figure 13 SAN Booting from the SAN Boot vHBA

A NIC card is attempted first, then the vHBA. The text “VHBA installing” should indicate the name of the vHBA configured for SAN Booting. If this text is incorrect, then you should reattempt this procedure.

After the SAN Boot vHBA is recognized as a boot device, the ESXi 5.0 server completes its boot up. Virtual NICs and virtual HBAs can be configured in the ESX server to provide the benefits of Xsigo virtual I/O.

Configuring ESX Classic 4.1 SAN Boot

SAN Boot is supported for ESX Classic 4.1. The SAN Boot configuration for ESX 4.1 servers occurs primarily through the ESX 4.1 Installer, with a minor amount of custom configuration for the Xsigo ESX host driver and InfiniBand stack.

The configuration procedure for ESX 4.1 Server SAN Boot has the following main parts:

- Create a Server Profile for SAN Boot including at least one vHBA that will connect to the LUN where the SAN Boot image will be kept.
- Have one LUN available for the install.
- Insert the latest ESX 4.1 install CD and follow the prompts until prompted with the Add Custom Drivers page.
- Insert the Xsigo Host Drivers ISO, and load the drivers.
- Enter the debug shell, and run the Xsigo `install-load`. This is an option for the **`esxcfg-xgutil`** command (`/tmp/drivers/user/sbin/esxcfg-xgutil install-load`).
- Return to the ESX 4.1 Installer and complete the install.
- Make sure to reboot the server.

Before configuring ESX 4.1 SAN boot, be aware of the following considerations:

- If a VMFS exists on the ESX Server 4.1 that will be configured for SAN Boot, that VMFS must be deleted from the ESX Server before configuring the ESX 4.1 SAN Boot. The process of configuring SAN Boot for ESX 4.1 creates a separate instance of VMFS, and sometimes does not remove any existing VMFS on the ESX server. For example, if you are upgrading from an earlier update of ESX 4.1 SAN Boot to ESX 4.1 SAN Boot, or if you are re-installing on an ESX 4.1 Server and configuring SAN Boot, you must remove any existing VMFS before beginning the ESX 4.1 SAN Boot configuration procedure.
- While up to 4 LUNs have been tested, Xsigo recommends that you make only one LUN available to the vHBA in the SAN Boot Server Profile.
- Also, be aware that the ESX installer does not allow for reformatting any unknown partitions, so you might need to clear entries from the partition table.


Note

The following procedure assumes the installation of the SAN Boot image through a “lights out” management solution.

To configure the ESX 4.1 SAN Boot, follow this procedure:

Step 1 Make sure that you have a SAN Boot server profile configured.

Step 2 Make sure that the vHBA present in the SAN Boot server profile can reach the LUN from which the server will be SAN Booted.

Step 3 Insert the latest ESX 4X install CD and wait for the CD to autorun and display the (top-level) *Options* menu.

Step 4 In *Options* menu, use the up and down arrows to select the *Install ESX in graphical mode* option. This option allows you to set or modify boot-loader arguments.

Step 5 With the *Install ESX in graphical mode option* highlighted, press *F2* key to enter edit mode for the Boot Options. Figure 14 shows the *Boot Option* with the default kernel memory shown (512M).

Figure 14 ESX 4.1 Installer — Entering Boot Options

Step 6 Backspace over the `mem=` argument and overwrite the default kernel memory with the recommended minimum amount of memory. By default, 1024 MB of memory is used, but you can set the value higher if desired), then press *Enter* to resume the installer, as shown in Figure 15 on page 46.

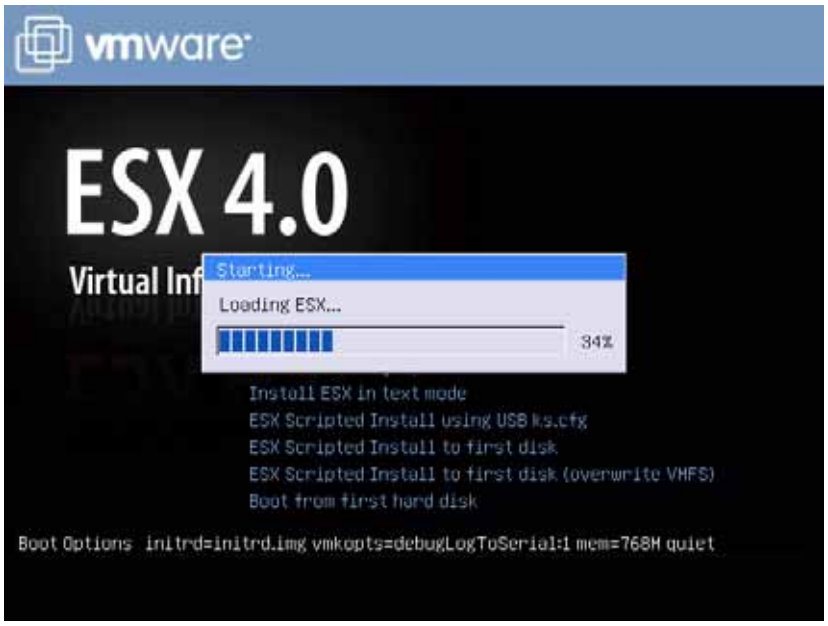


Figure 15 ESX 4.1 Installer — Loading

When the Install progress bar completes, the Welcome screen is displayed, as shown in Figure 16.

Figure 16 ESX Installer — Welcome Screen

Step 7 Click ***Next*** and proceed through the ESX Installer. You will need to read and acknowledge the license agreement, and answer all prompts until the Custom Drivers dialog is displayed.

Step 8 On the Custom Drivers dialog, in the Load Custom Drivers section, click *Yes*. This step displays a popup that alerts you to select the drivers that you want to install., as shown in Figure 17.

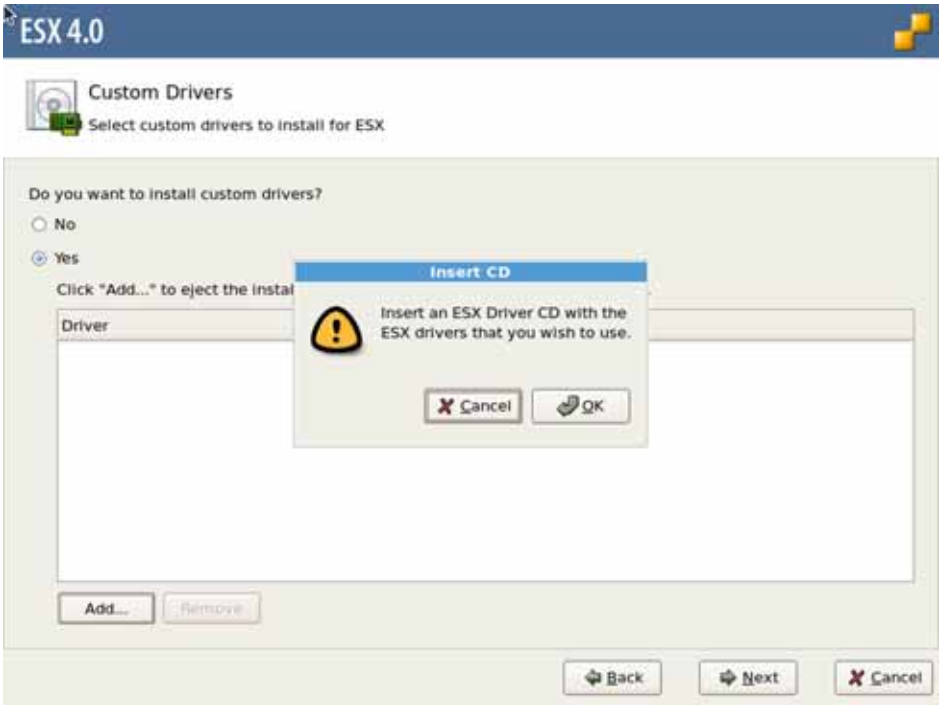


Figure 17 ESX Installer — Specify Custom Drivers

Step 9 Click ***OK*** to close the popup, then click ***Add...*** and browse to the location where the Xsigo host drivers are. You will need to install the driver ISO file to be able to connect to the Xsigo ESX host drivers ISO.

Note

The Xsigo ESX host drivers must be accessible to the ESX Installer for them to be successfully installed. For example, the Xsigo ESX host drivers can be added to the Installer directly from the `xsigo.iso` or from a network share.

Step 10 Double click each Xsigo Host Driver ISO to add them to the ESX Installer. The required modules are:

- `ib-basic`
- `vmware-esx-drivers-net-xxxx`
- `vmware-esx-drivers-ulp-xxxx`

Figure 18 on page 48 show the Xsigo host drivers added to the ESX installer.

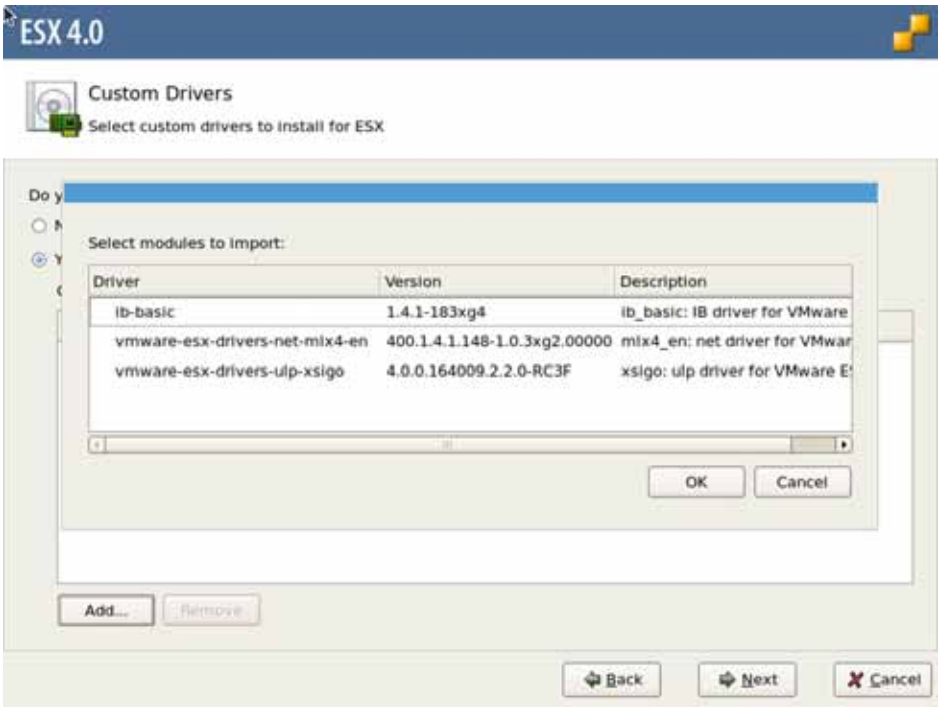


Figure 18 ESX Installer — Xsigo Drivers Specified

Step 11 Select (highlight) all 3 drivers and click ***OK*** to add them to the Drivers table in the ESX Installer.

You will see an additional popup that warns you about loading custom drivers. Click the “I Accept” check box, and click ***I Accept***. The Load Drivers popup is displayed as shown in Figure 19 on page 49.

Figure 19 ESX Installer — Load ESX and Custom Drivers

Step 12 Click ***Yes*** to continue the installation. A progress bar is displayed while the ESX drivers and Xsigo ESX host drivers progress through-installation.

Step 13 Continue the installer until you see the Network Configuration dialog. This dialog is where you will suspend the installer and enter the debug shell to run a Xsigo script.

Step 14 Press ***Alt+F2*** to enter the debug shell on the ESX Server. This step suspends the ESX Installer so that you can run the Xsigo install-load script. Figure 20 on page 50 shows the debug shell.

Figure 20 ESX Debug Shell

Step 15 In the debug shell, press ***Enter*** to activate the ESX console.

Step 16 At the prompt, run the Xsigo install-load script:

```
/tmp/drivers/usr/sbin/esxcfg-xgutil install-load
```

The script loads necessary software modules and verifies that the SAN Boot disk is available, as shown in Figure 21.

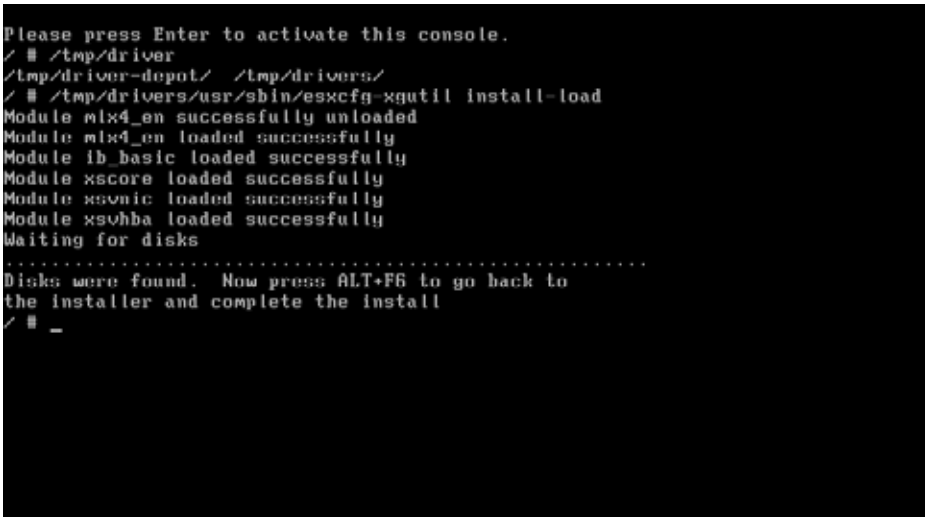


Figure 21 ESX Debug Shell — Running the Xsigo install-load Script

Step 17 When the script is completes, press ***Alt+F6*** to exit the debug shell and return to the ESX Installer.

Step 18 On the Specify ESX Datastore dialog, you will need to specify the datastore for virtual machines on the ESX Server. Make sure to select:

- Create new datastore
- Create on the same device as ESX
- Alternatively, you can preserve an existing data store by clicking *Use existing datastore* and specifying the partition. This option requires that the datastore be from the same version of ESX. For example, an ESX Classic 4.1 datastore used for SAN Booting an ESX 4.1 Classic server.

Figure 22 on page 51 shows a sample of the Specify a Datastore dialog.

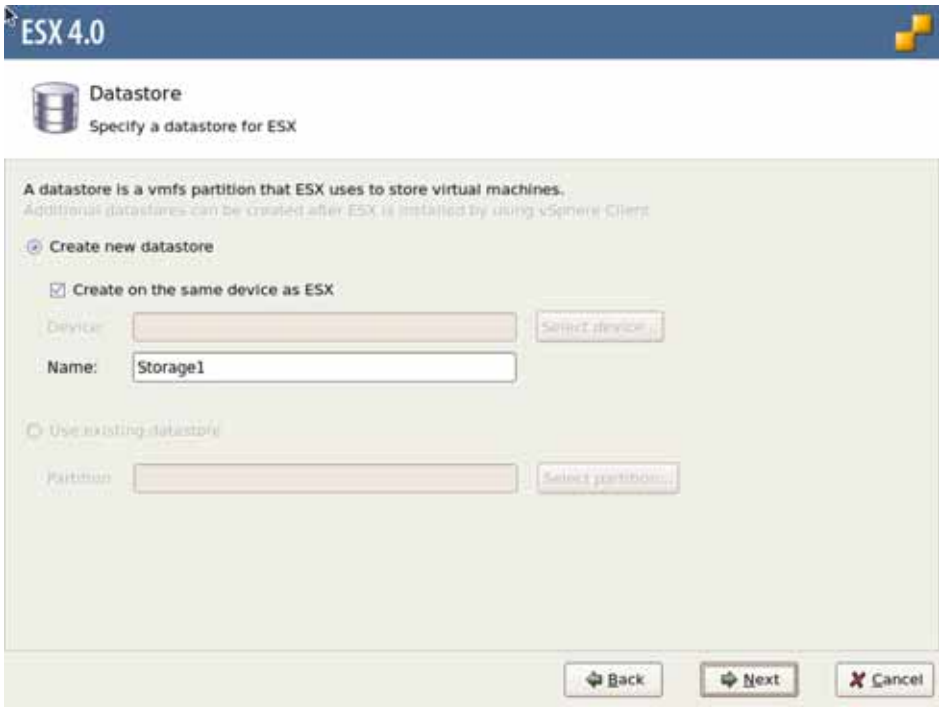


Figure 22 Specify the Datastore for ESX

Step 19 Continue the ESX Installer by configuring the standard information required by the ESX 4.1 Server, such as:

- Specifying the IP address method for the ESX Server.
- Selecting Standard Setup to boot off of a single LUN or hard drive
- Selecting the storage target that contains the LUN.

Note

If you are upgrading an existing ESX Server (or re-installing) and the target was previously associated with the ESX Server, you will see additional popups that assist you with performing a clean install.

- Selecting the time zone in which the ESX Server is being configured for SAN Boot.
- Specifying the NTP server (if applicable) with which the ESX Server will synchronize.
- Setting the administrator root password.

Step 20 When all ESX drivers and Xsigo host drivers, and SAN Boot options are specified, verify the intended configuration by reviewing the Summary of Installation dialog.Figure 23 shows an example of this dialog for illustrative purposes only. Your actual dialog will differ.

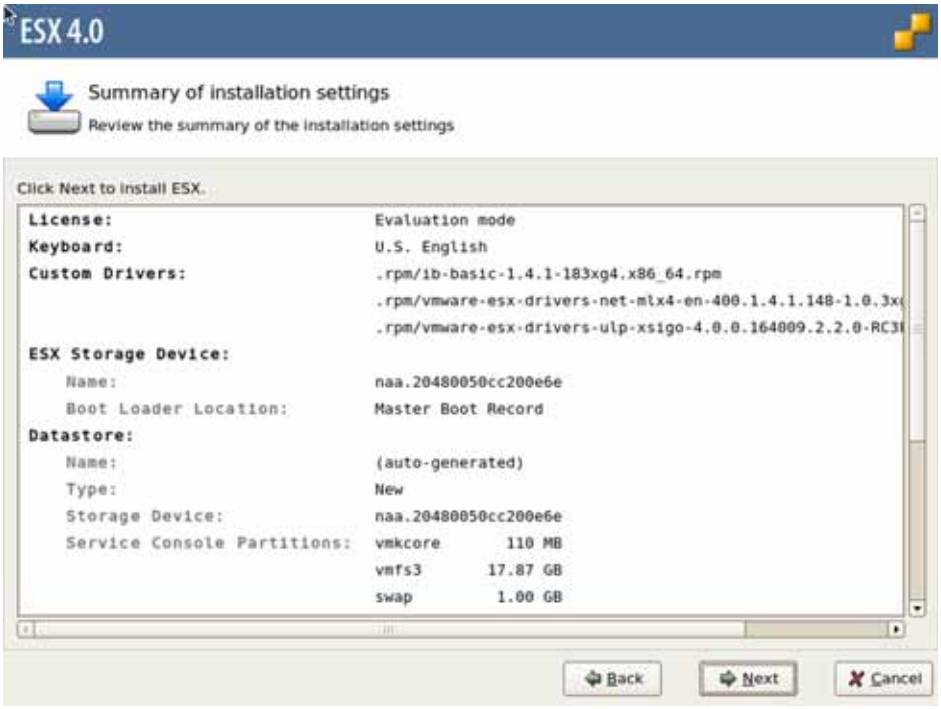


Figure 23 ESX Installer — Review SAN Boot and Other Settings

Step 21 Reconnect to the ESX Installer (or re-insert the physical medium if you are installing from DVD). If you see the popup shown in Figure 24 on page 53, you must reconnect to (or re-insert) the ESX installer DVD.

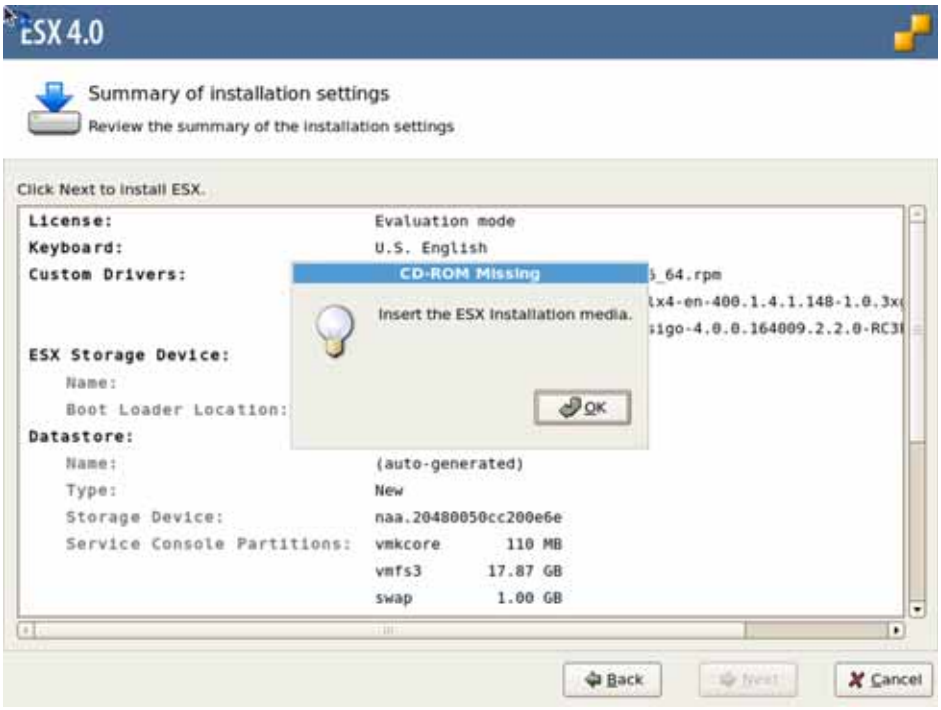


Figure 24 ESX Installer — Insert the Installation Medium

Step 22 Complete the installation by reviewing and confirming the remaining dialogs.

Step 23 Reboot the ESX Server.

Configuring ESXi 4.1 Server SAN Boot

SAN Boot is supported for the ESXi 4.1 servers through a procedure that has the following steps:

- Creating a modified ISO image for booting.
- Creating the SAN Boot server profile with a vHBA that can reach the target/LUN where the ISO image is installed.
- Installing the image on the ESXi 4.1 server.


Note

To complete this procedure, you will need some additional utilities (for example, mkisofs and tar/gzip/etc) for a typical default install.

Creating a Modified ISO Image

For this procedure, you will need both of the following:

- `VMware-VMVisor-Installer-4.1.0-171294.x86_64.iso` . Xsigo does not provide the VMware installation medium, so it is your responsibility to obtain it.
- `xsigo-esx-driver-disk-4.1.0.260247.3.5.0-14.iso`. This bundle contains a shell script (`xsigo-add-drivers-esx41i.sh`) that you will use to insert the Xsigo host drivers for ESXi 4.1 into the ISO image. Xsigo does provide this utility. You can download it from the Xsigo Support Portal as part of the host drivers bundle.

You will need to inject the Xsigo host drivers into the ESXi 4.1 bundle by decompressing the overall bundle and inserting the Xsigo host drivers, then recompressing all files into the ISO image. This modified image will be used as the boot image.

To create the modified ISO image, you will need:

- to be logged in to any Linux system with root privileges
- execute privileges on the ISO
- execute privileges on whatever directory you will use to uncompress and recompress the modified ISO.
- You will be using the `xsigo-add-drivers-esx41i.sh` shell script which has the following syntax:

```
xsigo-add-drivers-esx41i.sh --esxiso --driveriso --extrarpm --output -d
--isolinuxcfg --ksconfig --hca-installer-hooks
```

To create the modified ISO image, follow this procedure:

Step 1 Get the VMware installation medium onto a Linux host and place it into a working directory. For illustrative purposes, `opt/` is shown, but you can use whatever directory you want.

Step 2 Locate the `xsigo-4.1.0.260247.esx41i.tgz` and uncompress it to the same directory. For illustrative purposes, `opt/` is shown, but you can use whatever directory you want:

```
tar -zxvf xsigo-4.1.0.164009.3.5.0-14.esx41i.tgz opt/
```

Step 3 After the bundle is unzipped, run the `xsigo-add-drivers` script and recompress the bundle. This step will take a few minutes, but progress messages will be displayed to indicate the individual stages in the overall job.

sudo sh opt/xsigo/contrib/xsigo-remaster-esx41i-iso.sh --xgfile xsigo-4.1.0.164009.2.2.0.esx41i.tgz --iso VMware-VMvisor-Installer-4.1.0-171294.x86_64.iso

```
Copying base ISO VMware-VMvisor-Installer-4.1.0-260247.x86_64.iso

Unpacking ISO

Modifying the base installer image

Repacking installer image

Adding Xsigo-drivers to the base CD

Make ISO Image XG-VMware-VMvisor-Installer-4.1.0-260247.x86_64.iso

sudo sh remaster-esx41i-iso.sh --iso  --xgfile  -d  152.70s user 14.97s system 85% cpu 3:16.47 total
```

When the command completes, the new ISO image is created as XG-VMware-Visor-Installer. You will use this image to SAN boot the ESXi 4.1 Server.

Creating a SAN Boot Server Profile

If you have not already created a SAN Boot server profile, you must do so. The SAN Boot Server profile must have only one vHBA that is connected to the storage where the ESXi 4.1 boot image will reside.

Step 1 Create the server profile. For example, to create the server profile “esx41i” for “server10” which is connected through IB port 23 on Oracle Fabric Director “tuffy:

add server profile esx41i server10@tuffy:ServerPort23

Step 2 Create the vHBA for the SAN Boot Server Profile. For example, to create a vHBA named “vhba1” in server profile “esx41i” and have the vHBA terminated on the fibre channel port 1 in slot 8:

add vhba vhba1.esx41i 8/1

Step 3 Set the Server Profile for SAN Booting and connected to the WWN of the target where the boot image will reside:

set server-profile esx41i san-boot vhba1 22:00:00:50:CC:20:0E:6E 203

Step 4 Verify that the server profile is up and connected as shown.

show server-profile esx41i san-boot

```
server    role         vhba    mnt-type   lvm-grp    lvm-vol   dev    mnt-opts   disks
-------------------------------------------------------------------------------------
esx41i    loadmount    vhba1  static                     22:00:00:50:CC:20:0E:6E (203/LM)
```

Note

Make a note of the size of the LUN on which the boot image will reside. You will be prompted to select the correct LUN when you load the host drivers onto the ESXi 4.1 server, and knowing the LUN's size will help you select it from the list of connected storage targets.

With the modified ISO ready and the SAN Boot server profile created, you will need to load the ISO onto the ESXi 4.1 server and reboot the server so that the SAN Boot vHBA is recognized as the primary boot option for the ESXi 4.1 server.

Loading the Image into the ESXi 4.1 Server

When the modified ISO image has been created and put on a network reachable device, and a SAN Boot server profile has been created, you can now load the ISO image into the ESXi 4.1 server, and boot the server. When it boots, you will have an option to select the LUN that contains the SAN Boot ISO.

Note

The following procedure assumes the installation of the SAN Boot image and configuration of the SAN Boot feature through an ILO or DRAC.

To load the SAN Boot image into the ESXi 4.1 server, follow this procedure:

Step 1 From the ESXi 4.1 server, locate and mount the modified ISO (`XG-VMware-Visor-Installer`). When the ISO is mounted, ESXi 4.1 begins loading all pertinent OS files, as shown in Figure 25 on page 57.

Note

Be aware that this step can take a long time. You must wait for this step to complete.

When the `XG-VMware-Visor-Installer` is mounted, you will see output similar to the following.

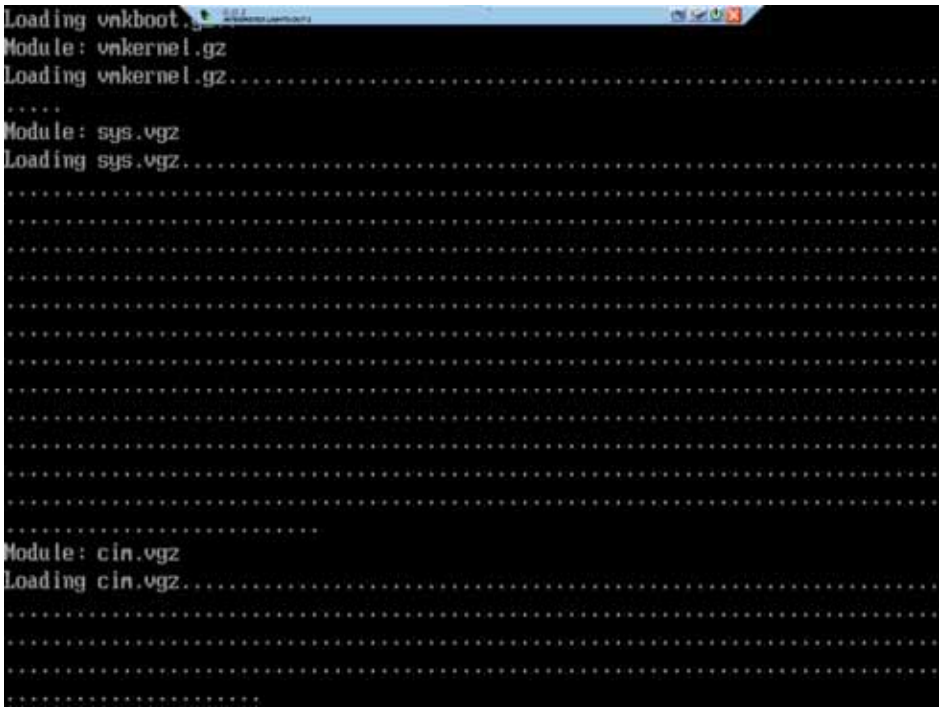


Figure 25 Loading OS Files from Modified ISO Image

Step 2 After the OS files have been loaded, the server boots to the ESXi 4.1 installer. See Figure 26 on page 58.

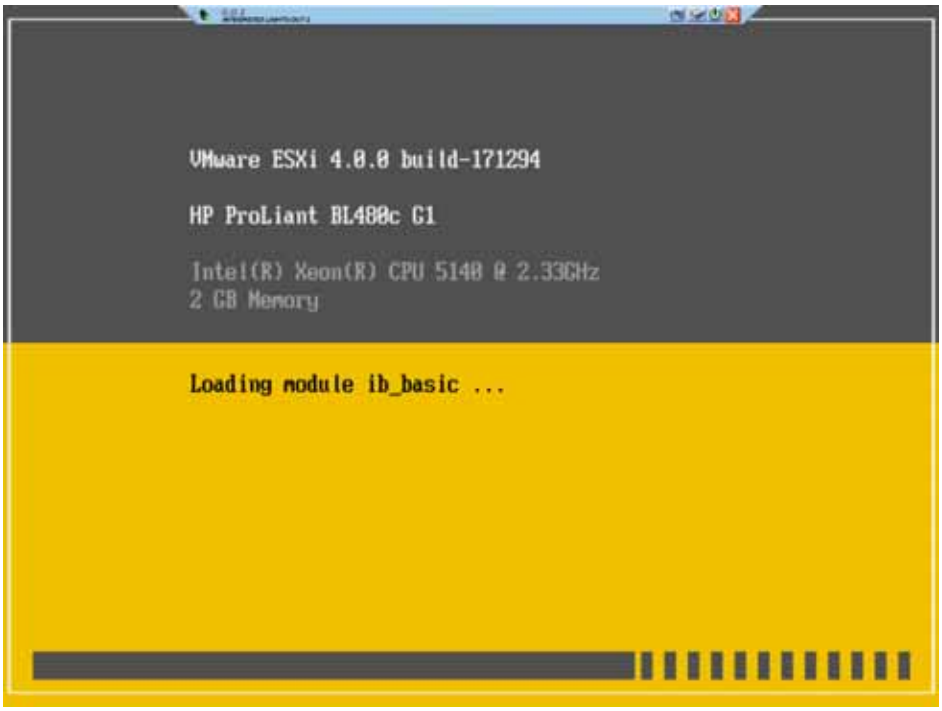


Figure 26 Boot Up

Step 3 As part of the boot sequence, you must read the license agreement.

Step 4 Accept (or Decline) the license agreement as needed. To continue the installation procedure, accept the license agreement and start the ESXi 4.1 installer. Figure 27 on page 59 shows the ESXi 4.1 installer.

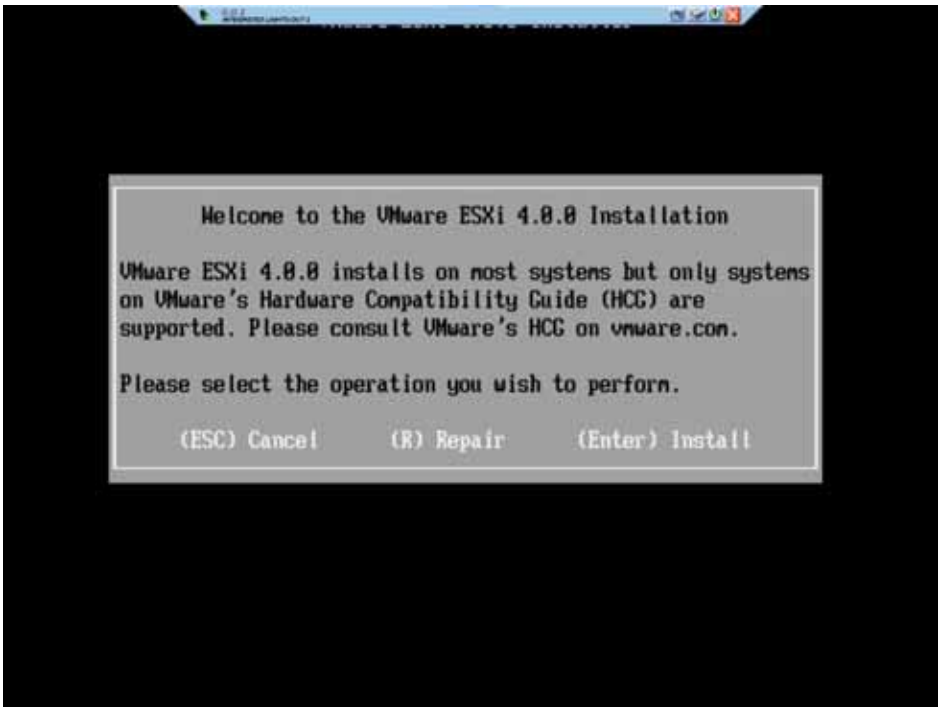


Figure 27 ESXi 4.1 Installer

Follow the installer until you are prompted to specify a boot disk on the Select a Disk dialog.

Step 5 On the Select a Disk dialog, select the LUN on which the SAN Boot vHBA is connected. Figure 28 on page 60 show an example of this dialog.

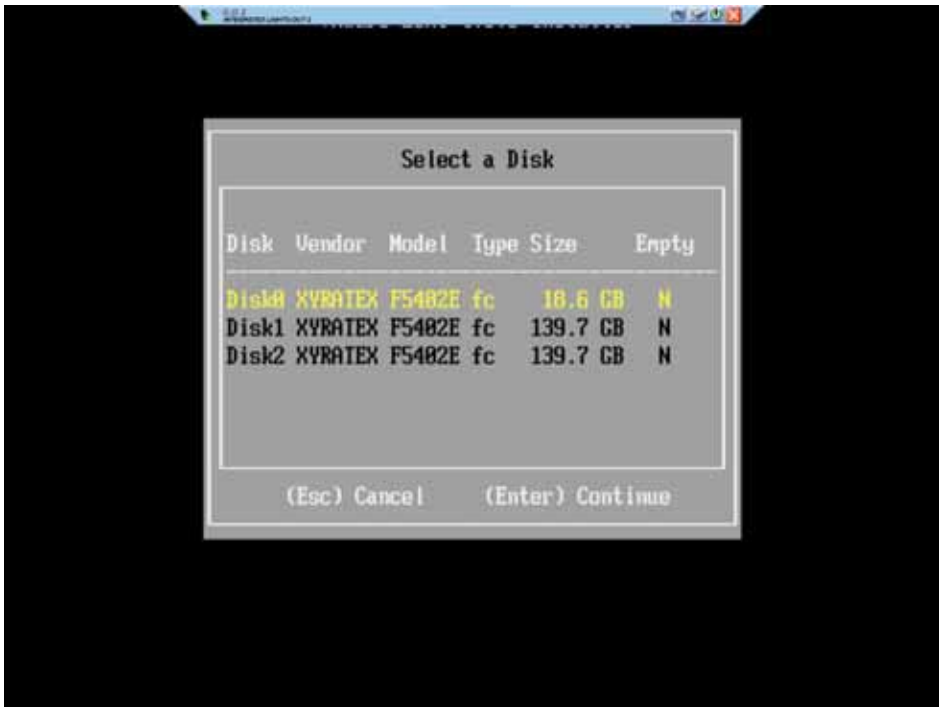


Figure 28 Select a Disk for Booting

Note

Notice that the specific target and LUN IDs are not displayed in the list. Because you made a note of the SAN Boot LUN's size, you should be able to determine which LUN to select as the boot disk.

Step 6 Press ***Enter*** to continue the installer until you see the Confirm Install dialog. See Figure 29 on page 61.

Figure 29 Confirm Install Dialog

Note

Xsigo recommends that the SAN Boot LUN be dedicated to the modified ISO. However, in some cases, this situation might not be possible. If your SAN Boot LUN already contains data, the following dialog will be displayed before the Confirm Install dialog.

If this dialog is displayed, be aware that the data on this LUN will be overwritten. You can either overwrite the existing data, or abort the current installation, store the data on a different LUN, then resume the installation.

Step 7 When the correct boot disk is confirmed, the installation runs to completion. The Installation Complete dialog is displayed. Figure 30 shows this dialog.

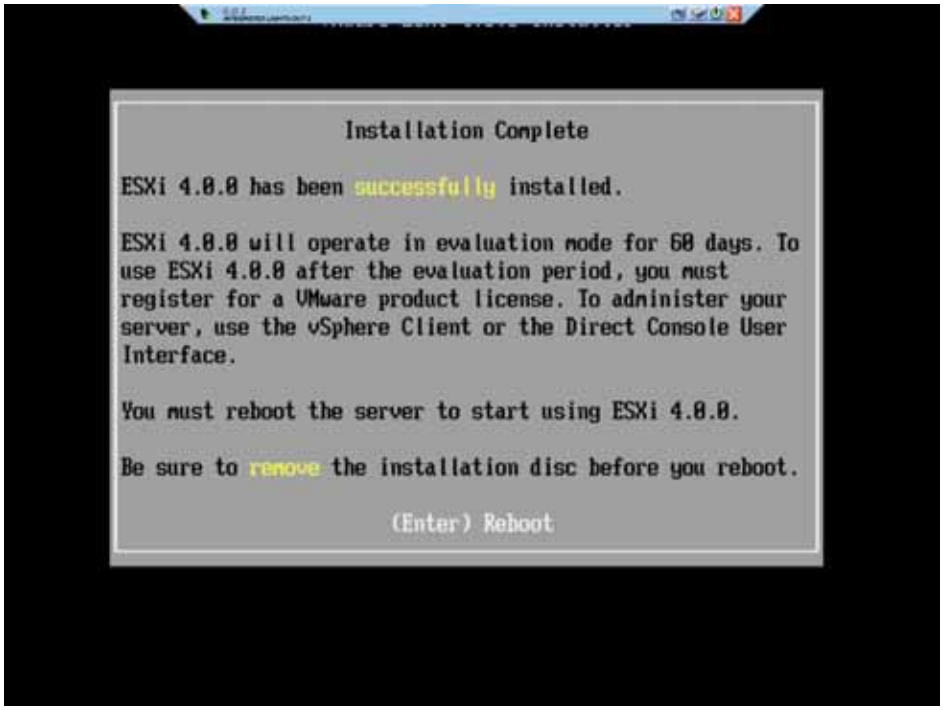


Figure 30 Installation Complete

Step 8 Make sure that any installation medium (CD or DVD) in the ESXi 4.1 server is removed, then allow the server to reboot, as shown in Figure 31 on page 63.

Figure 31 ESXi 4.1 Server Rebooting

Step 9 When the server reboots, it progresses through the boot devices until it locates the SAN Boot vHBA from which it retrieves the SAN Boot image, as shown in Figure 32 on page 64.

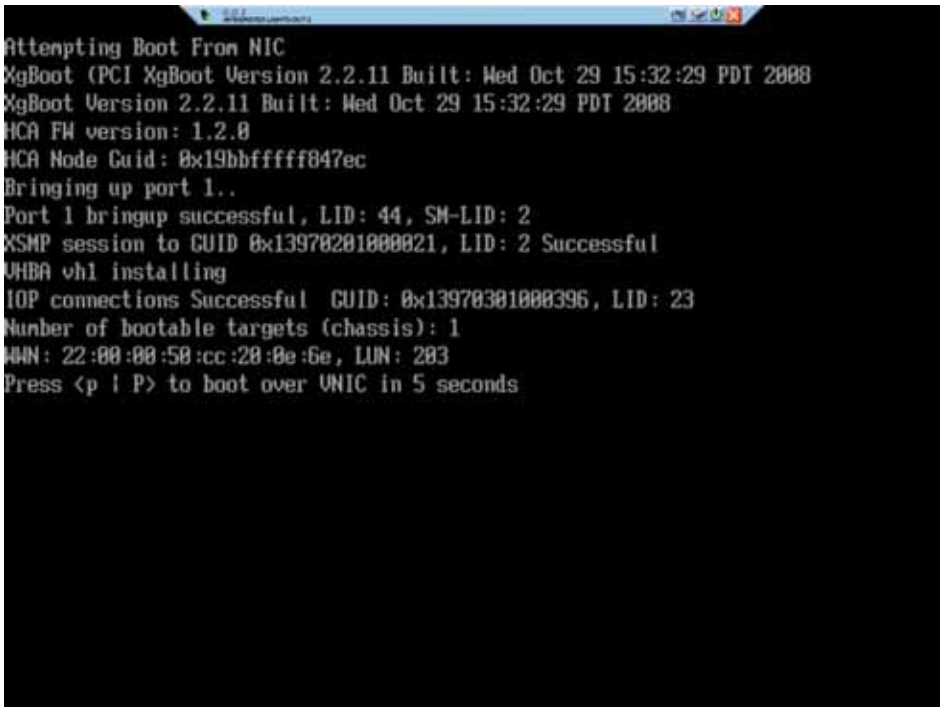


Figure 32 SAN Boot Dialog

A NIC card is attempted first, then the vHBA. The text “VHBA installing” should indicate the name of the vHBA configured for SAN Booting. If this text is incorrect, then you should reattempt this procedure.

After the SAN Boot vHBA is recognized as a boot device, the ESXi 4.1 server completes its boot up. Virtual NICs and virtual HBAs can be configured in the ESX server to provide the benefits of Xsigo virtual I/O.

Oracle's Xsigo Fabric Director supports booting a Red Hat Linux 6u1 server over a vNIC using an iSCSI connection. The remote disk to boot from is identified by a targer iSCSI Qualified Name (IQN) and Logical Unit Number (LUN) on an iSCSI storage disk array. Figure 1 illustrates the topology used to achieve iSCSI booting.

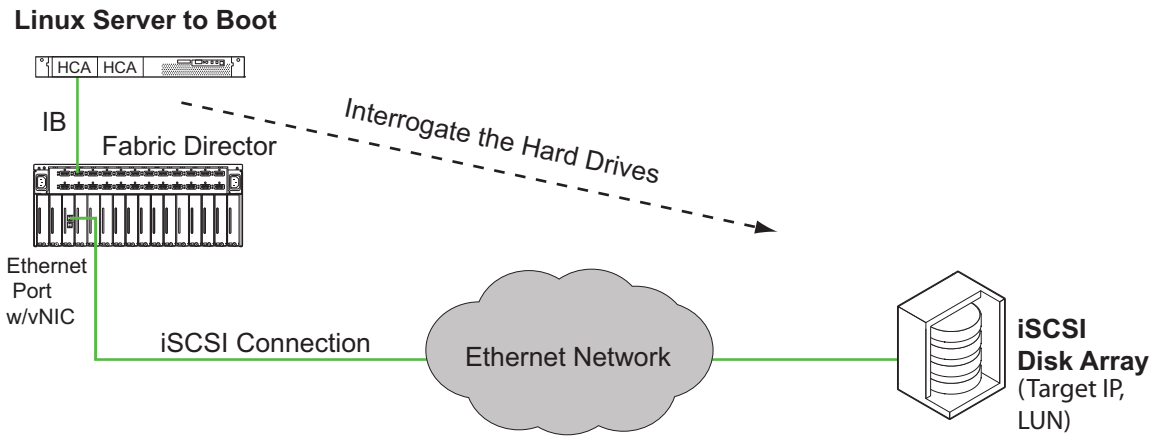


Figure 1 Linux Server iSCSI Boot Topology

This chapter explains how to set up iSCSI booting. It includes the following sections:

- Understanding iSCSI Booting
- Syntax for I/O Resource Configuration
- Creating the iSCSI Boot PXE Image

Understanding iSCSI Booting

When configuring iSCSI boot, you perform the same general steps as for any remote booting setup:

1. Configure the host server with drivers and firmware to enable remote booting.
2. Create the bootable server profile and vNIC.
3. Create the iSCSI LUN, which is where the iSCSI PXE boot image will be installed.
4. Make sure that the PXE Boot server is correctly set up. You will need access to the default config file. This document assumes that your PXE Boot server is already configured. Details about configuring your PXE Boot server are out of the scope of this document.

Terms

Table 2 lists some terminology specific to iSCSI booting.

Table 2 iSCSI Boot Terms

Term	Definition
initiator	The host server that is booting over an iSCSI connection
target	The iSCSI array
IQN	An iSCSI qualified name of an initiator or target
target IP	The IP address of the target filer or array. This is optional if you are using DHCP addressing on the vNIC.

Caveats

- Linux hosts can iSCSI Boot if they are running Red Hat Enterprise Linux (RHEL) 6u1 or later unless otherwise documented.
- To get the initiator IQN, the iSCSI boot profile will need to be created on Oracle's Xsigo Fabric Director and pushed to the server that will be iSCSI booted from. The initiator IQN will not be learned until the server profile is created.
- When installing the iSCSI Boot PXE image onto the iSCSI LUN, only one method is supported. See Using the append Command. Other methods exist, for example CD installs, but are not currently supported by Xsigo.

Syntax for I/O Resource Configuration

```
set server-profile <name> iscsi-boot [<vnic>|none] <targetIP>
set server-profile <name> iscsi-boot [<vnic>|none] <targetIP>
   [mount {direct </dev/node> | LABEL=<label>] | lvm <group-name>
   <volume-name>}]
set server-profile <name> iscsi-boot [<vnic>|none] <targetIP>
   -target-iqn=<targetIqn>
show server-profile <server> iscsi-boot [-detail]
```

Parameter Descriptions

set server-profile <name>—Identifies a named server profile to boot from.

iscsi-boot [<vnic>|**none**]—Creates a boot object and associates it with a named vNIC. The iSCSI boot object can use only the vNICs that are available to the server profile. You must have a previously created the vNIC.

none—Removes the iSCSI boot object from a server profile.

<targetIP>— Specifies the IP address of the storage array or filer. This is not the specific boot volume.

show server-profile <server> **iscsi-boot** [**-detail**]—Displays iSCSI boot information for a server profile.

-target-iqn—Specifies the iSCSI qualified name of the target array. By default, this IQN is not offered automatically, so you will need to set up your DHCP server to offer the IQN. By default For static addressing, always include this modifier.

Optional Modifiers

-lun—logical unit number of the boot volume. The default is LUN 0.

-port—port number on target filer or array. The default is port 3260.

-protocol—transport protocol. Leave this qualifier set to 6, TCP, unless you have a specific reason to change it.

-target-iqn—target iSCSI qualified name. This is optional when using DHCP addressing but required when using static addressing. If you use DHCP, when the initrd starts it runs the DHCP client to get the target's address. If you use static addressing, you must provide the target IQN on the command line.

-target-pg— a standard iSCSI target portal group

Creating the iSCSI Boot PXE Image

With RHEL 6.1 hosts, you will add stanzas to the default config file on the PXE boot server The stanzas will add the Xsigo host drivers. This process occurs through the use of the **append** command. This command allows you to simply add the Xsigo vNIC devices to the RHEL 6.1 OS image when the RHEL 6.1 server is PXE or iSCSI Booting. The **append** command is issued on the PXE Boot Server that booting hosts connect to during their PXE or iSCSI Boot phase, and the command allows you to add the Xsigo-specific information to the default config file.

Using the append Command

The **append** command allows you to place Xsigo devices into the default Red Hat 6.1 or later OS image. This procedure occurs on the PXE server default config file.


Note

The append command is the only method currently supported by Xsigo. Other methods, such as CD ROM installs, are not supported.

To inject the Xsigo host drivers into the RHDD, use the **append initrd** option:

```
append initrd=<initramfs-string> <xsigo-rhdd-image> ksdevice=<device-name>
network
```

where:

- `<initramfs-string>` is the Red Hat OS image and the Xsigo boot driver (RHDD) that you want to insert into the Red Hat OS image separated by a comma only--no blank.
- `<device-name>` specifies the particular interface that you want the installer (for example, eth2) to use for the kickstart process.

To use the **append** command, you will use the base RHEL 6.1 initramfs and the Xsigo RHDD image. For example:

```
append initrd=initrd-rhel6u1-x86_64.img,xsigo-rhdd-2.6.32-
131.0.15.el6.x86_64-3.6.9.LX3-x86_64.img ksdevice=<device-name>
kspath=<path-to-kickstart-file> network
```

Caution

Pay close attention to the syntax, especially the comma (,) separator between the RHEL image and the Xsigo RHDD. There should be no blank spaces between the RHEL image and the Xsigo RHDD--only a comma. If spaces exist, or the comma is not present, the command will fail and the Xsigo host drivers will not be appended to the RHEL image. As a result, the server will not present iSCSI disks to which you want to install.

Example: RHEL 6.1 iSCSI Boot Installation and Configuration for PXE Boot

For Red Hat to install the Xsigo virtual I/O, you will need to add the Xsigo modules into the default initramfs that Red Hat provides by using the **`append initrd`** command to append the Xsigo host drivers to the default intird. The Xsigo host drivers must be in the intird to allow the Red Hat server to recognize the Xsigo devices and attach the correct host drivers. Red Hat Enterpise uses the initrd for install, but afterward, uses the initramfs for subsequent boots.

If the Xsigo bootable vNIC is not in the initramfs, the Linux OS temporarily marks it as an unknown device, and it eventually is marked “not bootable.” When the Xsigo vNIC is marked not bootable, it is bypassed while the Red Hat initrd is loading, and the server will not be able to iSCSI boot from the Xsigo devices.

To use the **`append`** command, you will need the following software:

- Xsigo boot drivers (the RHDD image)
- Stock initrd (provided by Red Hat on your installation media)

Note

To complete this procedure, you will need specify the initiator IQN that is allowed to access the LUN where the boot initramfs is located. To get the initiator IQN, you must create and deploy the ISCSI boot server profile. The initiator IQN will be learned only after the iSCSI boot server profile is created.

To install RHEL6.x over a vNIC to iSCSI storage:

Step 1 On the host server, install an HCA containing Xsigo’s iSCSI Boot option ROM.

Note

Many iSCSI-boot-capable servers exist. However only certain BIOSes support certain HCAs with the Xsigo Option ROM. Consult Xsigo Support for the hardware compatibility matrix.

Step 2 On the PXE Boot servre, download the appropriate `xsigo-boot-<kernel>.tar` file unique to your distribution and architecture.

Step 3 On the PXE Boot Server, untar **`xsigo-boot-`**`<kernel>`**`.tar`** file and copy the xsigo-rhdd-<kernel>.img to the PXE Boot server.

From Linux:

`tar xvf xsigo-boot-``<kernel>`**`.tar`**

Step 4 On the PXE server, append the Xsigo information to the default config file, as documented in Using the append Command.

Step 5 On the Fabric Director, create a server profile for the server performing the RHEL6 install to storage. For the physical connection, use the host's HCA GUID.

You can get the server's GUID from the HCA packaging. Or, the GUID is also displayed during the server's power on self-test (POST), so you can watch for it during the server's POST. You should have the server's GUID available for this part of the procedure.

```
add server-profile pokemon 2c02123a5303
```

For more about assigning a server profile to a server that cannot boot yet, see "Connecting the Server Profile to the Server" on page 3.

Step 6 Set up the server profile to do iSCSI Boot. The **set server-profile** <name> **iscsi-boot** command provides the server with all the information that it needs to connect to storage upon boot up (such as, the vNIC to use for storage, IP address of the storage, and the target IQN).

```
set server-profile <name> iscsi-boot <vnic-to-storage> <target-ip>
<target-iqn>
```

For example:

```
set server-profile pokemon iscsi-boot eth5 192.168.8.108
-target-iqn=iqn.1992-08.com.netapp:sn.118047284
```

After the iSCSI boot server profile is created, you will need the initiator IQN to complete the boot up sequence.

Step 7 Display the details for the iSCSI Boot server profile, and get the i-iqn from the resulting output. Make a note of the i-iqn. You will need to use it to allow the iSCSI Boot sequence to complete.

```
show server-profile pokemon iscsi-boot -detail
-------------------------------------------------------------------------
server    pokemon
vnic      eth5
target    192.168.8.108
i-iqn     iqn.2006-09.com.xsigo:xg.139701f025
t-iqn     iqn.1992-08.com.netapp:sn.118047284
t-pg
lun       0
port      3260
proto     6
mnt-type  direct
lvm-grp
lvm-vol
dev
-------------------------------------------------------------------------
1 record displayed
```

Step 8 As an option, remove all physical hard drives from server. This step is not mandatory.

Step 9 After installation is complete but before rebooting the server, you will need to copy the kernel-IB and Xsigo host drivers to /mnt/sysimage/temp. However, at this point, you will still be in chroot jail, and need to escape.

Step 10 Escape chroot jail by issuing the following command:

```
chroot /mnt/sysimage
```

Step 11 Change directory into /tmp:

```
cd /tmp
```

Step 12 Before rebooting the server, you will need to install the kernel IB RPM:

```
rpm -ivh kernel-ib.rpm
```

Step 13 After the `kernel-ib` rpm is successfully installed, install the Xsigo host driver RPM:

```
rpm -ivh xsigo-host-drivers.rpm
```

Step 14 Reboot the server. During this reboot, the boot sequence will be interrupted and you will be prompted for the i-iqn. Enter it when prompted, then allow the server to boot up.

iSCSI Boot allows you to boot a VMware ESX Server from a LUN on an iSCSI array accessed through a vNIC. The remote disk to boot from is identified by a target iSCSI Qualified Name (IQN) and Logical Unit Number (LUN) on a storage disk array device.

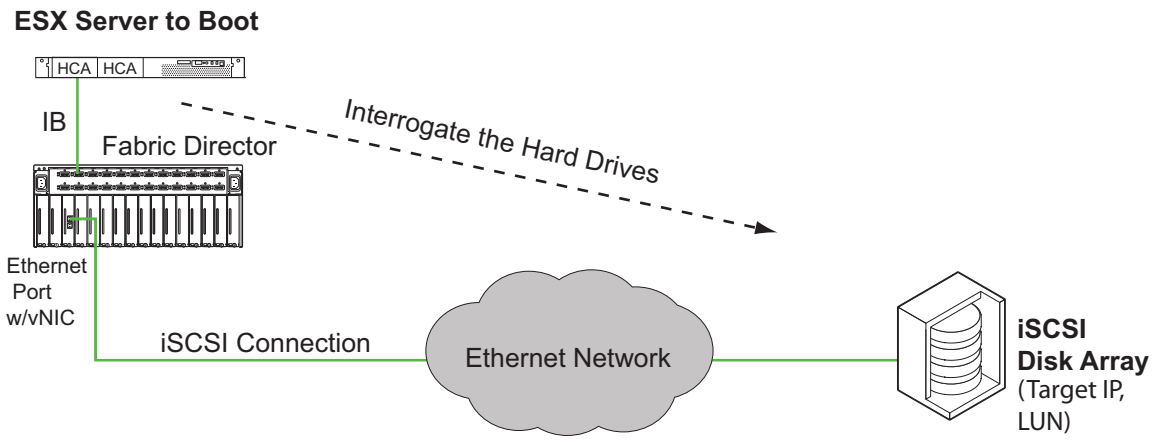


Figure 1 ESX Server iSCSI Boot Topology

This chapter describes iSCSI Boot in the context of the following ESX Server versions:

- ESXi Server 5.0 (and updates).
- ESXi Server 4.1 (and updates). In the following documentation, ESXi 4.1 is referred to as “ESXi 4.1.” Some screen captures might show ESX 4.0, but the procedures are appicable to ESX 4.1.
- ESX Server Classic 4.1 (and updates). In the following documentation, ESX Classic 4.1 is referred to as “ESX 4.1.” Some screen shots might show ESX 4.0, but the procedures are applicable to ESX 4.1.

This chapter contains the following sections:

Configuring ESXi 5.0 Server iSCSI Boot

Configuring iSCSI Boot for VMware ESXi Server 5.0 systems is conceptually similar to configuring an ESXi 4.1 system, with the exception of creating a modified ISO image that includes the Xsigo host drivers. iSCSI Boot is supported for ESXi 5.0 servers through a procedure that has the following steps:

- Creating a modified ISO image for booting.
- Creating the iSCSI Boot server profile with a vNIC that can reach the target/LUN where the ISO image is installed.
- Installing the image on the ESXi 5.0 server.


Note

To complete this procedure, you will need some additional utilities (for example, Microsoft PowerShell) for a typical default install.

Prerequisites

Before installing the new Xsigo host drivers, make sure that any previous version of host driver is removed. For example, if you are upgrading your ESX server from 4.1 to 5.0, you will need to uninstall the 4.1 host drivers.

```
esxupdate -b <bundle-id> remove
```

Creating a Modified ISO Image

To have the Xsigo vNICs and vNICs available to the ESXi 5.0 OS for PXE or iSCSI Booting, you will need to inject the Xsigo host drivers into the native ESX OS. This procedure documents how to inject the Xsigo devices into the ESXi 5.0 bundle. You will basically use an existing ESXi 5.0 bundle, use a downloadable tool to inject the Xsigo host drivers into the ESX OS, then use another downloadable tool to repackage the modified ISO.

Before creating a modified ISO image for ESXi 5.0 hosts, be aware of the following:

- Creating the custom ISO is accomplished through Microsoft Windows PowerShell—and specifically the VMware vSphere PowerCLI plug-in for PowerShell. The Windows server will need this tool installed. Make sure the Windows server has the correct requirements to run PowerShell and PowerCLI.
- Creating the custom ISO is supported on a Windows host server only. The server requirements are determined by the PowerShell application.
- You use a pre-configured ESXi bundle as a baseline, then inject the Xsigo bits into it:
  - For ESXi 5.0 Update 0 (GA), the ESXi bundle is `VMware-ESXi-5.0.0-469512-depot.zip` and is available from VMware's website.
  - For ESXi 5.0 Update 1, the ESXi bundle is `VMware-ESXi-5.0.0-623860-depot.zip`
- You will need full administrative rights on the Windows server where you will be creating the custom ISO.

The following procedure assumes the working directory is `\images\New` for the user "adminA". The procedure also uses the VMware 5.0 GA bundle (`469512`) for illustrative purposes. If your hosts are running ESXi 5.0 Update 1, you will need to use the bundle with build number `623860`.

To create the modified ISO for ESXi 5.0 hosts, follow this procedure:

Step 1 Install PowerShell on the Windows server if you have not done so already.

Step 2 Install the PowerCLI plug-in if you have not done so already.

Step 3 Download the `VMware-ESXi-5.0.0-469512-depot.zip` file to the Windows server.

Step 4 Start PowerCLI.

Step 5 In PowerCLI, run the following commands to import the ESXi 5.0 bundle and the Xsigo host drivers into PowerCLI:

```
Add-EsxSoftwareDepot -DepotUrl C:\Users\adminA\Desktop\images\New\VMware-ESXi-5.0.0-469512-depot.zip

Add-EsxSoftwareDepot -DepotUrl C:\Users\adminA\Desktop\images\New\xsigo_5.0.1ESX.1-1vmw.500.0.0.406165.zip
```

Step 6 Run the following commands to specify the file that you want to use when creating the output ISO. The profile determines metadata about the output ISO, such as formatting, compression method, and so on. In this example, the profile is named `ESXi-5.0.0-469512-standard-xsigo` for illustrative purposes.

```
New-EsxImageProfile -CloneProfile ESXi-5.0.0-469512-standard -name "ESXi-5.0.0-469512-standard-xsigo"
```


Note

Notice that the **`-name`** string supplied for the profile is enclosed in quotation marks. The quotation marks are required syntax for the profile's name.

Step 7 Run the following commands to add the IB stack and other dependencies to the depot.

```
Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage net-ib-core

Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage net-mlx4-core

Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage net-ib-mad

Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage net-ib-sa

Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage net-ib-ipoib

Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage net-mlx4-ib

Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage net-xscore
```

```
Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage net-xsvnic

Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage net-xve

Add-EsxSoftwarePackage -ImageProfile ESXi-5.0.0-469512-standard-xsigo
                       -SoftwarePackage scsi-xsvhba
```

Step 8 Run the following command to create single output ISO containing all required files from the depot. The following example assumes unsigned drivers to provide the most complete example.

```
Export-EsxImageProfile -ImageProfile ESXi-5.0.0-469512-standard-xsigo -ExportToIso
-FilePath C:\Users\adminA\Desktop\images\New\VMware-VMvisor-Installer-5.0.0-
469512_Xsigo.x86_64.iso -NoSignatureCheck
```

Note

Xsigo makes every effort to release signed, certified host drivers. However, on some occasions, unsigned drivers might be released. If you receive unsigned Xsigo host drivers, the **`Export-EsxImageProfile`** command has the **`-NoSignatureCheck`** option which will bypass signature checking.

Use the `-NoSignatureCheck` for unsigned drivers.

Omit the `-NoSignatureCheck` option if the drivers are signed.

Creating a iSCSI Boot Server Profile

If you have not already created a iSCSI Boot server profile, you must do so. The iSCSI Boot Server profile must have at least one vNIC that is connected to the storage where the ESXi 5.0 boot image will reside.

Step 1 Create the server profile. For example, to create the server profile “esx50” for “server10” which is connected through IB port 23 on the Oracle Fabric Director “tuffy:

```
add server profile esx50 server10@tuffy:ServerPort23
```

Step 2 Create the vNIC for the iSCSI Boot Server Profile. For example, to create a vNIC named “vNIC1” in server profile “esx50” and have the vNIC terminated on the fibre channel port 1 in slot 8:

```
add vnic vnic1.esx50 8/1 -boot-capable=true
```

Step 3 Set the Server Profile for iSCSI Booting and connected to the WWN of the target where the boot image will reside:

```
set server-profile esx50 iscsi-boot vnic1
                         -target-iqn=iqn.1992-
                         08.com.netapp:sn.11804728455 -lun=203
```

Step 4 Verify that the server profile is up and connected as shown.

```
show server-profile esx50 iscsi-boot

server    role         vNIC    mnt-type   lvm-grp    lvm-vol         dev        mnt-opts disks
-------------------------------------------------------------------------------------------
esx50     loadmount    vnic1                                 22:00:00:50:CC:20:0E:6E (203/LM)
```

Note

Make a note of the size of the LUN on which the boot image will reside. You will be prompted to select the correct LUN when you load the host drivers onto the ESXi 5.0 server, and knowing the LUN's size will help you select it from the list of connected storage targets.

With the modified ISO ready and the iSCSI Boot server profile created, you will need to load the ISO onto the ESXi 5.0 server and reboot the server so that the iSCSI Boot vNIC is recognized as the primary boot option for the ESXi 5.0 server.

Set the Server HCA High in the Boot Order

In order to iSCSI Boot, the server uses the Xsigo Option ROM on HCA. To correctly boot the server, you must set the HCA Option ROM higher in the boot order than the virtual CD. In order to do so,:

Step 1 Enter the server's BIOS.

Step 2 Select the Option ROM in the boot devices list.

Step 3 Move the Option ROM higher than the virtual CD device in the boot order.

Step 4 Exit the server's BIOS.

Setting the server to boot through the HCA/Option ROM is a requirement to correctly populate the iSCSI Boot Firmware Table (iBFT) with the disks to install the boot image on.

When the HCA is set high in the boot order, proceed to the next section.

Loading the Image into the ESXi 5.0 Server

When the modified ISO image has been created and put on a network reachable device, and a iSCSI Boot server profile has been created, you can now load the ISO image into the ESXi 5.0 server, and boot the server. When it boots, you will have an option to select the LUN that contains the iSCSI Boot ISO.

Note

The following procedure assumes the installation of the iSCSI Boot image and configuration of the iSCSI Boot feature through an ILO or DRAC.

To load the iSCSI Boot image into the ESXi 5.0 server, follow this procedure:

Step 1 From the ESXi 5.0 server, locate and mount the modified ISO (`ESXi-5.0.0-Standard-Xsigo Installer`) as shown in Figure 2 on page 76.

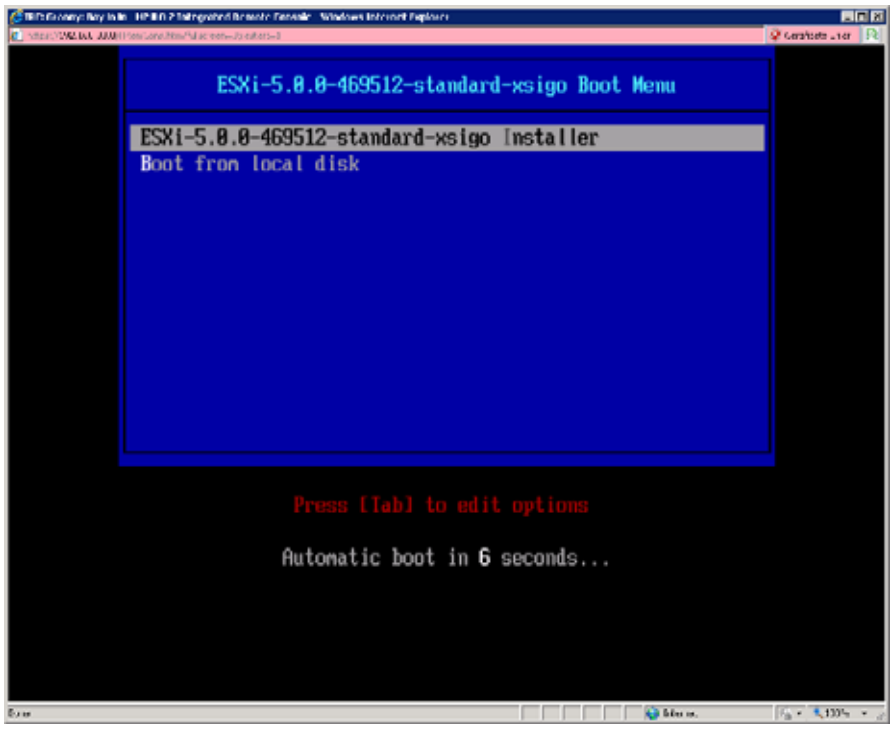


Figure 2 Selecting the Xsigo ESXi 5.0.0 Installer

When the `ESXi-5.0.0-standard-xsigo Installer` is mounted, files are loaded onto the server. You will see output similar to Figure 3 on page 77.

Note

Be aware that loading files onto the server can take a long time. You must wait for it to complete.

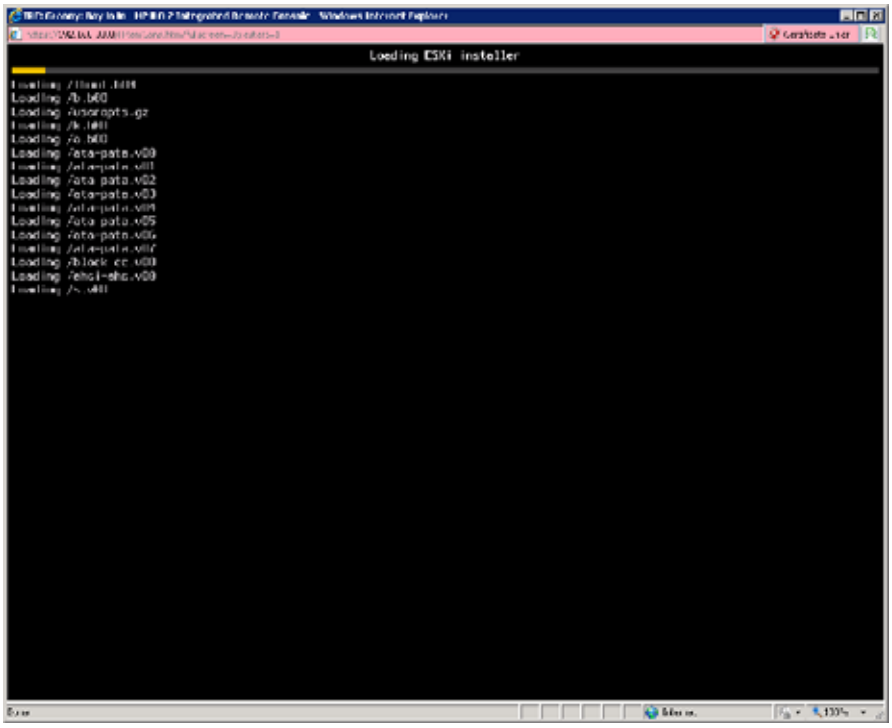


Figure 3 Loading OS Files from Modified ISO Image

Step 2 After the OS files have been loaded, the server boots to the ESXi 5.0 native operating system. See Figure 4 on page 78.

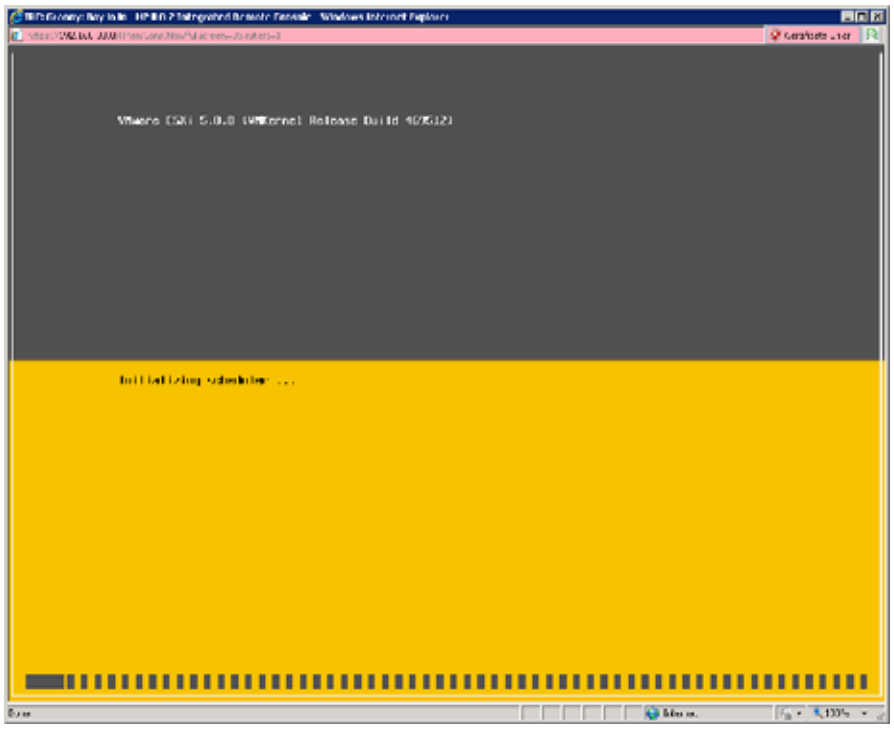


Figure 4 Boot Up

Step 3 As part of the boot sequence, you must acknowledge license agreement. See Figure 5 on page 79.

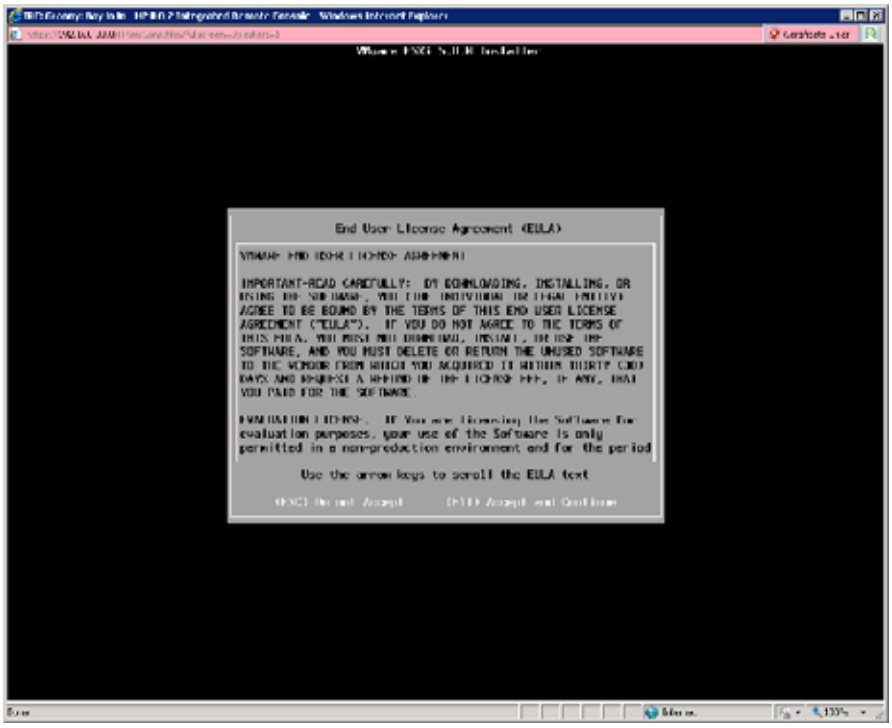


Figure 5 End User License Agreement for ESXi 5.0

Step 4 Accept (***F11)*** the license agreement (or decline by pressing ***Esc***) as needed. To continue the installation procedure, accept the license agreement and start the ESXi 5.0 installer. Figure 6 on page 80 shows the ESXi 5.0 installer Welcome screen.

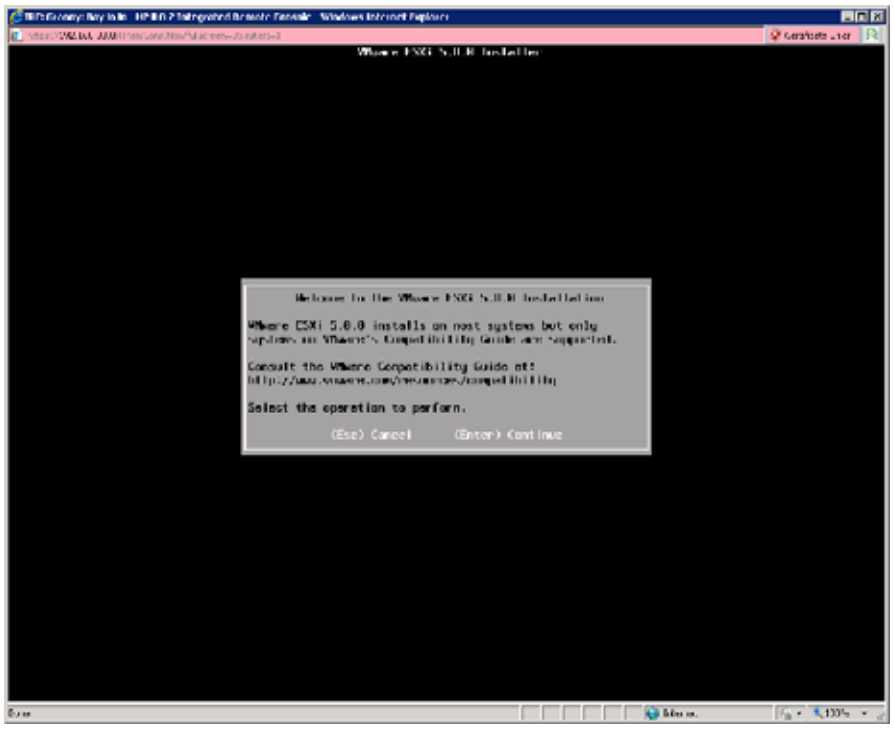


Figure 6 ESXi 5.0 Installer

Step 5 Press ***Enter*** to continue the installer.

Follow the installer until you are prompted to specify a boot disk on the Select a Disk to Install or Upgrade dialog.

Step 6 On the Select a Disk to Install or Upgrade dialog, select the LUN on which the iSCSI Boot vNIC is connected. Figure 7 on page 81 show an example of this dialog.

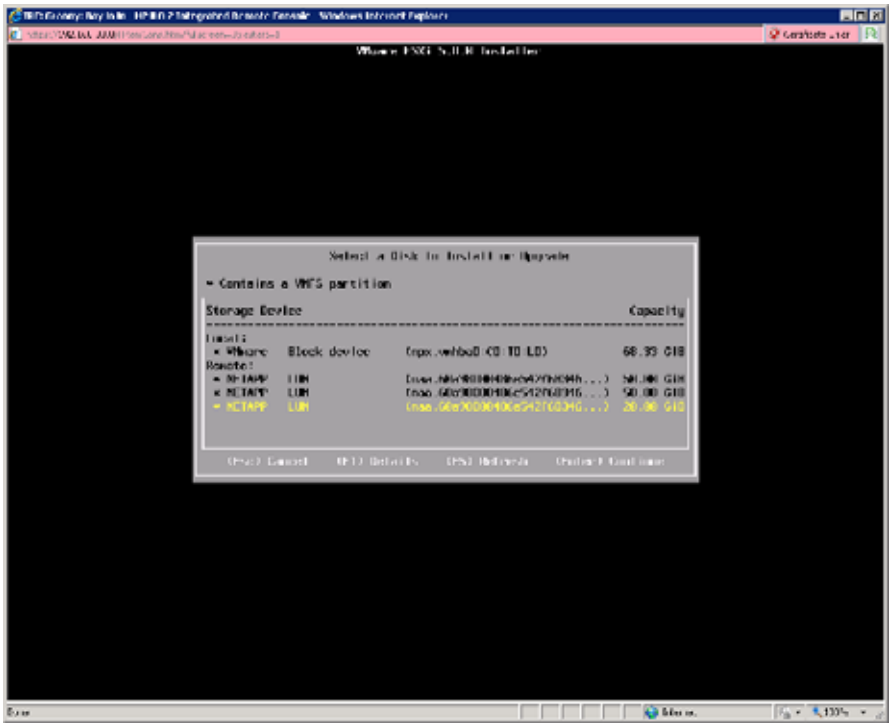


Figure 7 Select a Disk for Booting

Note

Notice that you can get details about a LUN (if desired) by pressing ***F1***.

Step 7 Press ***Enter*** to continue the installer until you see the Confirm Install dialog.

If the LUN has already had ESXi 5.0 installed, a VMware file system will be labelled on the LUN. The installer will prompt you to determine how you want to proceed, as shown in Figure 8 on page 82.

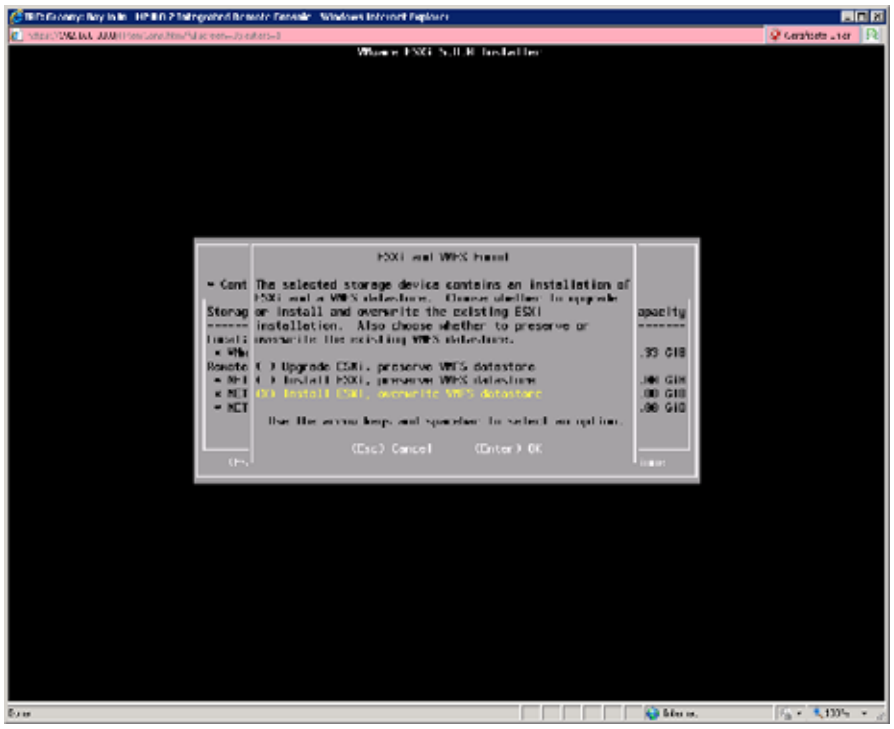


Figure 8 Format VMFS Dialog

Step 8 On the VMFS Format dialog, select the option as needed for your installation:

- Upgrade ESXi, preserve VMFS datastore.
- Install ESXi, preserve VMFS datastore
- Install ESXi, overwrite VMFS datastore. This is required for a fresh install.

Note

Additional dialogs for the keyboard type and language are displayed. Make sure to select the correct option for your ESX installation.

Step 9 When prompted, log in to the ESXi 5.0 server. You will need to log in to the server to complete the installation. Figure 9 on page 83 shows the log in prompt.

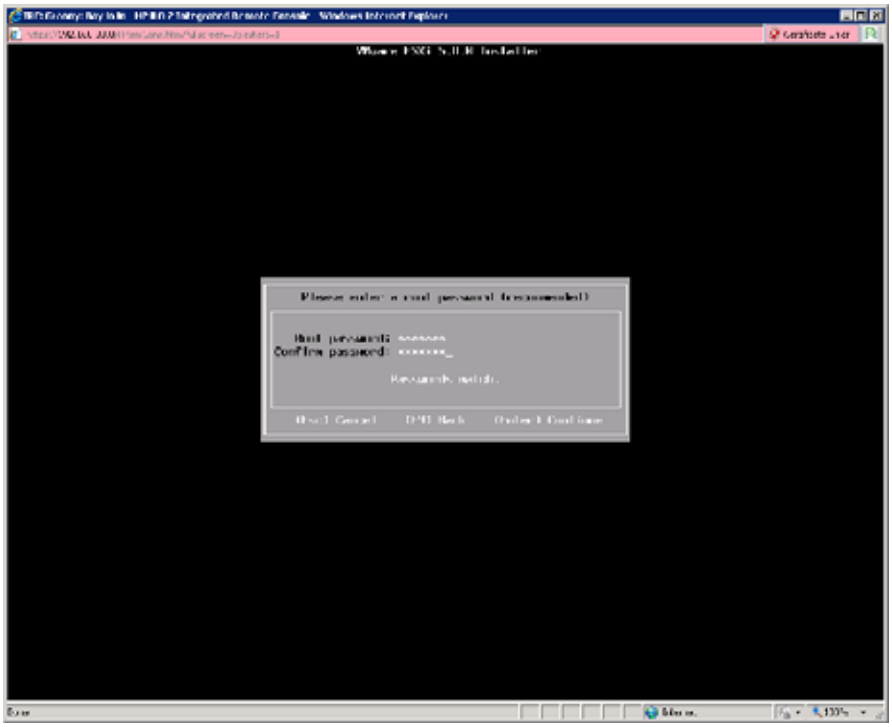


Figure 9 Log In to ESXi 5.0 Server

After you log in, the installation runs to completion. The Confirm Installation dialog is displayed. Figure 10 shows this dialog.

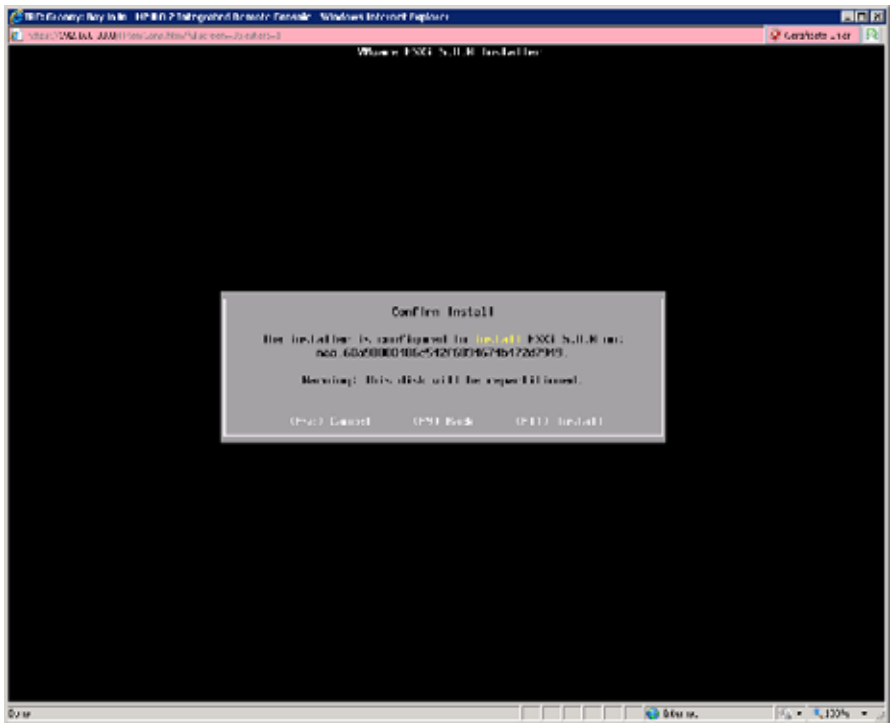


Figure 10 Confirm Install

Step 10 Press ***F11*** to confirm that you want to continue the install. When you press ***F11***, the OS and Xsigo host drivers are loaded onto the ESXi 5.0 Server, as shown in Figure 11 on page 85.

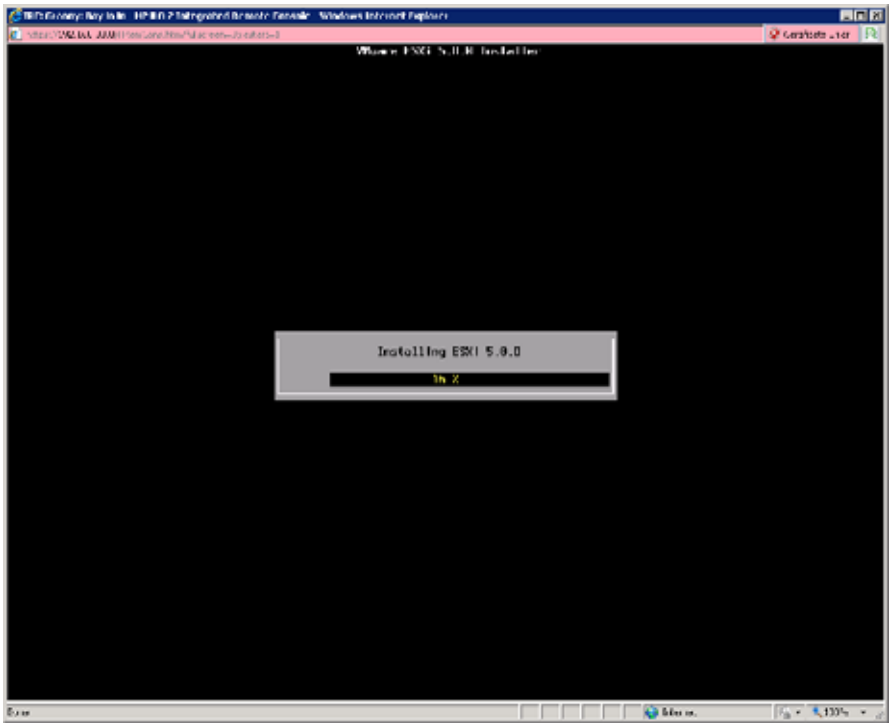


Figure 11 ESXi 5.0 OS and Drivers Loading

Step 11 When the OS and host drivers are loaded onto the ESXi 5.0 Server, you are prompted to finalize the installation, as shown in figure Figure 12 on page 86.

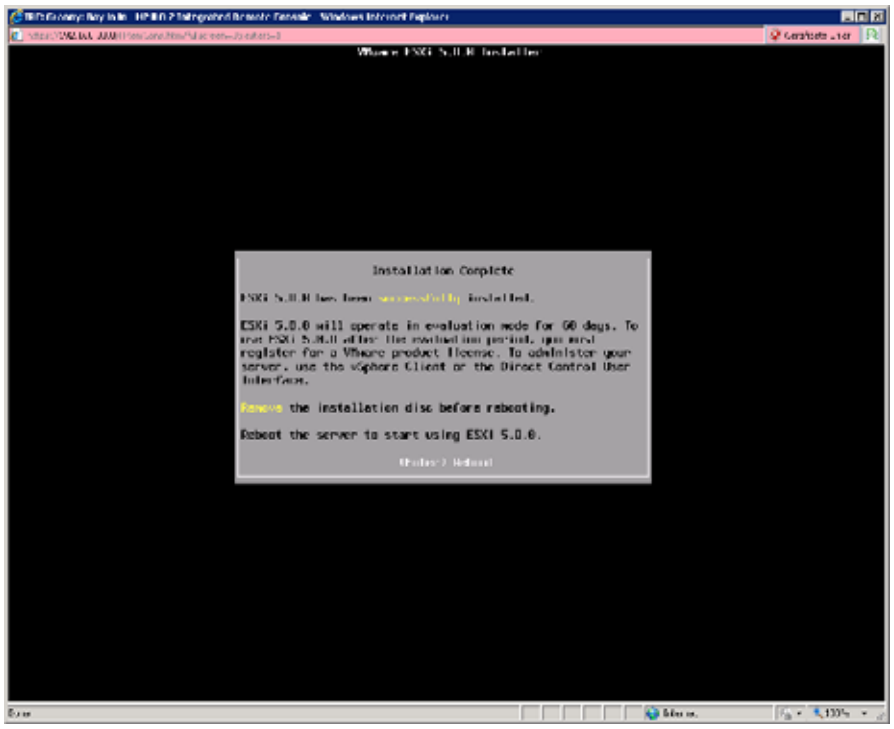


Figure 12 Finalizing the Install

As part of finalizing the OS and host driver installation, the server must be rebooted to load the new OS and host drivers into memory.

Step 12 Press ***Enter*** to reboot the server. When the server reboots, it progresses through the boot devices until it locates the iSCSI Boot vNIC from which it retrieves the iSCSI Boot image, as shown in Figure 13 on page 87.

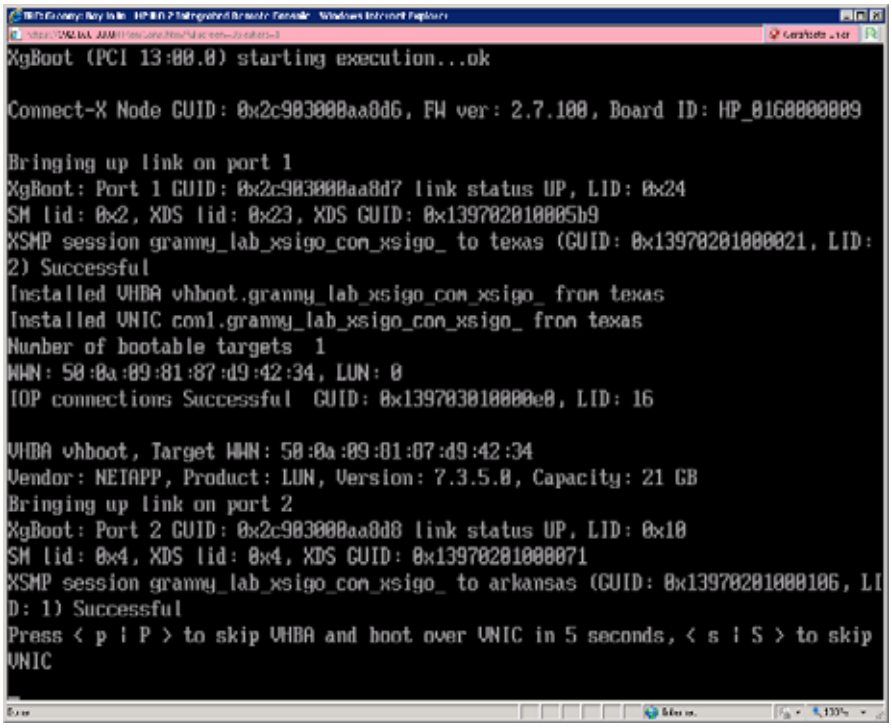


Figure 13 iSCSI Booting from the iSCSI Boot vNIC

Note

The first boot, the vHBA is attempted by default. When no OS is recognized on the vHBA, the server will skip the vHBA and try the vNIC. When the vNIC is used, the server logs in to the iSCSI aerate and gets the IQN information it needs. Because the vNIC has an OS available to boot from (on the iSCSI LUN connected to the vNIC), that OS is used and the server boots over the vNIC. This procedure occurs each time the server is iSCSI booted.

Also, when booting from a vNIC, the login message while the vNIC is logging in to the iSCSI array is displayed on screen so rapidly that you might not recognize that the server is booting from vNIC. Also, no IQN or other recognizable iSCSI information is displayed on the screen. In Figure 13, the server is booting over vNIC even though the vHBA information (for example, WWN) is displayed.

After the iSCSI Boot vNIC is recognized as a boot device, the ESXi 5.0 server completes its boot up. Virtual NICs and virtual HBAs can be configured in the ESX server to provide the benefits of Xsigo virtual I/O.

Configuring ESXi 4.1 Server iSCSI Boot

iSCSI Boot is supported for the ESXi 4.1 servers through a procedure that has the following steps:

- Creating a modified ISO image for booting.
- Creating the iSCSI Boot server profile with a vNIC that can reach the target/LUN where the ISO image is installed.
- Installing the image on the ESXi 4.1 server.


Note

To complete this procedure, you will need some additional utilities (for example, mkisofs and tar/gzip/etc) for a typical default install.

Creating a Modified ISO Image

For this procedure, you will need both of the following:

- `VMware-VMVisor-Installer-4.1.0-171294.x86_64.iso` . Xsigo does not provide the VMware installation medium, so it is your responsibility to obtain it.
- `xsigo-esx-driver-disk-4.1.0.260247.3.5.0-14.iso`. This bundle contains a shell script (`xsigo-add-drivers-esx41i.sh`) that you will use to insert the Xsigo host drivers for ESXi 4.1 into the ISO image. Xsigo does provide this utility. You can download it from the Xsigo Support Portal as part of the host drivers bundle.

You will need to inject the Xsigo host drivers into the ESXi 4.1 bundle by decompressing the overall bundle and inserting the Xsigo host drivers, then recompressing all files into the ISO image. This modified image will be used as the boot image.

To create the modified ISO image, you will need:

- to be logged in to any Linux system with root privileges
- execute privileges on the ISO
- execute privileges on whatever directory you will use to uncompress and recompress the modified ISO.
- You will be using the `xsigo-add-drivers-esx41i.sh` shell script which has the following syntax:

```
xsigo-add-drivers-esx41i.sh --esxiso --driveriso --extrarpm --output -d
--isolinuxcfg --ksconfig --hca-installer-hooks
```

To create the modified ISO image, follow this procedure:

Step 1 Get the VMware installation medium onto a Linux host and place it into a working directory. For illustrative purposes, `opt/` is shown, but you can use whatever directory you want.

Step 2 Locate the `xsigo-4.1.0.260247.esx41i.tgz` and uncompress it to the same directory. For illustrative purposes, `opt/` is shown, but you can use whatever directory you want:

```
tar -zxvf xsigo-4.1.0.164009.3.5.0.esx41i.tgz opt/
```

Step 3 After the bundle is unzipped, run the `xsigo-add-drivers` script and recompress the bundle. This step will take a few minutes, but progress messages will be displayed to indicate the individual stages in the overall job.

```
sudo sh opt/xsigo/contrib/xsigo-remaster-esx41i-iso.sh --xgfile
xsigo-4.1.0.164009.2.2.0.esx41i.tgz --iso VMware-VMvisor-Installer-
4.1.0-171294.x86_64.iso

Copying base ISO VMware-VMvisor-Installer-4.1.0-260247.x86_64.iso

Unpacking ISO

Modifying the base installer image

Repacking installer image

Adding Xsigo-drivers to the base CD

Make ISO Image XG-VMware-VMvisor-Installer-4.1.0-260247.x86_64.iso

sudo sh remaster-esx41i-iso.sh --iso  --xgfile  -d  152.70s user 14.97s
system 85% cpu 3:16.47 total
```

When the command completes, the new ISO image is created as XG-VMware-Visor-Installer. You will use this image to iSCSI Boot the ESXi 4.1 Server.

Creating a iSCSI Boot Server Profile

If you have not already created a iSCSI Boot server profile, you must do so. The iSCSI Boot Server profile must have only one vNIC that is connected to the storage where the ESXi 4.1 boot image will reside.

Step 1 Create the server profile. For example, to create the server profile “esx41i” for “server10” which is connected through IB port 23 on Oracle Fabric Director “tuffy:

```
add server profile esx41i server10@tuffy:ServerPort23
```

Step 2 Create the vNIC for the iSCSI Boot Server Profile. For example, to create a vNIC named “vNIC1” in server profile “esx41i” and have the vNIC terminated on the fibre channel port 1 in slot 8:

```
add vnic vnic1.esx41i 8/1 -boot-capable=true
```

Step 3 Set the Server Profile for iSCSI Booting and connected to the WWN of the target where the boot image will reside:

```
set server-profile esx41i iscsi-boot vNIC1
                          -target-ign=iqn.1992-08.com.netapp:sn.118047284
                          -lun=203
```

Step 4 Verify that the server profile is up and connected as shown.

```
show server-profile esx41i iscsi-boot

server    role         vNIC    mnt-type   lvm-grp    lvm-vol   dev    mnt-opts   disks
-------------------------------------------------------------------------------------
esx41i    loadmount    vnic1  static         iqn.1992-08.com.netapp:sn.118047284 (203/LM)
```

Note

Make a note of the size of the LUN on which the boot image will reside. You will be prompted to select the correct LUN when you load the host drivers onto the ESXi 4.1 server, and knowing the LUN's size will help you select it from the list of connected storage targets.

With the modified ISO ready and the iSCSI Boot server profile created, you will need to load the ISO onto the ESXi 4.1 server and reboot the server so that the iSCSI Boot vNIC is recognized as the primary boot option for the ESXi 4.1 server.

Loading the Image into the ESXi 4.1 Server

When the modified ISO image has been created and put on a network reachable device, and a iSCSI Boot server profile has been created, you can now load the ISO image into the ESXi 4.1 server, and boot the server. When it boots, you will have an option to select the LUN that contains the iSCSI Boot ISO.

Note

The following procedure assumes the installation of the iSCSI Boot image and configuration of the iSCSI Boot feature through an ILO or DRAC.

To load the iSCSI Boot image into the ESXi 4.1 server, follow this procedure:

Step 1 From the ESXi 4.1 server, locate and mount the modified ISO (`XG-VMware-Visor-Installer`). When the ISO is mounted, ESXi 4.1 begins loading all pertinent OS files, as shown in Figure 14 on page 91.

Note

Be aware that this step can take a long time. You must wait for this step to complete.

When the `XG-VMware-Visor-Installer` is mounted, you will see output similar to the following.

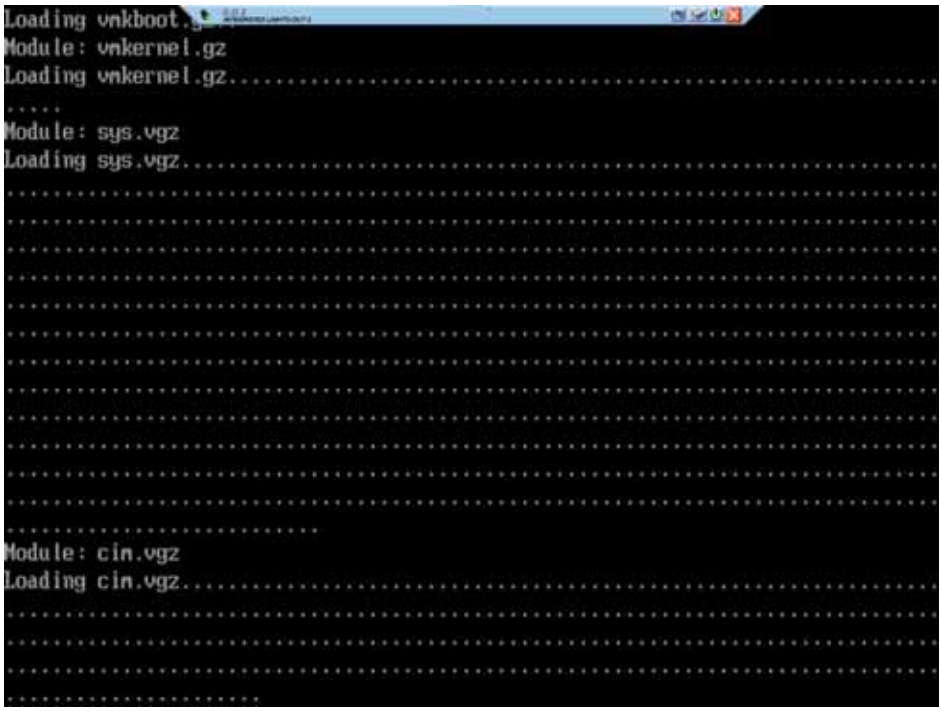


Figure 14 Loading OS Files from Modified ISO Image

Step 2 After the OS files have been loaded, the server boots to the ESXi 4.1 installer. See Figure 15 on page 92.

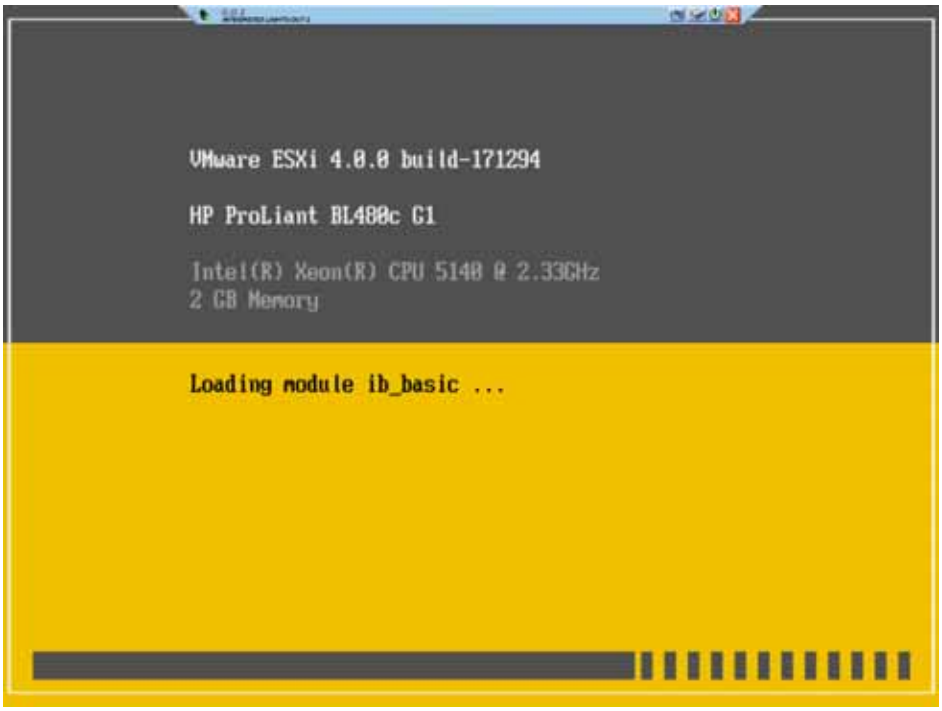


Figure 15 Boot Up

Step 3 As part of the boot sequence, you must read the license agreement.

Step 4 Accept (or Decline) the license agreement as needed. To continue the installation procedure, accept the license agreement and start the ESXi 4.1 installer. Figure 16 on page 93 shows the ESXi 4.1 installer.

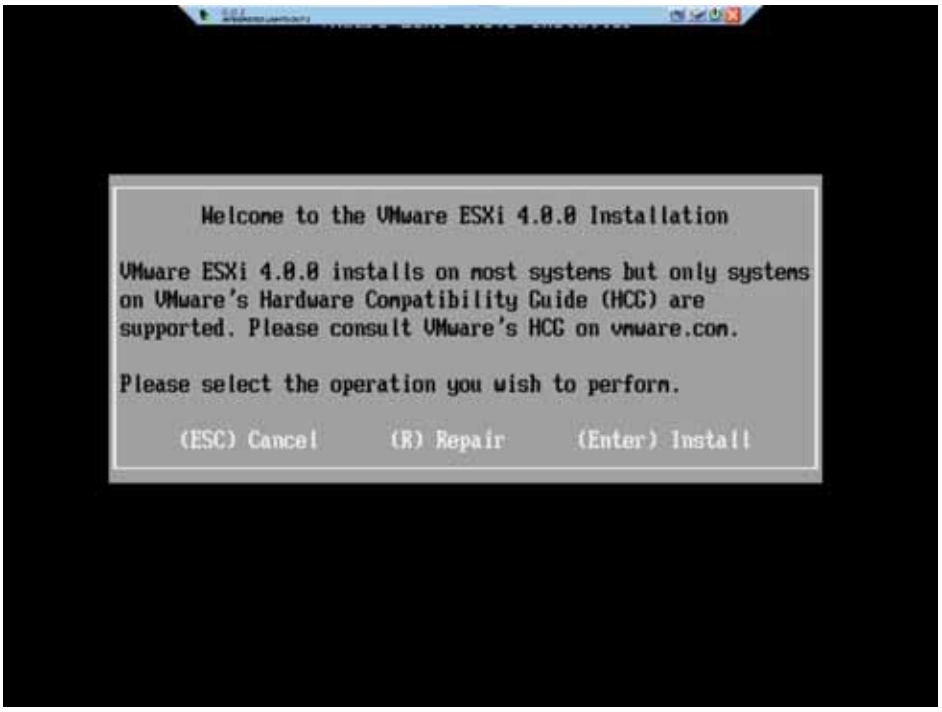


Figure 16 ESXi 4.1 Installer

Follow the installer until you are prompted to specify a boot disk on the Select a Disk dialog.

Step 5 On the Select a Disk dialog, select the LUN on which the iSCSI Boot vNIC is connected. Figure 17 on page 94 show an example of this dialog.

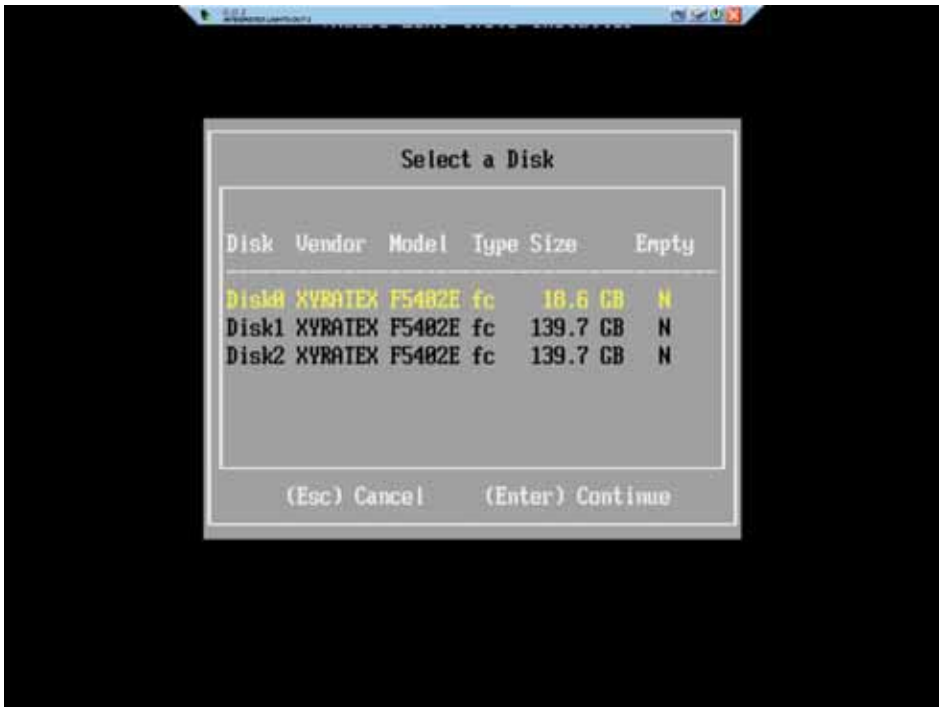


Figure 17 Select a Disk for Booting

Note

Notice that the specific target and LUN IDs are not displayed in the list. Because you made a note of the iSCSI Boot LUN's size, you should be able to determine which LUN to select as the boot disk.

Step 6 Press ***Enter*** to continue the installer until you see the Confirm Install dialog. See Figure 18 on page 95.

Figure 18 Confirm Install Dialog

Note

Xsigo recommends that the iSCSI Boot LUN be dedicated to the modified ISO. However, in some cases, this situation might not be possible. If your iSCSI Boot LUN already contains data, the following dialog will be displayed before the Confirm Install dialog.

If this dialog is displayed, be aware that the data on this LUN will be overwritten. You can either overwrite the existing data, or abort the current installation, store the data on a different LUN, then resume the installation.

Step 7 When the correct boot disk is confirmed, the installation runs to completion. The Installation Complete dialog is displayed. Figure 19 shows this dialog.

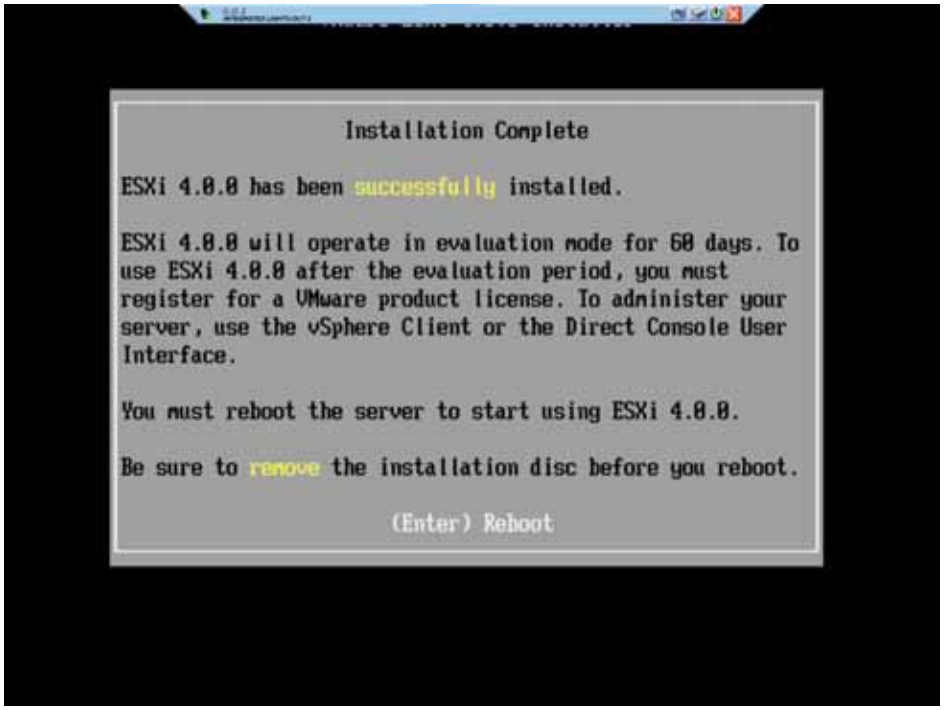


Figure 19 Installation Complete

Step 8 Make sure that any installation medium (CD or DVD) in the ESXi 4.1 server is removed, then allow the server to reboot, as shown in Figure 20 on page 97.

Figure 20 ESXi 4.1 Server Rebooting

Step 9 When the server reboots, it progresses through the boot devices until it locates the iSCSI Boot vNIC from which it retrieves the iSCSI Boot image, as shown in Figure 21 on page 98.

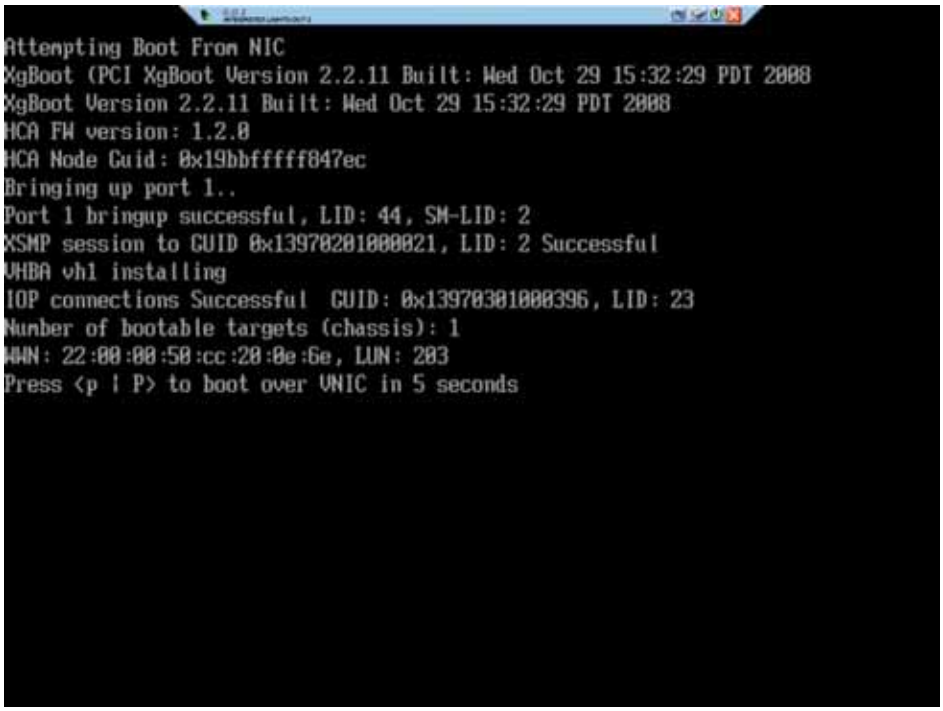


Figure 21 iSCSI Boot Dialog

Note

The first boot, the vHBA is attempted by default. When no OS is recognized on the vHBA, the server will skip the vHBA and try the vNIC. When the vNIC is used, the server logs in to the iSCSI array and gets the IQN information it needs. Because the vNIC has an OS available to boot from (on the iSCSI LUN connected to the vNIC), that OS is used and the server boots over the vNIC. This procedure occurs each time the server is iSCSI booted.

Also, when booting from a vNIC, the login message while the vNIC is logging in to the iSCSI array is displayed on screen so rapidly that you might not recognize that the server is booting from vNIC. Also, no IQN or other recognizable iSCSI information is displayed on the screen.

After the iSCSI Boot vNIC is recognized as a boot device, the ESXi 4.1 server completes its boot up. Virtual NICs and virtual HBAs can be configured in the ESX server to provide the benefits of Xsigo virtual I/O.

Configuring ESX Classic 4.1 iSCSI Boot

iSCSI Boot is supported for ESX Classic 4.1. The iSCSI Boot configuration for ESX 4.1 servers occurs primarily through the ESX 4.1 Installer, with a minor amount of custom configuration for the Xsigo ESX host driver and InfiniBand stack.

The configuration procedure for ESX 4.1 Server iSCSI Boot has the following main parts:

- Create a Server Profile for iSCSI Boot including at least one vNIC that will connect to the LUN where the iSCSI Boot image will be kept.
- Have one LUN available for the install.
- Insert the latest ESX 4.1 install CD and follow the prompts until prompted with the Add Custom Drivers page.
- Insert the Xsigo Host Drivers ISO, and load the drivers.
- Enter the debug shell, and run the Xsigo `install-load`. This is an option for the **`esxcfg-xgutil`** command (`/tmp/drivers/user/sbin/esxcfg-xgutil install-load`).
- Return to the ESX 4.1 Installer and complete the install.
- Make sure to reboot the server.

Before configuring ESX 4.1 iSCSI Boot, be aware of the following considerations:

- If a VMFS exists on the ESX Server 4.1 that will be configured for iSCSI Boot, that VMFS must be deleted from the ESX Server before configuring the ESX 4.1 iSCSI Boot. The process of configuring iSCSI Boot for ESX 4.1 creates a separate instance of VMFS, and sometimes does not remove any existing VMFS on the ESX server. For example, if you are upgrading from to a later update of ESX 4.1 iSCSI Boot, or if you are re-installing on an ESX 4.1 Server and configuring iSCSI Boot, you must remove any existing VMFS before beginning the ESX 4.1 iSCSI Boot configuration procedure.
- While up to 4 LUNs have been tested, Xsigo recommends that you make only one LUN available to the vNIC in the iSCSI Boot Server Profile.
- Also, be aware that the ESX installer does not allow for reformatting any unknown partitions, so you might need to clear entries from the partition table.


Note

The following procedure assumes the installation of the iSCSI Boot image through a “lights out” management solution.

To configure the ESX 4.1 iSCSI Boot, follow this procedure:

Step 1 Make sure that you have a iSCSI Boot server profile configured.

Step 2 Make sure that the vNIC present in the iSCSI Boot server profile can reach the LUN from which the server will be iSCSI Booted.

Step 3 Insert the latest ESX 4X install CD and wait for the CD to autorun and display the (top-level) *Options* menu.

Step 4 In *Options* menu, use the up and down arrows to select the *Install ESX in graphical mode* option. This option allows you to set or modify boot-loader arguments.

Step 5 With the *Install ESX in graphical mode option* highlighted, press *F2* key to enter edit mode for the Boot Options. Figure 22 shows the *Boot Option* with the default kernel memory shown (512M).

Figure 22 ESX 4.1 Installer — Entering Boot Options

Step 6 Backspace over the `mem=` argument and overwrite the default kernel memory with the recommended minimum amount of memory. By default, 1024 MB of memory is used, but you can set the value higher if desired), then press *Enter* to resume the installer, as shown in Figure 23 on page 101.

Figure 23 ESX 4.1 Installer — Loading

When the Install progress bar completes, the Welcome screen is displayed, as shown in Figure 24.

Figure 24 ESX Installer — Welcome Screen

Step 7 Click ***Next*** and proceed through the ESX Installer. You will need to read and acknowledge the license agreement, and answer all prompts until the Custom Drivers dialog is displayed.

Step 8 On the Custom Drivers dialog, in the Load Custom Drivers section, click *Yes*. This step displays a popup that alerts you to select the drivers that you want to install., as shown in Figure 25.

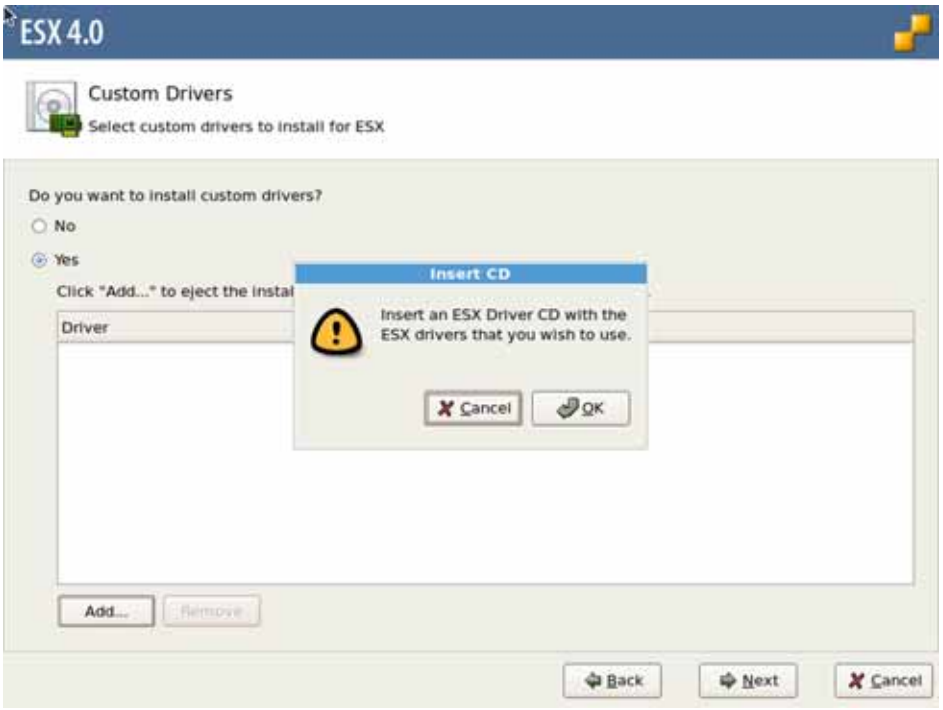


Figure 25 ESX Installer — Specify Custom Drivers

Step 9 Click ***OK*** to close the popup, then click ***Add...*** and browse to the location where the Xsigo host drivers are. You will need to install the driver ISO file to be able to connect to the Xsigo ESX host drivers ISO.

Note

The Xsigo ESX host drivers must be accessible to the ESX Installer for them to be successfully installed. For example, the Xsigo ESX host drivers can be added to the Installer directly from the `xsigo.iso` or from a network share.

Step 10 Double click each Xsigo Host Driver ISO to add them to the ESX Installer. The required modules are:

- `ib-basic`
- `vmware-esx-drivers-net-xxxx`
- `vmware-esx-drivers-ulp-xxxx`

Figure 26 on page 103 show the Xsigo host drivers added to the ESX installer.

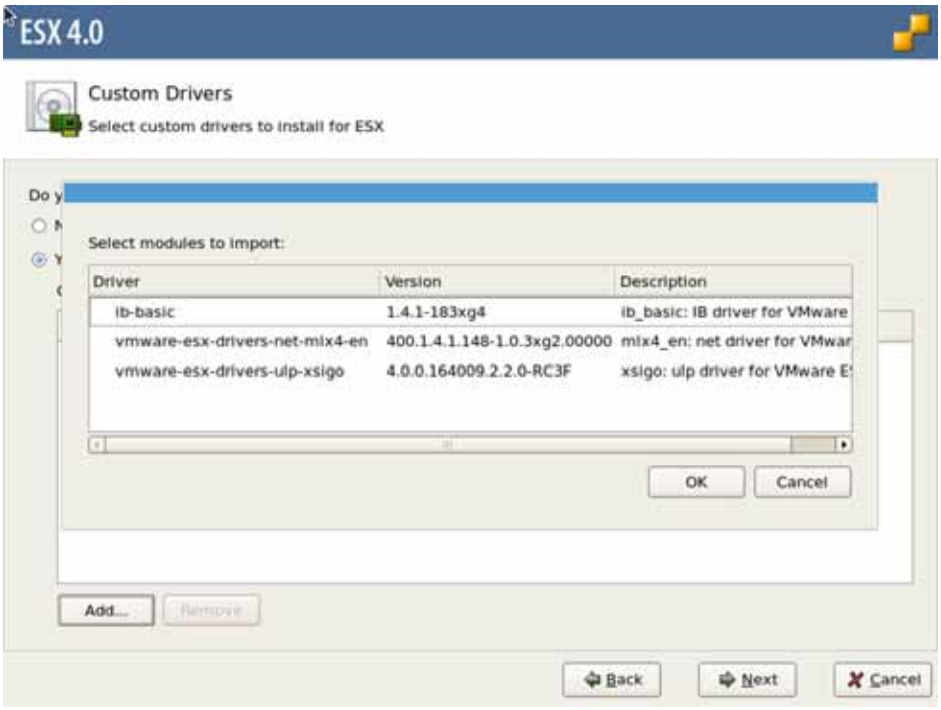


Figure 26 ESX Installer — Xsigo Drivers Specified

Step 11 Select (highlight) all 3 drivers and click ***OK*** to add them to the Drivers table in the ESX Installer.

You will see an additional popup that warns you about loading custom drivers. Click the “I Accept” check box, and click ***I Accept***. The Load Drivers popup is displayed as shown in Figure 27 on page 104.

Figure 27 ESX Installer — Load ESX and Custom Drivers

Step 12 Click ***Yes*** to continue the installation. A progress bar is displayed while the ESX drivers and Xsigo ESX host drivers progress through-installation.

Step 13 Continue the installer until you see the Network Configuration dialog. This dialog is where you will suspend the installer and enter the debug shell to run a Xsigo script.

Step 14 Press ***Alt+F2*** to enter the debug shell on the ESX Server. This step suspends the ESX Installer so that you can run the Xsigo install-load script. Figure 28 on page 105 shows the debug shell.

Figure 28 ESX Debug Shell

Step 15 In the debug shell, press ***Enter*** to activate the ESX console.

Step 16 At the prompt, run the Xsigo install-load script:

```
/tmp/drivers/usr/sbin/esxcfg-xgutil install-load
```

The script loads necessary software modules and verifies that the iSCSI Boot disk is available, as shown in Figure 29.

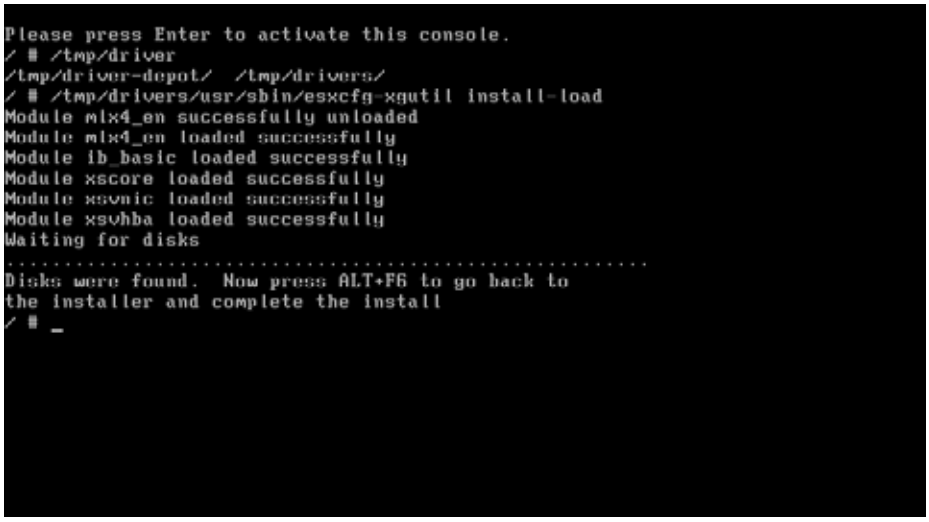


Figure 29 ESX Debug Shell — Running the Xsigo install-load Script

Step 17 When the script is completes, press ***Alt+F6*** to exit the debug shell and return to the ESX Installer.

Step 18 On the Specify ESX Datastore dialog, you will need to specify the datastore for virtual machines on the ESX Server. Make sure to select:

- Create new datastore
- Create on the same device as ESX
- Alternatively, you can preserve an existing data store by clicking *Use existing datastore* and specifying the partition. This option requires that the datastore be from the same version of ESX. For example, an ESX Classic 4.1 datastore used for iSCSI Booting an ESX 4.1 Classic server.

Figure 30 on page 106 shows a sample of the Specify a Datastore dialog.

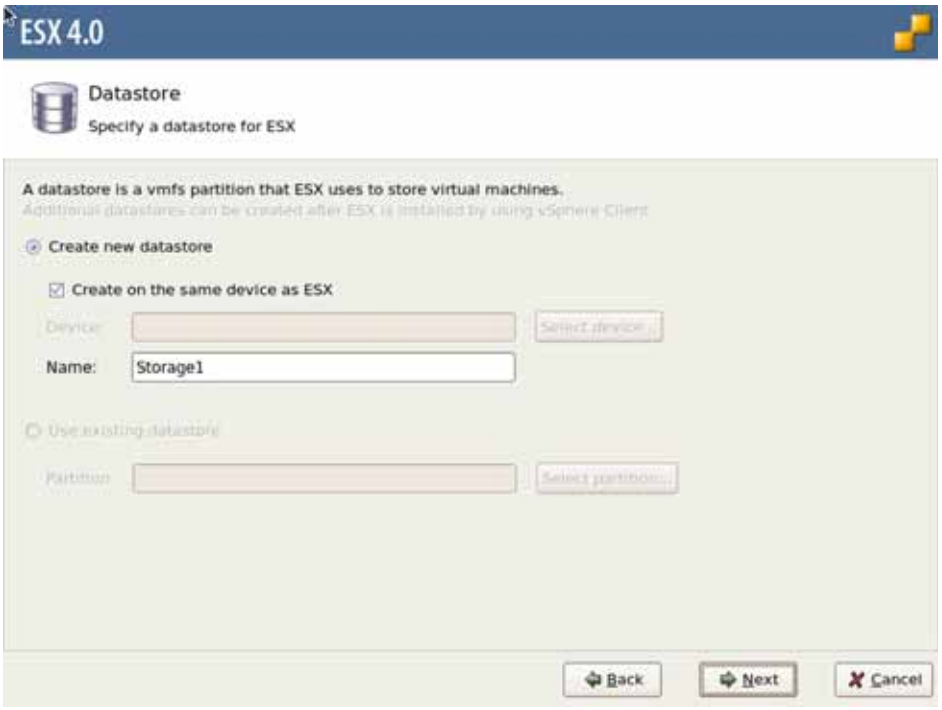


Figure 30 Specify the Datastore for ESX

Step 19 Continue the ESX Installer by configuring the standard information required by the ESX 4.1 Server, such as:

- Specifying the IP address method for the ESX Server.
- Selecting Standard Setup to boot off of a single LUN or hard drive
- Selecting the storage target that contains the LUN.

Note

If you are upgrading an existing ESX Server (or re-installing) and the target was previously associated with the ESX Server, you will see additional popups that assist you with performing a clean install.

- Selecting the time zone in which the ESX Server is being configured for iSCSI Boot.
- Specifying the NTP server (if applicable) with which the ESX Server will synchronize.
- Setting the administrator root password.

Step 20 When all ESX drivers and Xsigo host drivers, and iSCSI Boot options are specified, verify the intended configuration by reviewing the Summary of Installation dialog.Figure 31 shows an example of this dialog for illustrative purposes only. Your actual dialog will differ.

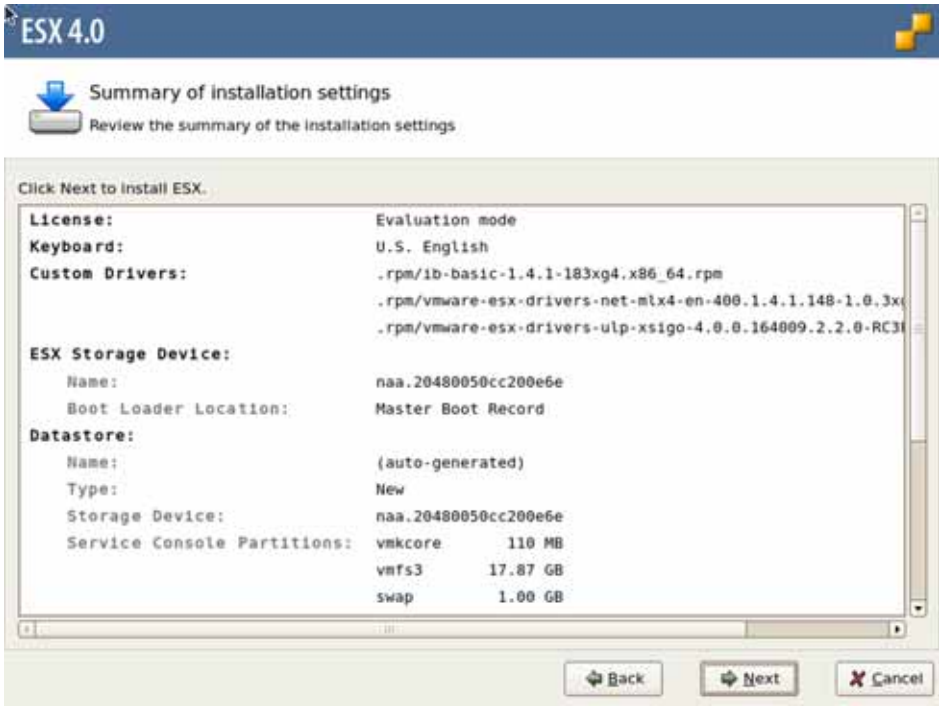


Figure 31 ESX Installer — Review iSCSI Boot and Other Settings

Step 21 Reconnect to the ESX Installer (or re-insert the physical medium if you are installing from DVD). If you see the popup shown in Figure 32 on page 108, you must reconnect to (or re-insert) the ESX installer DVD.

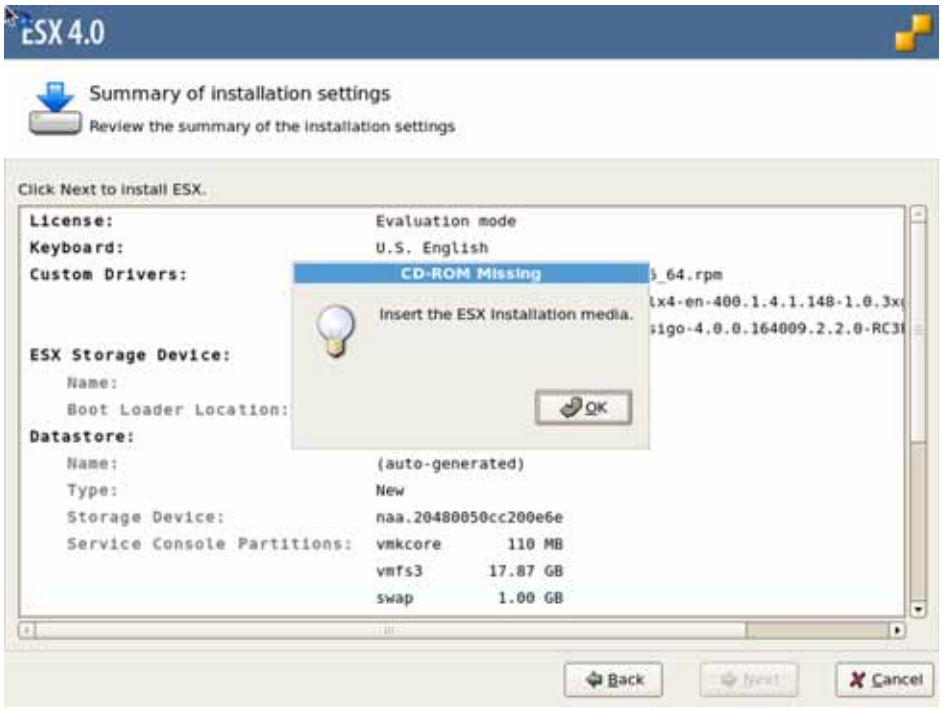


Figure 32 ESX Installer — Insert the Installation Medium

Step 22 Complete the installation by reviewing and confirming the remaining dialogs.

Step 23 Reboot the ESX Server.

SAN Boot allows you to boot a supported Windows OS from a SAN volume accessed over a vHBA. See Figure 1.

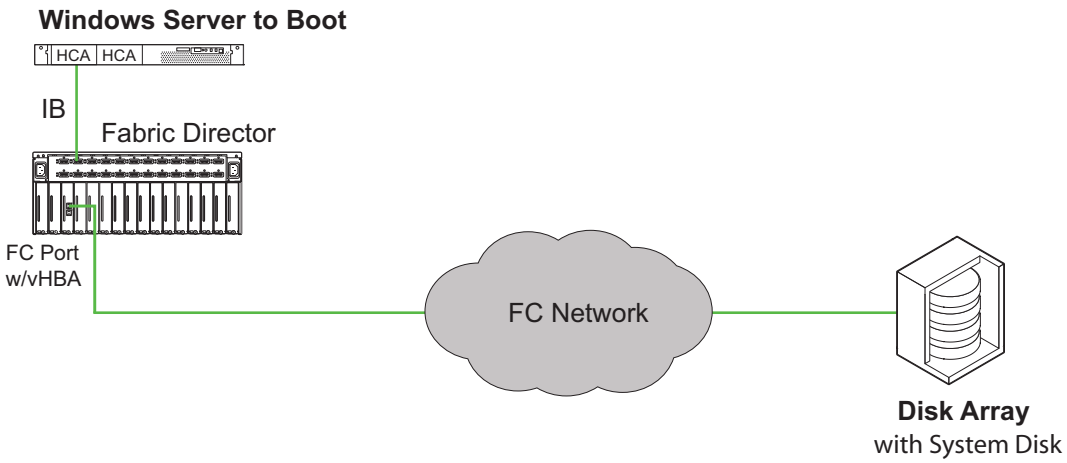


Figure 1 Windows SAN Boot Topology

This chapter explains how to set up SAN Boot for a Windows server. It contains the following sections:

- Understanding Windows SAN Boot
- Requirements for SAN Boot Installation
- Working with the Windows PE Disk
- SAN Booting Windows Server 2008 Hosts without PE
- SAN Booting Windows 2003 Hosts
- SAN Booting Windows 2003 Hosts Without PE

Note

A separate document exists for information about SAN installing the modified boot image on the SAN. If you need information about SAN Install, see *SAN Install For Windows Server 2008 Servers*.

Understanding Windows SAN Boot

When configuring a Windows server to boot over an Fabric Director vHBA, remember the following requirements:

- Each server requires read/write access to its own dedicated boot volume.

  Servers cannot share boot volumes.
- You place a fully configured operating system image, including Xsigo host drivers, on the SAN volume.

  Windows servers do not use the Xsigo initrd for SAN Booting.

Requirements for SAN Boot Installation

Before you begin either SAN-boot configuration process, you need:

- CDs or installation files for the supported Windows OS version
- CD or an unzipped archive of the Xsigo host driver software
- WinPE for a preliminary boot of the host server

  Xsigo recommends that you use WinPE 3.0 included with the Windows 7 Automated Installation Kit 3.0. For information about creating the Windows PE disk, see Working with the Windows PE Disk.

  If you create a custom WinPE disk for your installations, you must include the Xsigo host drivers so that you can see the vNICs and vHBAs. To include the host drivers, follow the instructions in Creating the Windows PE RAM CD.

Working with the Windows PE Disk

With Windows SAN Boot, you can use either Windows PE disk or Windows Deployment Service (WDS). For illustrative purposes, this procedure documents the Windows PE disk method.

Windows Preinstallation Environment (WinPE) is a scaled-down version of a Windows operating system that provides minimal functionality, but just enough to allow the Windows host server to boot. The WinPE disk allows the server to boot up so that Xsigo host drivers can be installed. For more information about WinPE and a WinPE disk, see Microsoft's website. As an alternative, Xsigo has created a sample PE disk with drivers included. You can download the sample WinPE disk by contacting Xsigo Customer Support.

When you create the Windows PE disk, you will need to load the Xsigo host drivers in PE so that you can access the virtual resources (vNICs and vHBAs).


Note

If you will need to flash the HCA firmware and Option ROM in the Windows Preboot environment (PE), be aware that you must use the 32-bit PE. The reason for this requirement is that the tools to burn the PE image are 32-bit only. If a 64-bit PE system is used you cannot burn the PE image because a 64-bit PE has no 32-bit subsystem where the tools (which are 32-bit) will run.

The overall process of creating the Windows PE disk has the following phases:

- Obtaining the WAIK 3.0 package from Microsoft. This package is freeware, so you can download it and install it.
- Obtaining the `loaddriver.bat` file.
- Creating the Windows PE RAM CD with Xsigo drivers.

For Windows SAN Boot, WinPE 3.0 is recommended.

Getting the Windows 7 Automated Installation Kit

The Windows 7 Automated Installation Kit (WAIK) enables the creation of a bootable WinPE disk. WAIK is freeware, so you can download it from Microsoft:

http://www.microsoft.com/download/en/details.aspx?id=5753

To get WAIK 3.0:

Step 1 Download WAIK from Microsoft's website.

Step 2 Install it on any supported Windows system, which will be the "Technician PC." The "Technician PC" should have a CD/DVD ROM burner installed in order to write the PE image onto the CD.

Step 3 After installing WAIK, read Microsoft's "Getting Started Guide" for WAIK, as well as any readme file(s) that accompanied WAIK for information about system requirements, known issues, and so on.

Creating the Windows PE RAM CD

This section documents how to create a bootable WinPE CD with Windows 7 Automated Installation Kit (WAIK).


Note

By default, the WinPE environment does not allow running visual basic scripts. If you foresee the need to be able to run scripts in WinPE (for example, to update the Option ROM on the HCA), you will need to add scripting capability to the WinPE CD. For information about adding scripting capabilities, see Adding Scripting Capability to the WinPE Disk.

Requirements

To perform the procedure in this section, you must have the following:

- WAIK, Windows Automated Installation Kit for Windows 7. If you do not have this package, get it now. See Getting the Windows 7 Automated Installation Kit.
- CDBurn or DVDBurn (or other compatible burning utility), which allows burning a DVD from an ISO image (Windows Server 2003 Resource Kit Tools).
- Technician PC, which is a Windows-compatible workstation or PC with a CD/DVD ROM Drive (RW capable) with Windows AIK installed on it.
- A blank CD or DVD, which will be used to create the bootable Windows PE RAM CD.
- The loaddrivers.bat file, which must be downloaded from the Xsigo FTP site. You will need a valid user name and password to access the file.

Creating the Bootable Disk

When the listed requirements have been met, you can create a bootable disk by following this procedure:

Step 1 Run the following commands from "Windows PE Tools Command Prompt":

```
copype.cmd x86 c:\winpe_x86
```

Step 2 Open with Windows AIK command prompt to perform the following steps.

Step 3 Change directory, (**cd**) to `c:\winpe_x86\iso\sources`

Step 4 Copy the provided `winpe.wim` to `boot.wim`:

```
copy C:\Program Files\Windows AIK\Tools\PETools\x86\winpe.wim
c:\winpe_x86\iso\sources\boot.wim
```

Step 5 Make the mount directory:

```
md c:\winpe_x86\iso\sources\mount
```

Step 6 Using `dism.exe`, mount and edit the `boot.wim` file:

```
dism /mount-wim /wimfile:c:\winpe_x86\iso\sources\boot.wim /index:1
/mountdir:mount
```

Step 7 Copy the utilities needed to perform the required tasks in Windows PE:

For `imagex.exe`:

```
copy C:\Program Files\Windows AIK\Tools\x86\imagex.exe
C:\winpe_x86\iso\sources\mount\windows
```

You will use `imagex.exe` as a tool when the server is booted into WinPE. Another tool, `dism.exe`, is similar to `imagex.exe`. You will use dism.exe to build the PE disk.

```
copy C:\Program Files\Windows AIK\Tools|petools\x86\bootsect.exe
C:\winpe_x86\iso\sources\mount\windows\
```

Step 8 Log in to the Xsigo FTP site, and download the `loaddrivers.bat` file to the following directory on the local server:

```
C:\winpe_x86\iso\sources\mount\
```

For the `loaddrivers.exe` file to load properly, Xsigo's device query tool `xginstdev32.exe` needs to be copied to the root of the PE image:

Step 9 Copy Xsigo device diagnostic tool to the mounted image:

```
copy c:\xsigos-3.0.0-whql\xsigo\xginstdev32.exe
C:\winpe_x86\iso\sources\mount\
```

Step 10 Create a temporary folder to contain the WinPE driver package for your environment—for example, a `PEdrivers` folder.

```
md C:\winpe_x86\iso\sources\mount\PEdrivers
```

Step 11 Copy the WinPE driver package appropriate for your environment into the root of the mounted image:

```
copy c:\xsigos-3.0.0-whql\xsigo\pedrivers\san-install\x86\*.*
C:\winpe_x86\iso\sources\mount\PEdrivers\
```

Step 12 As an option, you can add scripting capability, which allows scripts to be executed in the WinPE environment. If you will be adding scripting capability to the WinPE disk, do so before burning the WinPE image onto the physical medium (CD or DVD). Proceed to Adding Scripting Capability to the WinPE Disk.

Step 13 Unmount the mounted WIM image:

```
dism /unmount-wim /mountdir:C:\winpe_x86\iso\sources\mount /commit
```

Step 14 Create the ISO for the bootable disk:

```
oscdimg -n -bC:\winpe_x86\etfsboot.com C:\winpe_x86\iso C:\winpe_x86\winpe_x86.iso
```

Step 15 Burn the ISO file to a DVD or CD-ROM:

- To burn the ISO file to a DVD:
  ```
  dvdburn d: C:\winpe_x86\winpe_x86.iso
  ```
- To burn the ISO file to a CD-ROM:
  ```
  cdburn d: C:\winpe_x86\winpe_x86.iso
  ```

Step 16 Boot into Windows PE and execute the Xsigo-provided Loaddrivers.bat file from the root of `X:`.

Step 17 When the server is booted into the PE image, the root of `X:` should contain the following:

- Xginstdev32.exe
- loaddrivers.bat
- `PEdrivers` folder containing the Xsigo PE host drivers.

Adding Scripting Capability to the WinPE Disk

In some situations, you will want to be able to run visual basic (`.vbs`) scripts in the Windows pre-boot environment. For example, you might need to run the Xsigo Firmware Update script to update the option ROM on an HCA in your Windows server. By default, WinPE does not support `.vbs` functionality, so if you want to be able to execute a script, you will need to add scripting capability to the Windows PE disk that you are creating.

Adding scripting capability is not mandatory for a standard WinPE disk. If you do not explicitly add this functionality, you cannot run scripts from the WinPE disk. In a Windows SAN Boot environment, this can be a serious drawback due to not being able to update Option ROM firmware.


Caution

If you will be adding scripting capability to the WinPE disk, do so before burning the WinPE image onto the physical medium (CD or DVD).

Assuming you have completed the steps in the previous section up through Step 11, you can add scripting capability by following this procedure:

Step 1 Mount the `boot.wim` image if not already mounted:

```
dism /mount-wim /wimfile:c:\winpe_x86\iso\sources\boot.wim
/index:1 /mountdir:c:\winpe_x86\iso\sources\mount
```


Note

If you have already mounted `boot.wim` and run the command as shown, an error message will be displayed because you are attempting mount `boot.wim` when it is already mounted. The error can be ignored; it does not affect the procedure.

```
dism /image:c:\winpe_x86\iso\sources\mount /Add-package
/PackagePath:"C:\Program Files\Windows AIK\Tools\PETools\x86\WinPE_FPs\winpe-
scripting.cab"
dism /image:c:\winpe_x86|iso\sources\mount /Add-package
/PackagePath:"C:Program Files\WindowsAIK\Tools\PETools\x86\WinPE_FPs\winpe-wmi.cab"
```

Step 2 Return to Step 13 on page 113 to complete creation of the bootable disk.

SAN Booting Windows Server 2008 Hosts without PE

SAN Boot is supported through multiple ways on different Windows OSes. This section documents how to SAN Boot a server by using Shadow Copy. Shadow Copy is part of the Volume Shadow Copy Service (VSS) which is included in the Windows Server 2008 OS. With Shadow Copy, you can take a snapshot of the data on a specific local volume on the server. After the volume is captured, you can place it on the appropriate SAN LUN as needed.

Overview

To configure the server for SAN Booting, you will need to create some commands and scripts and enter the correct content. To create these files, you will use any standard text editor (for example, Notepad). These files will be created as part of the procedure, and the file content will be given at the appropriate point of the procedure. You will also need to call different files that are embedded in Windows.

The procedure has the following general work flow:

1. Create the correct batch files and scripts on the server. The following user-configured files will be created:
   - fixbcd.cmd
   - fixbootsector_W2k8.cmd
   - fixregistry.cmd
   - removescript.script

   To support these files, the following standard Windows files will be required. Make sure that they are available on the server:
   - Bootsect.exe
   - Imagex.exe
   - bcdboot.exe

- diskshadow.exe
- reg.exe

2. Copy the boot image onto the SAN LUN.
3. Create a SAN Bootable server profile that connects the server to the SAN LUN where the boot image exists.
4. Edit the server's BIOS to set the Xsigo vHBA as the boot device. By doing so, the server will retrieve the OS from the SAN LUN and boot to runtime.

Considerations

Following this procedure provides support for SAN Booting a Windows Server 2008 R2 host. However, after completing this procedure, be aware of the following considerations:

- the server will no longer be able to boot to either safe mode or the recovery console.
- the SAN LUN must be use either master boot record (MBR) or GUID Partition Table (GPT) partitioning. Dynamic disks are not supported with Xsigo host drivers.
- If you will be SAN Booting multiple servers, you will need to perform this procedure once for each of the servers. Also, each server must have its own SAN LUN and its own bootable vHBA connected to it. You cannot connect the server to the same SAN LUN and have all servers boot from a common boot image on a central SAN LUN.

Procedure: Configuring SAN Boot with Shadow Copy

This procedure documents a way to convert an existing server that boots off of its local drive to a Xsigo SAN-Booted server. After the server is converted, the server's local hard drive will not be used to boot the server to its OS. In fact, the drive must be completely disabled. No local disks should be available as boot devices when the conversion is complete.

To convert an existing server that locally boots from hard disk to a SAN Boot system, you will perform various parts of this procedure on the server and Oracle's Xsigo Fabric Director.


Note

This procedure assumes that one partition is used on one disk.

If multiple partitions are created on the disk, capturing the image will be more complex. You will have to modify the tools listed below to capture both the small boot partition where the BootMGR and Boot Configuration Database (BCD) exist as well as capturing the partition where the OS is installed.

For this procedure, assume that the SAN LUN is `h:`.

On the server:

Step 1 Install the OS to a local disk in the server. Do not attempt to install the OS to the SAN LUN.

Step 2 Install the Xsigo host drivers and reboot as required. If you need more information, see the "Windows Host Software" chapter of the *Fabric Director Hardware and Host Drivers Installation Guide*. If needed, update the HCA firmware and Option ROM to the correct version for your hardware type. See the *Xsigo Compatibility Matrix* for additional information. You can find the *Xsigo Compatibility Matrix* on the Xsigo corporate website if needed.

On the Fabric Director:

Step 3 Add a Server for the physical server you plan to SAN Boot. For example:

```
add server-profile webapps1
```

Step 4 Add a vHBA with one LUN to your server-profile. The LUN should be the appropriate size to contain the OS and applications you intend to install.

```
add vhba vh1.webapps1 8/1
```

Step 5 Bind the Server Profile to the server by using the **`set server-profile`** `<name>` **`connect`** `<hostname>@<director-name>:<ServerPort>` command. For example:

```
set server-profile webapps1 connect webserver1@Director1:ServerPort18
```

Note

Because the Xsigo host drivers are already installed (see Step 2), the host is identifiable by host name, as shown in the example. If you need to find the port on which the server is connected, you can issue **`show fabric-port`** command, and scroll to find the connection for the relevant host.

On the server:

Step 6 Using Disk Manager, bring the newly added disk online. Figure 2 shows using Server Manager to bring a disk online.

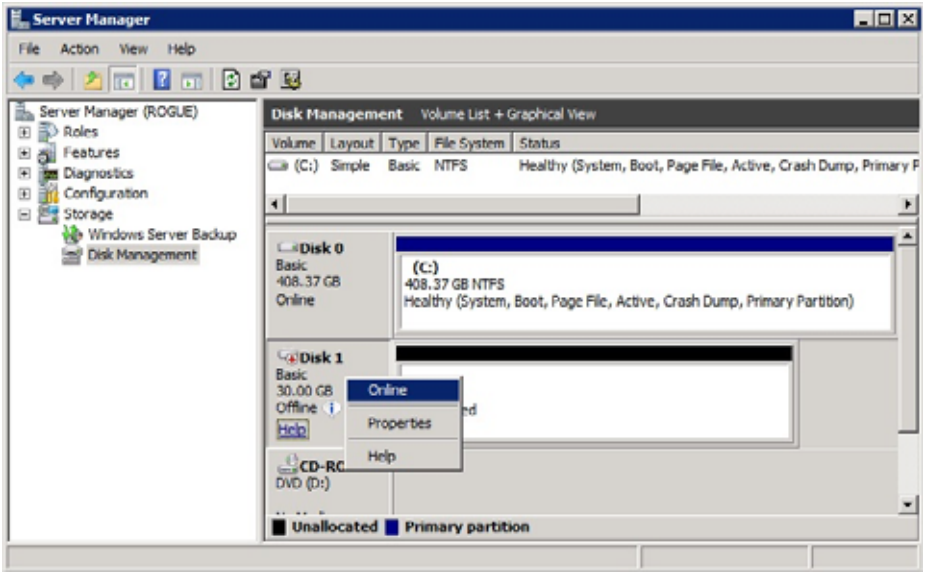


Figure 2 Server Manager — Bringing Disk Online

Step 7 Using Disk Manager, format the disk and mark it “active.” Figure 3 on page 117 shows marking the disk active through Server Manager.

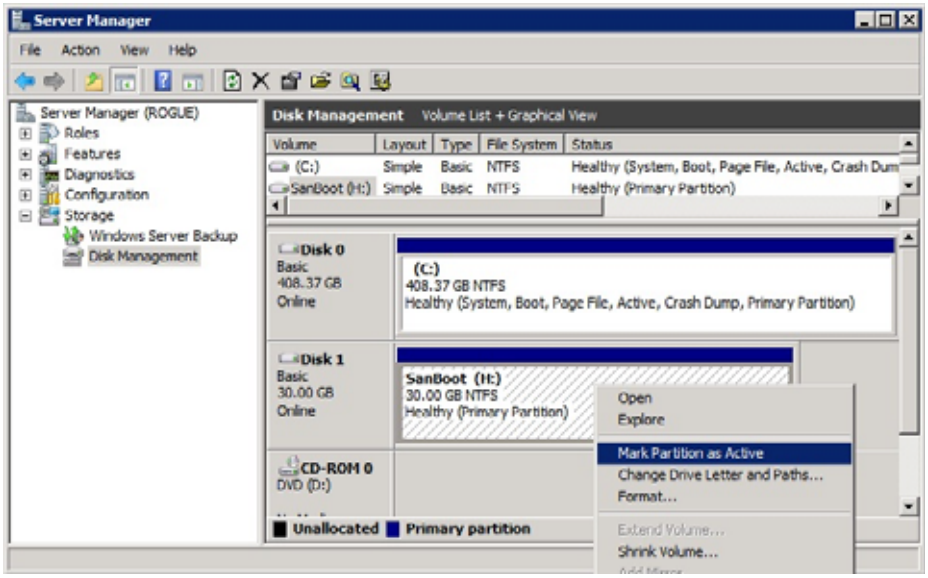


Figure 3 Server Manager — Marking Disk Active

Step 8 Using a text editor, create the `backupscript.cmd` file.

Step 9 In the `backupscript.cmd` file, enter the following content:

```
diskshadow.exe /s backupscript.script
```

When this script runs, it capture the local disk and places a copy onto the lun.

Step 10 Create the `backupscript.script` file.

Step 11 In the `backupscript.script` file, enter the following content:

```
SET CONTEXT PERSISTENT NOWRITERS
SET VERBOSE ON
BEGIN BACKUP
ADD VOLUME C: ALIAS systemVolumeShadow
CREATE
EXPOSE %systemVolumeShadow% p:
EXEC w2k8_clone_diskshadowImage.cmd
END BACKUP
```

Step 12 Create the `w2k8_clone_diskShadowImage.cmd` file.

Step 13 In the file, enter the following content:

```
imagex.exe /CAPTURE /boot P: c:\SanBoot.wim "Windows Server 2008 R2"
format.com h: /q /V:Sanboot /y
bootsect.exe /nt60 h: /force
imagex.exe /APPLY c:\SanBoot.wim 1 H: /VERIFY
diskshadow.exe /s removescript.script
call FixBCD.cmd
call FixBootSector-2k8.cmd
call fixregistry.cmd
```

Step 14 Run the following command to delete the existing BCD on the SAN LUN:

```
delete h:\Boot\BCD
```

Step 15 Run the following command to create a new BCD on the SAN LUN:

```
bcdboot.exe c:\Windows  /s h:
```

This executable creates a new BCD on the LUN and forces the new configuration information into the new BCD.

Step 16 Create the `FixBCD.cmd` file.

Step 17 In the file, enter the following syntax:

```
bcdedit.exe -store h:\boot\bcd /set {bootmgr} device boot
bcdedit.exe -store h:\boot\bcd /set {default} device boot
bcdedit.exe -store h:\boot\bcd /set {default} osdevice boot
```

When this executable is run, it updates the boot database with the new boot partition information.

Step 18 Create the `fixbootsector_W2k8.cmd` file.

Step 19 In the file enter the following syntax:

```
bootsect.exe /nt60 h: /force
```

When this executable is called, it places the boot sector on the new LUN.

Step 20 Create the `fixregistery.cmd` file.

Step 21 In the file, enter the following syntax:

```
reg.exe load hklm\sanboot h:\windows\system32\config\system
reg.exe delete hklm\sanboot\mounteddevices /va /f
reg.exe unload hklm\sanboot
```

When this executable is called, it removes any previously assigned drive letters. If the old drive letters are not removed, the system will boot to the H: drive. If the local disk controller had more than just the boot info and the OS, any additional drive letters will have to be reassigned once the server boots to SAN.

Step 22 Create the `removescript.script` file.

Step 23 In the file, enter the following syntax:

```
delete shadows exposed p:
```

On the Fabric Director:

Step 24 Make the Server Profile SAN Bootable, by setting the **`san-boot`** flag for the Server Profile. Issue the **`set server profile`** `<name>` **`san-boot`** `<vHBA> <WWPN> <LUN ID>` command. For example:

```
set server-profile webapps1 san-boot vh1.webapps1 11:22:33:44:55:66:77:88
1
```

On the server:

Step 25 Reboot the server and interrupt the POST to enter the BIOS configuration utility.

Step 26 Remove the local disk from the boot devices list by disabling the on-board storage device. However, if disabling the on-board storage is not preferable, you can remove the physical hard drive(s) from the server, but be aware that some array controllers can behave unpredictably if the drives are removed.

Step 27 Set the HCA high enough in the boot devices list order to enable the Xsigo Option ROM to complete before another device is considered for booting the server.

Step 28 Save the settings and exit the BIOS. The next time the server boots, it will use the bootable vHBA.

Step 29 If you will be converting multiple servers from local boot to SAN Boot, repeat this procedure as needed for each server.


Note

The sysprep.exe tool is not supported for this method of SAN Booting Windows Server 2008 R2 hosts.

SAN Booting Windows 2003 Hosts

SAN Booting Windows 2003 hosts supports using sysprep. However, sysprep is not required. The choice of using sysprep on your server(s) is one that you will need to make. The procedure for SAN Booting Windows 2003 Servers is similar depending on whether or not the servers will be sysprepped:

- For SAN Booting a server and using sysprep, see Procedure: SAN Booting Windows Server 2003 Hosts Plus Sysprep.
- For SAN Booting a server but not using sysprep, see Procedure: SAN Booting Windows 2003 Servers Without Sysprep.

The following considerations are applicable to SAN Booting Windows Server 2003 servers regardless of whether or not sysprep is used.

Considerations

This procedure provides support for SAN Booting a Windows Server 2003 host. Be aware of the following considerations:

- Installing directly to the SAN LUN is not supported.
- After completing this procedure, the server will no longer be able to boot to safe mode.
- The SAN LUN must be use either master boot record (MBR) or GUID Partition Table (GPT) partitioning. Dynamic disks are not supported with Xsigo host drivers.
- If you will be SAN Booting multiple servers, you will need to perform this procedure once for each of the servers. Also, each server must have its own SAN LUN and its own bootable vHBA connected to it. You cannot connect the server to the same SAN LUN and have all servers boot from a common boot image on a central SAN LUN.
- This procedure requires using WinPE. You must have a WinPE disk created for a host with Xsigo virtual I/O. If you do not have such a WinPE disk available, you must create one before attempting this procedure. For information about creating the WinPE for a host with Xsigo virtual I/O, see Working with the Windows PE Disk.

- Required files:
  - `loaddrivers.bat`. This file is available through the Xsigo FTP site. Download it before attempting this procedure.
  - `diskpart.exe`. This file is native to WinPE. You do not need to download it.
  - `Xginstdev32.exe`. This file is included in the Xsigo host driver package.

Procedure: SAN Booting Windows Server 2003 Hosts Plus Sysprep

Using sysprep allows you to revert a server back its original baseline configuration. If you want multiple servers to have the exact same configuration you should sysprep the server you used to create the SAN Boot configuration. By doing so, you ensure that all servers that will be SAN Booted have the same configuration. Consider the following example.

Assume you are building a small server farm of 50 servers that are running web-based applications. Also, assume that they will all require the same configuration. If you sysprepped the server to create a deployable Windows SAN boot configuration, you can duplicate the LUN so that additional servers can be deployed with the same configuration.

To SAN Boot a Windows 2003 Server while using sysprep, follow this procedure:

Step 1 Install a supported HCA (or two, as needed) in your server. For information about supported HCAs, see the *Xsigo Compatibility Matrix* for additional information. You can find the *Xsigo Compatibility Matrix* on the Xsigo corporate website if needed.

Step 2 Install the Windows Server 2003 OS on local disk `C:` on the server.

Step 3 Boot the server from its local disk `C:` to runtime.

Step 4 Unzip the Xsigo host drivers, and run the `setup.exe` command to install the host drivers. If you need additional information, see the “Windows Host Software” chapter in the *Fabric Director Hardware and Host Driver Install Guide*.

Step 5 If needed, update the HCA firmware and Option ROM to the correct version for your hardware type. See the *Xsigo Compatibility Matrix* for additional information. You can find the *Xsigo Compatibility Matrix* on the Xsigo corporate website if needed.

On the Fabric Director:

Step 6 Add a Server Profile for the physical server you plan to SAN Boot. For example:

```
add server-profile webapps1 <HCA port-guid>
```

You can get the GUID from the server documentation or sticker on the HCA(S) that you are installing. Alternatively, you can check the server port (**`show physical-server`**) if the Xsigo host drivers are installed on the host.

Step 7 Add a vHBA with one LUN to your server-profile. The LUN should be the appropriate size to contain the OS and applications you intend to install.

```
add vhba vh1.webapps1 8/1
```

Step 8 Reboot the server, making sure the server is booting from local disk C:.

Step 9 Verify Xsigo vNICs and vHBAs are up and running:

```
show server profile webapps1 vnics
show server profile webapps1 vhbas
```

Step 10 On the disk attached to the vHBA, create a new volume through Disk Manager by right-clicking the unallocated disk space and selecting “New Simple Volume”.

When creating the new volume, make sure of the following as shown in Figure 4.

- Use the entire disk
- Format the partition as NTFS

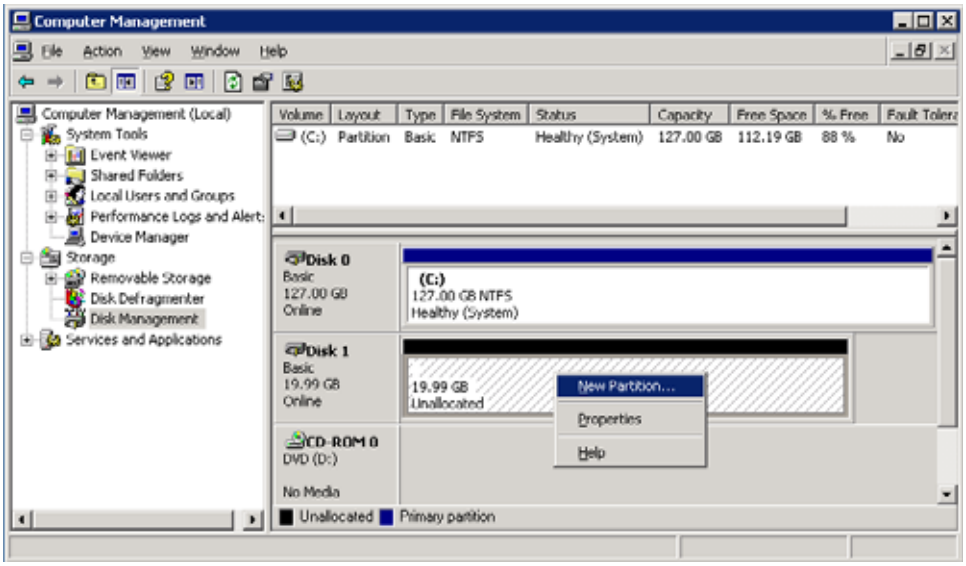


Figure 4 Server Manager — Creating New Partition

Step 11 Right-click the newly created volume and select “Mark Partition as Active” as shown in Figure 5.

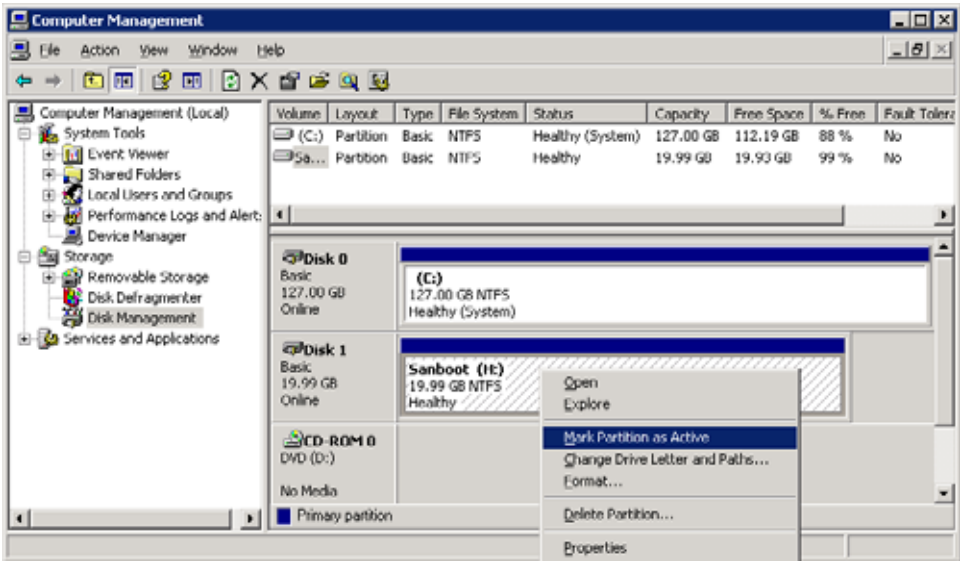


Figure 5 Server Manager — Marking Partition Active

Step 12 Locate the `deploy.cab` file in `\support\tools\deploy.cab` on the Windows Server 2003 Installation media. The `deploy.cab` file contains the following files:

- cvtarea.exe
- deploy.chm
- factory.exe
- oformat.com
- readme.txt
- ref.chm
- setupcl.exe
- setupmgr.exe
- sysprep.exe
- wfinf_guide.doc

Step 13 In the `deploy.cab` file, extract all of the individual files to a working folder named “sysprep” at `C:\sysprep` on the host.

Step 14 When the files have been extracted, refer to `ref.chm` for help about creating a `sysprep.inf` that suits your needs.

Step 15 Run `sysprep.exe` and allow it to run to completion.

Step 16 When sysprep is finished, boot the server into WinPE by using the WinPE disk that you created.

Note

If you do not have a WinPE disk for a host with Xsigo virtual I/O, abort the procedure and create one now based on the instructions in Working with the Windows PE Disk. When the WinPE disk is present, you can resume this procedure at this step.

Step 17 In WinPE, run `loaddrivers.bat` to load the Xsigo host drivers.

Step 18 In WinPE, discover the LUN from the Fabric Director. The LUN you want to discover is the LUN that is connected to the vHBA.

Step 19 In WinPE, run `diskpart` to determine what drive letter was granted to the LUN:

```
diskpart
list vol
```

Step 20 Using `imagex.exe`, capture `C:` to an image then apply that image to the LUN attached to the vHBA:

```
Imagex /capture c: d:\sanboot.wim “2003 32bit”
Imagex /apply d:\sanboot.wim 1 d:\
```

Step 21 Run the following command to apply the Windows 2003 boot sector to the drive:

```
Bootsect.exe /nt52 d: /force
```

Step 22 Exit WinPE by using any of these methods:

- Enter **`exit`**
- Enter **`wputil reboot`**
- Close the WinPE window

On the Fabric Director:

Step 23 Make the Server Profile SAN Bootable, by setting the **`san-boot`** flag for the Server Profile. Issue the **`set server profile`** `<name>` **`san-boot`** `<vHBA> <WWPN> <LUN ID>` command. For example:

```
set server-profile webapps1 san-boot vh1.webapps1 11:22:33:44:55:66:77:88
1
```

On the server:

Step 24 Reboot the server and interrupt the POST to enter the BIOS configuration utility.

Step 25 Remove the local disk from the boot devices list by disabling the on-board storage device. However, if disabling the on-board storage is not preferable, you can remove the physical hard drive(s) from the server, but be aware that some array controllers can behave unpredictably if the drives are removed.

Step 26 Set the HCA high enough in the boot devices list order to enable the Xsigo Option ROM to complete before another device is considered for booting the server.

Step 27 Save the settings and exit the BIOS. The next time the server boots, it will use the bootable vHBA.

Step 28 If you will be converting multiple servers from local boot to SAN Boot, repeat this procedure as needed for each server.

Procedure: SAN Booting Windows 2003 Servers Without Sysprep

Windows Server 2003 server can be SAN booted without sysprep. By doing so, the server is not reverted to an original configuration containing the OS. Instead, any additional configuration remains. As a result, if multiple servers require the same baseline configuration, a server that is not sysprepped will not be suitable for duplication.

To SAN Boot a Windows 2003 Server without using sysprep, follow this procedure:

Step 1 Install a supported HCA (or two, as needed) in your server. For information about supported HCAs, see the *Xsigo Compatibility Matrix* for additional information. You can find the *Xsigo Compatibility Matrix* on the Xsigo corporate website if needed.

Step 2 Install the Windows Server 2003 OS on local disk `C:` on the server.

Step 3 Boot the server from its local disk `C:` to runtime.

Step 4 Unzip the Xsigo host drivers, and run the setup.exe command into install the host drivers. If you need additional information, see the "Windows Host Software" chapter in the *Fabric Director Hardware and Host Driver Install Guide*.

Step 5 If needed, update the HCA firmware and Option ROM to the correct version for your hardware type. See the *Xsigo Compatibility Matrix* for additional information. You can find the *Xsigo Compatibility Matrix* on the Xsigo corporate website if needed.

On the Fabric Director:

Step 6 Add a Server for the physical server you plan to SAN Boot. For example:

```
add server-profile webapps1 <HCA port-guid>
```

You can get the GUID from the server documentation or sticker on the HCA(S) that you are installing. Alternatively, you can check the server port (**`show fabric-port`**) if the Xsigo host drivers are installed on the host.

Step 7 Add a vHBA with one LUN to your server-profile. The LUN should be the appropriate size to contain the OS and applications you intend to install.

```
add vhba vh1.webapps1 8/1
```

Step 8 Reboot the server, making sure the server is booting from local disk C:.

Step 9 Verify Xsigo vNICs and vHBAs are up and running:

```
show server profile webapps1 vnics
show server profile webapps1 vhbas
```

Step 10 Open the Disk Management management console for the disk attached to the vHBA.

Step 11 On the disk attached to the vHBA, create a new volume by right-clicking the unallocated disk space and selecting "New Simple Volume".

When creating the new volume, make sure of the following as shown in Figure 6.

- Use the entire disk
- Format the partition as NTFS

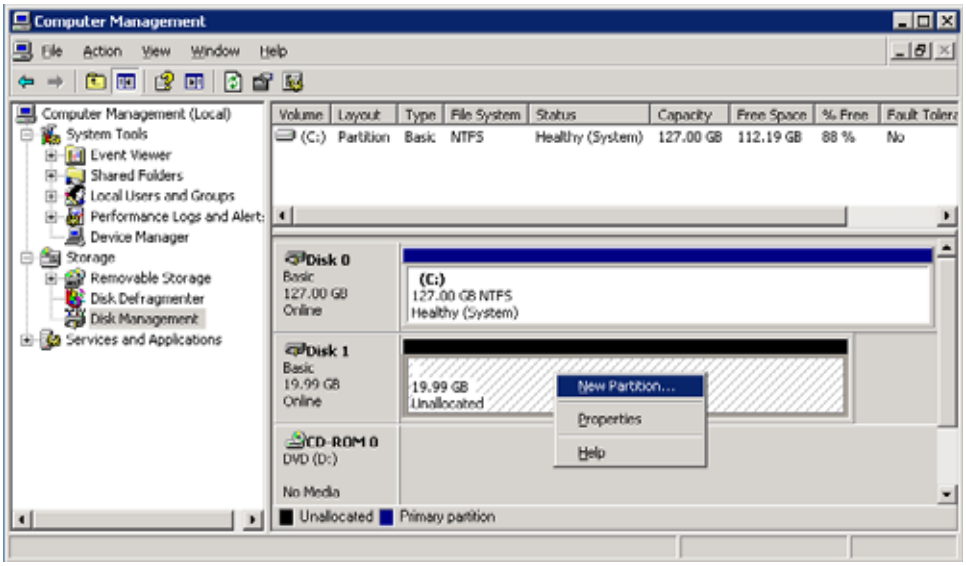


Figure 6 Server Manager — Creating New Partition

Step 12 Right-click the newly created volume and select “Mark Partition as Active” as shown in Figure 7.

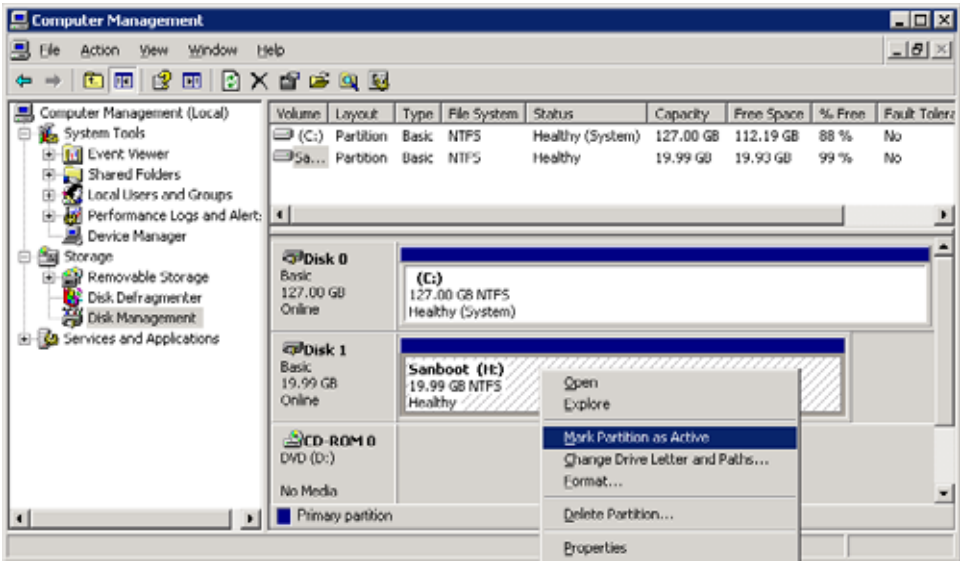


Figure 7 Server Manager — Marking Partition Active

Step 13 Boot the server into WinPE by using the WinPE disk that you created.


Note

If you do not have a WinPE disk for a host with Xsigo virtual I/O, abort the procedure create one now based on the instructions in Working with the Windows PE Disk, then resume this procedure at this step.

Step 14 In WinPE, run `loaddrivers.bat` to load the Xsigo host drivers.

Step 15 In WinPE, discover the LUN from the Fabric Director. The LUN that you want to discover is the LUN that you have connected to the vHBA.

Step 16 In WinPE, run `diskpart` to determine what drive letter was granted to the LUN:

```
diskpart
list vol
```

Step 17 Using `imagex.exe`, capture `C:` to an image then apply that image to the LUN attached to the vHBA:

```
Imagex /capture c: d:\sanboot.wim “2003 32bit”
Imagex /apply d:\sanboot.wim 1 d:\
```

Step 18 Run the following command to apply the Windows 2003 boot sector to the drive:

```
Bootsect.exe /nt52 d: /force
```

Step 19 Exit WinPE by using any of these methods:

- Enter **`exit`**
- Enter **`wputil reboot`**
- Close the WinPE window

On Oracle’s Xsigo Fabric Director:

Step 20 Make the Server Profile SAN Bootable, by setting the **`san-boot`** flag for the Server Profile. Issue the **`set server profile`** `<name>` **`san-boot`** `<vHBA> <WWPN> <LUN ID>` command. For example:

```
set server-profile webapps1 san-boot vh1.webapps1 11:22:33:44:55:66:77:88
1
```

On the server:

Step 21 Reboot the server and interrupt the POST to enter the BIOS configuration utility.

Step 22 Remove the local disk from the boot devices list by disabling the on-board storage device. However, if disabling the on-board storage is not preferable, you can remove the physical hard drive(s) from the server, but be aware that some array controllers can behave unpredictably if the drives are removed.

Step 23 Set the HCA high enough in the boot devices list order to enable the Xsigo Option ROM to complete before another device is considered for booting the server.

Step 24 Save the settings and exit the BIOS. The next time the server boots, it will use the bootable vHBA.

Step 25 If you will be converting multiple servers from local boot to SAN Boot, repeat this procedure as needed for each server.

SAN Booting Windows 2003 Hosts Without PE

SAN Boot is supported through multiple ways on different Windows OSes. This section documents how to SAN Boot a server by using Shadow Copy. Shadow Copy is part of the Volume Shadow Copy Service (VSS) which is included in the Windows Server 2003 R2 SP2 OS. With Shadow Copy, you can take a snapshot of the data on a specific local volume on the server. After the volume is captured, you can place it on the appropriate SAN LUN as needed.

Overview

To configure the server for SAN Booting, you will need to create some commands and scripts and enter the correct content. To create these files, you will use any standard text editor (for example, Notepad). These files will be created as part of the procedure, and the file content will be given at the appropriate point of the procedure. You will also need to call different files that are embedded in Windows.

The procedure has the following general work flow:

1. Update the HCA firmware (if needed).
2. Create the following batch file on the Windows Server 2003 server:
    - CloneSystemDiskUsingImagex.cmd

    To support this file, the following standard Windows files will be required. Make sure that they are available on the server:
    - Bootsect.exe
    - Imagex.exe
    - VShadow.exe


Note

VShadow is included in the Microsoft Windows Software Development Kit (SDK) for Windows Vista and later. The VSS 7.2 SDK includes a version of VShadow that runs only on Windows Server 2003.

3. Copy the boot image onto the SAN LUN.
4. Create a SAN Bootable server profile that connects the server to the SAN LUN where the boot image exists.
5. Edit the server's BIOS to set the HCA high enough in the boot devices list to enable the Xsigo Option ROM to complete before another device is considered for booting the server.

Considerations

Following this procedure provides support for SAN Booting a Windows Server 2003 R2 SP2 host. However, after completing this procedure, be aware of the following considerations:

- The server will no longer be able to boot to safe mode.
- The SAN LUN must be use either master boot record (MBR) or GUID Partition Table (GPT) partitioning. Dynamic disks are not supported with Xsigo host drivers.

- If you will be SAN Booting multiple servers, you will need to perform this procedure once for each of the servers. Also, each server must have its own SAN LUN and its own bootable vHBA connected to it. You cannot connect the server to the same SAN LUN and have all servers boot from a common boot image on a central SAN LUN.

Procedure: SAN Booting Windows 2003 Servers Without PE

Follow this procedure if you are configuring SAN Boot for Windows Server 2003 servers and are not using a WinPE disk:

Step 1 Install the OS to a local disk in the system. These instructions are not intended for installing directly to the SAN disk.

If multiple partitions are created on this disk, capturing the image will be more complex. You will have to modify the tools listed below to capture the small boot partition where the BootMGR and bcd exist in addition to the partition where the OS is installed to.

Step 2 Install the Xsigo host drivers and reboot as required.

Step 3 Using the Supported HCA Firmware list, update the HCA firmware and Option ROM to the correct version for your hardware type.

Step 4 Add a Server Profile bound to the physical hardware you plan to SAN Boot. For example:

```
add server-profile webapps1 <HCA port-guid>
```

You can get the GUID from the server documentation or sticker on the HCA(s) that you are installing. Alternatively, you can check the server port (**`show physical-server`**) if the Xsigo host drivers are installed on the host.

Step 5 Add a vHBA with a LUN to your Server Profile. The LUN should be sized appropriately to contain the OS and applications you intend to install.

Step 6 On the disk attached to the vHBA, create a new volume by right-clicking the unallocated disk space and selecting “New Partition” as shown in Figure 8.

When creating the new volume, make sure of the following:

- Use the entire disk
- Format the partition as NTFS

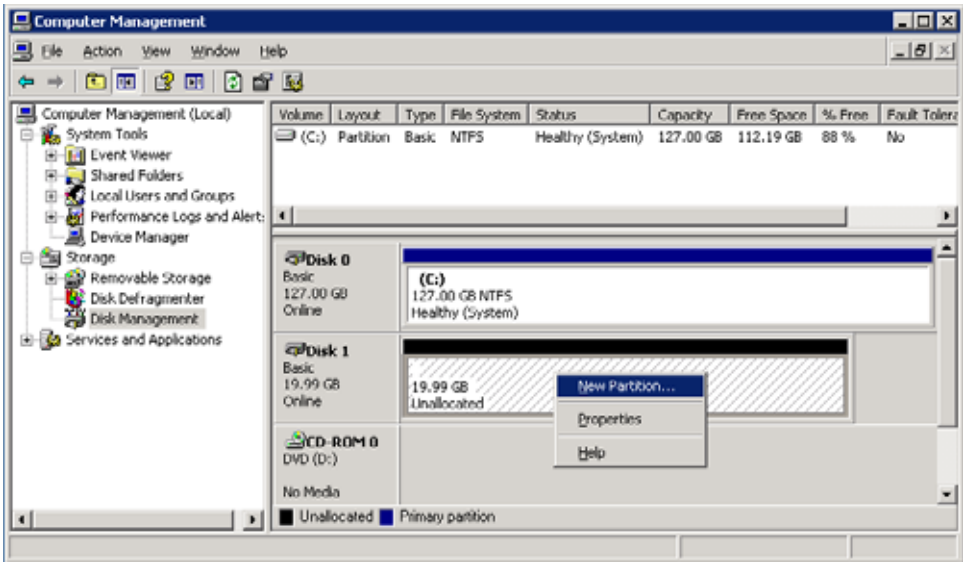


Figure 8 Server Manager — Creating New Partition

Step 7 Right-click the newly created volume and select “Mark Partition as Active” as shown in Figure 9.

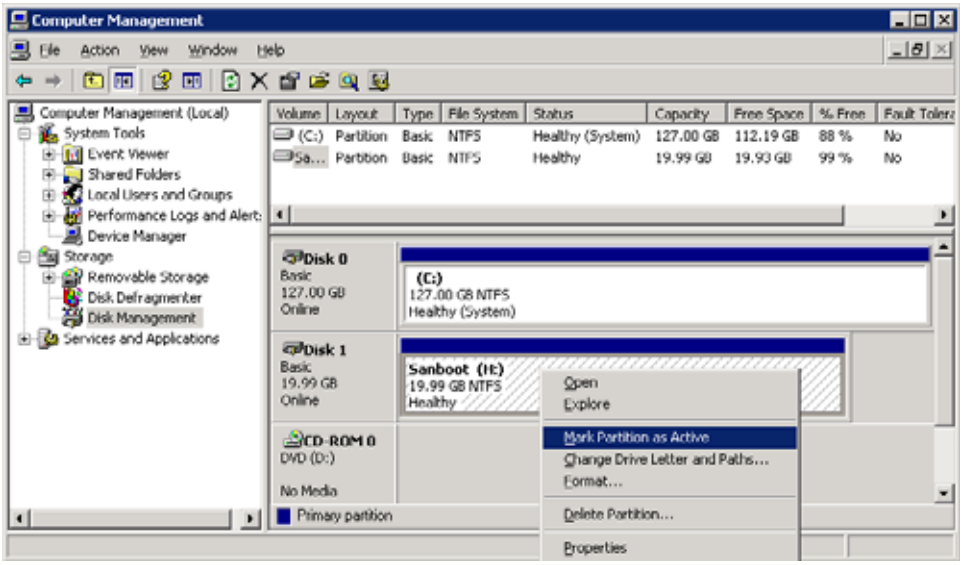


Figure 9 Server Manager — Marking Partition Active

Step 8 Create a script named `CloneSystemDiskUsingImagex.cmd` to capture the local disk and place a copy onto the LUN. The script requires the following syntax:

```
If no defined RootDriveLetter(
echo
```

```
echo
echo**********************************************************************
echo.
echo Example: CloneSystemDiskUsingImagex.cmd C: P: C:\Sanboot.wim H:
echo          Where C: is the disk to clone - OS
echo          Where P: is the shadow copy
echo          Where C:\Sanboot.wim is location of the image to be captured
echo         Where h: is location of the image to be restored - san-disk
echo.
echo
echo**********************************************************************
echo.
echo.
pause
goto :exit
)


If %PROCESSOR_ARCHITECTURE% == x86 (
set ImagexEXE=.\imagex_x32\Imagex.exe
) ELSE (
set ImagexEXE=.\imagex_x64\Imagex.exe
)

@ECHO RootDriveLetter = %RootDriveLetter%
@ECHO TempDriveLetter = %TempDriveLetter%
@ECHO DstLocationWithPath = %DstLocationWithPath%
@ECHO DstDriveLetter = %DstDriveLetter%

@ECHO Cmd to be running:
@ECHO CloneSystemDiskUsingImagex.cmd %RootDriveLetter% %TempDriveLetter%
%DstLocationWithPath% %DstDriveLetter%
@ECHO.


@ECHO vshadow -script=shadow.cmd -p %RootDriveLetter%
vshadow -script=shadow.cmd -p %RootDriveLetter%

@ECHO call shadow.cmd
call shadow.cmd

@ECHO vshadow -el=%SHADOW_ID_1%,%TempDriveLetter%
vshadow -el=%SHADOW_ID_1%,%TempDriveLetter%

%ImagexEXE% /CAPTURE /boot %TempDriveLetter% %DstLocationWithPath%
"Windows Server 2003 SP2 sanboot image" 1 /verify

@ECHO vshadow -ds=%SHADOW_ID_1%
vshadow -ds=%SHADOW_ID_1%

@ECHO imagex.exe /APPLY %DstLocationWithPath% 1 %DstDriveLetter%\
%ImagexEXE% /APPLY %DstLocationWithPath% 1 %DstDriveLetter%\ /verify

:exit
```

Step 9 Place the proper boot sector onto the LUN.

```
bootsect /nt52 h: /force
```

Step 10 Configure the Server Profile for SAN Booting by setting the **san-boot** flag. For example:

```
set server-profile webapps1 san-boot vh1.webapps1 11:22:33:44:55:66:77:88
1
```

Step 11 Reboot the server and interrupt the POST to enter the BIOS configuration utility.

Step 12 Remove the local disk by either of the following methods:

- Disabling the on-board storage device. This option is preferred.
- Removing the physical hard drive(s). This option is not preferred because some disk array controller do not tolerate having individual drives removed from them and then replaced later. However, this option can be done, but be aware that this option can cause instability in the array.

Step 13 Set the HCA high enough in the boot order to allow execution of the Xsigo Option ROM before booting to another device.

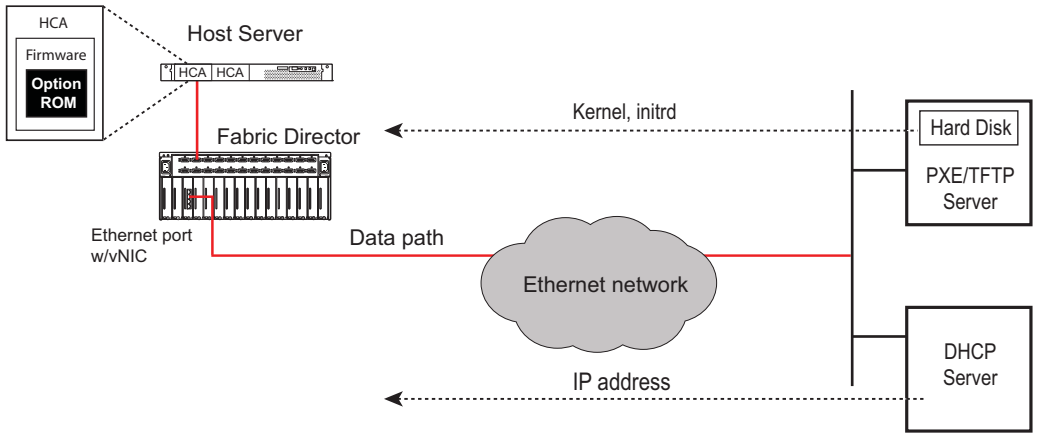
Step 14 Save the settings and exit the BIOS.

Preboot Execution Environment (PXE boot) enables a host server to boot up over the network. Using the PXE boot protocol, the host server's HCA option ROM obtains the kernel boot image and initial RAM disk (initrd) from the network (not a local disk).

This chapter explains how to configure PXE boot for a vNIC. It contains the following sections:

PXE Boot Configuration Process

Figure 1 illustrates a boot-capable server and its network connections.




```
add vnic <mypxeboot>.<myserver> <slot/port>
set vnic <mypxeboot.myserver> -addr-type=dhcp -boot-capable=yes -ip-addr=<myipaddress>
```

Figure 1 Network Boot-Capable Host Server

During the BIOS boot sequence, the host server's PXE agent (option ROM) scans the network for a PXE image and reads its boot up instructions from a boot file stored on a boot server.

The following PXE boot services are supported:

Booting to disk—The kernel and initrd come from the network, then the host server mounts modules that enable its local disk to be visible on the network.

A network install—The host server runs an OS installer program over the network and chooses any of the available options.

Configure a vNIC on the Fabric Director to support IP connectivity to the boot server. One or more servers must be reachable for the DHCP and PXE/TFTP services. These services can be hosted and running on either the same, or different physical servers or virtual machines.

DHCP and TFTP servers can be any freeware or commercially available product. Most Linux server distributions include a freeware DHCP server. For example, a distribution can include Internet Systems Consortium DHCP Server which is freeware available through the ISC website at https://www.isc.org/software/dhcp.

For configuring PXE boot, you will need to configure an instance of a DHCP server and an instance of a TFTP server. In the TFTP server, you will edit the default file (pxelinux.cfg/default) to point to the location where the Linux distribution and a kickstart script. The kickstart script is used for configuring installing options. When the kickstart script runs, the boot process brings up the servers that are connected to the Fabric Director through one or more vNICs.

The following section documents:

- getting MAC address information from the Fabric Director
- configuring a DHCP server and starting DHCP services through the `/etc/dhcpd.conf` file

- configuring the TFTP server
- configuring a TFTP boot file
- configuring a Fabric Director vNIC to support PXE booting

MAC Addresses for DHCP Requests

This section documents how to retrieve MAC address information from the Fabric Director. You will use this information when you configure the DHCP server.

Determine a range of MAC addresses that will be allowed to receive DHCP packets for PXE boot by issuing the **show hardware** command. For example:

```
admin@iowa[xsigo] show hardware
# Xsigo System Hardware Status
# Model: VP780-CH-SDR
# Serial: 050610240
# Base MAC: 00:13:97:01:80:00
# Base WWN: 50:01:39:70:00:00:80:00
# Locator LED: on
#
# Date: Sun May 11 01:32:18 GMT 2008
# User: admin
...
```

In this example, the starting MAC address in the pool of MAC addresses is allowed to receive DHCP requests.


Note

The Fabric Director's MAC address consists of hard-coded bits and variable bits. The MAC Address Mask determines the variable bits of the address. For example, the MAC Address and MAC Address Mask fields in the displayed dialog are 00:13:97:01:80:00 and 12, respectively. The MAC address mask of 12 indicates that the last 12 bits of the address are variable. As a result, the address can be interpreted as 00:13:97:01:8x:xx where x is a configurable part of the MAC address.

DHCP Server Configuration

To configure PXE boot through DHCP, first configure the DHCP server. Configuring the DHCP server is accomplished by configuring DHCP lease variables and network and subnet addresses in the /etc/dhcpd.conf file. The minimum parameters to provide DHCP services are:

- default DHCP lease time
- maximum DHCP lease time
- DDNS update style
- subnet address, netmask, and address and netmask ranges that can be used by booting servers
- PXE boot Linux kernel
- IP interface address of the vNIC (or physical Ethernet interface) where the TFTP boot server will be
- path to the Linux distributions

Configure and start the DHCP server:

Step 1 Edit `/etc/dhcpd.conf` and add the following lines:

```
default-lease-time 600;
max-lease-time 7200;
```

The lines above set the DHCP lease and renew parameters.

```
subnet 1.22.0.0 netmask 255.255.0.0{
range 1.22.0.1 1.255.255.254
```

The lines above set the addresses that can be assigned to the booting servers.

```
filename ‘pxelinux.0’;
```

The line above identifies the boot loader used for PXE booting. The boot loader is part of the `syslinux` RPM.

```
next-server 1.255.255.249;
```

The line above should match the vNIC or physical NIC interface address that will host the TFTP boot server. This address is usually the same as the DHCP server host.

```
option root-path “/tftpboot/”;
}
```

The line above identifies the locations of the Linux ramdisk.

Step 2 If the DHCP service is not active, start it by issuing the following command:

```
service dhcpd start
```

This step activates the DHCP daemon.

Tip DHCP servers post all error and warning messages to a log file. When you enable DHCP, if errors occur, you can check the log file at /var/log/messages

Step 3 If the DHCP services are already on, you can keep them persistent through boot cycles by issuing the following command:

```
chkconfig dhcpd on
```

TFTP Server Configuration

Configuring a TFTP server consists of creating a directory for the Linux distributions, editing the `tftp` file and reloading the `xinetd` file.

To configure a TFTP server, a TFTP RPM must be installed on the PXE server. You can verify that a TFTP RPM is installed by issuing the following command:

```
rpm -qa | grep tftp
```

If a TFTP server is not installed on the PXE server, you can download and install it by issuing the following command assuming the RPM is in currently in the your working directory:

```
rpm ivh tftp-server-*.rpm
```

Note

The following installation and configuration files are required for PXE boot up in a Red Hat Linux environment. Unless otherwise indicated, these files are supplied by you, the customer:
- pxelinux.0 which is the network bootable file.
- vmlinuz-2.6.9-42.EL which is the bootable compressed Linux kernel.
- initrd-rehl4u4_xsigo-i386.img, which is the Xsigo ramdisk image file. This file is provided by Xsigo.
- pxelinux.cfg, which is the directory that contains a configuration file named “default”.

When a TFTP server is installed, proceed with configuring the TFTP server for PXE booting.

Step 1 Create a directory for the Linux distributions:

```
mkdir /tftpboot
```

Step 2 Set ownership of the directory to “nobody”:

```
chown nobody:nobody /tftpboot
```

Also, make sure that /tftpboot has read-write permissions set for any accounts that are using this directory.

Step 3 Edit the /etc/xinetd.d/tftp file to change the following line to “no”

```
disable = no
```

This step is mandatory.

Step 4 As an option, you can add the following new lines at the prompt of the `/etc/xinetd.d/tftp` file:

```
log_type       =FILE /var/log/ftplog
log-on-success +=HOST EXIT DURATION
log-on-failure +=HOST USERID ATTEMPT
```

Step 5 Load the `xinetd` file onto the server:

```
service xinetd reload
```

PXE Installation

Xsigo’s PXE solution does not modify the Linux installer (Red Hat’s Anaconda installer). Instead, Xsigo uses the versions supplied directly by the OS provider and adds in Xsigo’s driver disks. A driver disk is an image that provides the installer with the required drivers. In Xsigo’s case, the vNIC and vHBA drivers are needed to use them for installation.

To use a driver disk, you have to make it available to the installer during runtime. The typical way to do this is using a CD or floppy. To do this, an image called **`xsigo-rhdd-`**`<kversion>-<version>-<arch>`**`.img`** is provided. This file is an Anaconda driver disk.

To use it, add the kernel argument **`dd`** to the kernel on boot up. Or at the boot: prompt, type **`linux dd`**. Anaconda will prompt you for a driver disk when it runs. The easiest way is to burn the ISO image onto a CD and put it in the CD drive. Anaconda will read the CD and install the Xsigo drivers.

When the drivers are loaded (which may take up to 30 seconds), Anaconda will then detect new vNICs along with any physical Ethernet interfaces present on the server. The vNICs appear with the same name as was given on the Xsigo chassis (vnic0, and so on).

Note

Due to a limitation in the Anaconda installer, the vNICs may appear with the description “Unknown device: 199d:8202” instead of a more legible string.

Xsigo recommends that you modify the stock Anaconda initrd by inserting our driver disk into it. This can be done using the script **`xg-insert-dd`**. If you do this, the “Unknown device” error will not be present and the vNICs will appear as Xsigo vNIC

For information about installing the SAN volume, see Chapter 3, “Installing the SAN Volume (Linux Servers).”

Inserting the Driver Disk into the Anaconda initrd

If you want to avoid using a CD to load the driver disk, you can insert the disk into the Anaconda initrd using the script **`xg-insert-dd`**. To do this, you must have a built driver disk and an Anaconda initrd (obtained from your Red Hat installation disks). Running the **`xg-insert-dd`** script creates a new initrd prefixed with the string “xsigo-”. If you use this in place of the default Anaconda initrd, the Xsigo drivers will be loaded at boot time without the need to specify “dd” on the kernel command line or having to insert a CD.

Special Consideration for Red Hat 4

Red Hat 4 is a mature operating system now. It uses an older kernel (2.6.9) and an older Anaconda installer (version 10). This causes us some problems resulting in different behavior when using Anaconda. Here are some special rules that need to be followed in order to use RHEL4:

1. The script **`xg-insert-dd-ext2`** must be used to insert the driver disk into the Anaconda initrd.
2. You need to add the kernel argument: **`dd=path:/xsigo-dd.img`**.
3. You will see 30 additional Ethernet interfaces numbered sequentially from the last physical port (eth2, eth3...eth31).
4. You will have to call your bootable vNIC “eth<x>” where <x> is a number from 2 to 31.

The reasons for these are that the kernel is too old to support proper addition of PCI devices with given names. It automatically creates a full PCI bus and you cannot change the name given to the device.

The Anaconda supplied with rhel4 has a bug that causes it to crash if you supply a 'dd.img' in the initrd (this should cause the driver disk to be automatically loaded). That is why you have to supply the **`dd=path:/xsigo-dd.img`** as a kernel argument.

Xsigo HCA Firmware Configuration

To support PXE boot, you must flash the Xsigo HCA firmware with the new PXE boot programming options. Check the *XgOS Software Upgrade Guide* or *Release Notes* for the Oracle XgOS release on your Fabric Director to determine the compatible firmware levels. To upgrade firmware and enable the option ROM, refer to “Linux Firmware and Option ROM” on page 154

PXE Boot Virtual NIC Configuration

The next steps in configuring a vNIC to support PXE boot, is to create a server profile and populate it with a vNIC.

Step 1 Locate a physical server on which you can configure a server profile:

```
show fabric-port
-------------------------------------------------------------------------------
name         alexander
type         hcaPort
descr
port         iowa:ServerPort23
id           2c90200204cbe
state        NA/up
m-key        0
lid          14
sm-lid       1
link-width   4x
link-speed   2_5_Gbps
-------------------------------------------------------------------------------
```

In this example, the red text highlights the host name of the physical server. Make a note of this string. You will use it in the next step. If host names are not assigned to the servers, you can use the server's GUID which is indicated in the `id` field in the output of the **`show fabric-ports`** command.

Step 2 Add a server profile and specify the host name or GUID of the physical connection. For example, to add a server profile called "pxeboot" to the physical server "alexander", you would enter the following command:

```
add server-profile pxeboot alexander@ServerPort23
```

Step 3 Add a vNIC, specify its port and interface number. For example, to add a vNIC called vn1.pxeboot to slot 6, port 2:

```
add vnic vn1.pxeboot 6/2
```

Step 4 On the vNIC, set the following parameters:

- set the address type to **`dhcp`**
- set **`-boot-capable=true`**
- configure an IP address and netmask (only if addresses are not assigned buy DHCP)

For example, to configure vn1.pubstest2 for PXE boot through DHCP, and configure IP address, enter:

```
set vnic vn1.pubstest2 -addr-type=dhcp -boot-capable=true
```

Step 5 Reboot the host, and while the host is booting, enter the BIOS setup.

Step 6 In the server BIOS, select the HCA as a boot device. You will want to set this HCA either as the first boot device, or to a relatively high position in the server's boot devices list, to speed up the boot process. If other devices are higher in the boot order, they will be attempted, but will eventually fail and pass to the HCA.

Step 7 When the host boots from the *pxelinux.cfg* directory, various boot options are offered. Two common boot options are *linux* and *local*, but others might be present:

- Select *linux* for performing a PXE boot again.
- Select *local* for booting from hard drive.

PXE Boot Configuration for an ESX 4.1 Classic Server

PXE Boot is supported for the ESX 4.1 (ESX 4.1 Classic) servers through a procedure that has the following stages:

- Extracting the “make PXE” Tool from the Xsigo host driver bundle. This stage typically occurs on the PXE Server.
- Creating the Modified initrd for booting. This stage typically occurs on the PXE Server.
- Creating a PXE Boot Server Profile with a “bootable” vNIC that can reach the device where the modified initrd is installed. This stage occurs on the Oracle Fabric Director.
- Editing the PXE Config file. This stage occurs on the PXE Server.
- Making the New PXE Config Available to Booting Servers. This stage occurs on the PXE Server.

To perform this procedure, you will need:41i

- Root access on the ESX 4.1 server
- The ESX 4.1 Server’s OS CD
- The Xsigo ESX 4.1 Classic host driver software


Note

Configuring PXE boot for an ESXi 4.1 Server is a different process. For information about configuring PXE boot for an ESXi 4.1 installation, see PXE Boot Configuration for an ESXi 4.1 Server.

Extracting the “make PXE” Tool

To include the Xsigo host drivers in the PXE boot disk, you will run a shell script called `xsigo-mkpxe-initrd.sh` (also called “make PXE”) to inject the Xsigo host drivers into the ESX 4.1 initrd. The “make PXE” tool is included as part of the standard Xsigo host drivers bundle for ESX 4.1 Classic.

To extract the “make PXE” tool follow this procedure:

Step 1 Login as root user on the ESX 4.1 server.

Step 2 Copy the ESX initrd off of the ESX 4.1 OS CD. You can put the ESX initrd in any directory. For this procedure, assume we will use `opt/`

Step 3 Put the Xsigo host drivers on the ESX 4.1 server.

Step 4 Locate the `xsigo-4.1.0.164009.2.2.0.esx41i.tgz` file and decompress it to a directory. For illustrative purposes, the `opt/` directory is shown, but you can use whatever directory you want as long as the ESX initrd and the `xsigo-4.1.0.164009.2.2.0.esx41i.tgz` file reside in the same directory:

```
tar -zxvf xsigo-4.1.0.164009.2.3.0.esx4.tgz opt/
```

Step 5 When the bundle is extracted, locate the `xsigo-mkpxe-initrd.sh` file. This is the tool you will use to inject the Xsigo host drivers into the ESX initrd.

Proceed to the Creating the Modified initrd.

Creating the Modified initrd

To create the modified initrd, you will need to modify the initrd on a Linux server. You will need the following:

- to be logged in to the Linux server with root privileges
- execute privileges on whatever directory you will use for injecting the Xsigo host driver

To create the modified initrd, you will run the "make PXE" tool (`xsigo-mkpxe-initrd.sh`) to inject the Xsigo host drivers into the ESX initrd.

The "make PXE" tool has the following usage help, which can be invoked by running the tool without any of the mandatory qualifiers.

```
xsigo-mkpxe-initrd.sh --initrd <VMware-initrd> --xgfile <Xsigo driver tgz file>
[--output <filename>] [-d]

--initrd <VMware initrd>   required VMware initrd from VMware ESX Classic CD

--xgfile <XG File>         required file containing compatible Xsigo drivers

--output                   optional filename for output. Default is XG-<INITRD FILENAME>

-d                         debug. Produces a logfile of xsigo-mkpxe-initrd.log
```

By default, when the Xsigo host drivers are successfully injected into the ESX initrd, a new initrd is created called `XG-initrd.img`.

For this procedure, you need to use only the **`--initrd`** and **`--xgfile`** qualifiers.

- You can use **`--output`** to rename the modified initrd if needed. This option supports changing the default name of the modified initrd from `XG-initrd.img` to something else.
- You can use the **`-d`** option if you encounter errors while attempting to inject the Xsigo host drivers into the ESX initrd, or if you are explicitly requested to do so by Xsigo Customer Support.

Follow this procedure:

Step 1 Move the `xsigo-mkpxe-initrd.sh` into the same directory as the Xsigo host drivers.

Step 2 Inject the Xsigo host driver into the ESX initrd by running the "make PXE" tool:

```
sh xsigo-mkpxe-initrd.sh --initrd initrd.img --xgfile xsigo-
4.1.0.164009.2.3.0-7.esx4.tgz

Outputting XG-initrd.img
```

Step 3 When the modified initrd is created, list the contents of the directory (**`ls`**). The modified initrd (`XG-initrd.img`) should be present.

```
ls
initrd.img xsigo-mkpxe-initrd.sh XG-initrd.img xsigo-4.1.0.164009.2.3.0-
7.esx4.tgz
```

If the `XG-initrd.img` is not listed, you can run the "make PXE" tool in debug mode (with the **`-d`** qualifier) and check the log file (`xsigo-mkpxe-initrd.log`) for pertinent error messages. If the log file does not provide useful information, you can contact Xsigo Customer Support.

Proceed to Creating a PXE Boot Server Profile.

Creating a PXE Boot Server Profile

If you have not already created a PXE Boot server profile, you must do so. The PXE Boot Server profile must have only one bootable vNIC that is connected to the PXE server where the ESX4 boot image will reside.

Step 1 Create the server profile for the PXE Server. For example, to create the server profile “esx4” for the PXE boot server “gorgon” which is connected through IB port 23 on the Fabric Director “tuffy:

```
add server profile esx4 gorgon@tuffy:ServerPort23
```

Step 2 Create the vNIC for the PXE Boot Server Profile. For example, to create a vNIC named “vnic1” in server profile “esx4” and have the vNIC terminated on port 1 in slot 8:

```
add vnic vnic1.esx4 8/1
```

Step 3 Add Set the vNIC for PXE Booting:

```
set vnic vnic1.esx4 -boot-capable=true
```

Step 4 If the server has an OS loaded, verify that the server profile is configured correctly and in the `up/up` state.

```
show server-profile esx4
name     state   descr   connection                    def-gw   vnics   vhbas
-------------------------------------------------------------------------------
esx4     up/up           gorgon@tuffy:ServerPort23              1
1 record displayed
```

If the server does not have an OS loaded, the Server Profile will not be `up/up`. Instead, it will be `up/unassigned`.

When the modified initrd is ready, and the PXE Boot server profile is created, you will need to edit the PXE config file on the PXE boot server. Proceed to Editing the PXE Config.

Editing the PXE Config

On your network’s PXE server, you will need to move the modified initrd onto the PXE server, and edit the PXE server’s config file to add the ESX 4.1 Classic bulletins provided with the Xsigo host drivers.

Step 1 Move the modified initrd image to the PXE server. You can use SCP or any other common file-transfer protocol to get the modified initrd onto the PXE server.

Step 2 Using **vi** (or some other common Linux editor), add the following lines to the PXE server’s PXE config file.

The following example shows using a kickstart named “vmware/4.1.0-164009” in the `method` string. If you are using a kickstart script, it will most likely have a different name.

```
label esx41-pxe3 #3.5.0 drivers
        kernel vmlinuz-esx-260247
        append initrd=XG-3.5.0-initrd-260247.img mem=768M askmedia
```

Step 3 Save the changes and quit the editor.

Proceed to Making the New PXE Config Available to Booting Servers.

Using kickstart For an Unattended Installation

When using kickstart to set up your SAN Boot environment, refer to the Red Hat Linux *Installation Guide* at https://www.redhat.com/docs/manuals/enterprise/ and follow the instructions for preparing a kickstart installation. The information that follows provides the specifics of using kickstart to deploy the xsigo-initrd.

Before setting up the kickstart installation, prepare the xsigo-initrd and Xsigo host drivers for a standard manual install. If you are using multipathing software, customize your xsigo-initrd as described in Modifying the initrd for Multipathing with RHEL 5.x Hosts on page 19.

When your xsigo-initrd is ready, write a script to copy the initrd to the SAN volume. For example, if your xsigo-initrd is available from an internal web server, your script might include the following:

```
cd /boot
wget http://sample_server.domain.com/linux/xsigo/xsigo-initrd-<kernel>-
<version>.img
cd /tmp
sed -e 's/initrd-<kernel>/xsigo-initrd-<kernel>-<version>/' /boot/grub/
grub.conf > /tmp/grub.conf
mv -vf /tmp/grub.conf /boot/grub/grub.conf
```

In this example, the script gets the Xsigo image from the web server and modifies the GRUB configuration file. When your script is complete, invoke it from the %post section of your kickstart script.

Making the New PXE Config Available to Booting Servers

After all modifications are made and the new initrd is on the PXE Server, you will need to flag the newly created initrd so that it can be used to boot any server that has received the PXE boot image. For example, the steps in this section set the options in the kickstart script so that a booting server can receive required install-time settings.

Step 1 Edit the kickstart script to include labels that append the initrd.

The following example shows using a kickstart named “vmware/4.1.0-164009” in the `method` string. If you are using a kickstart script, it will most likely have a different name.

```
label esx4-pxe

kernel vmlinuz-esx-164009

append initrd=XG-initrd.img method=http://<PXE-server>.<domain>.com/
vmware/4.1.0-164009 mem=1024M
```

You must set the memory allocation to 1024 MB or higher. Do not use a lower memory value, and do not forget to include this parameter.

Step 2 Close the kickstart script, making sure to preserve the changes.

PXE Boot Configuration for an ESXi 4.1 Server

PXE Boot is supported for the ESXi 4.1 servers through a procedure that has the following steps:

- Creating a PXE Boot Server Profile with a “bootable” vNIC that can reach the device where the ISO image is installed
- Editing the PXE Config File to append the relevant files for PXE booting
- Loading the Image into the ESXi 4.1 Server


Note

Configuring PXE boot for an ESX 4.1 Server is a different process. For information about configuring PXE boot for an ESX 4.1 installation, see PXE Boot Configuration for an ESX 4.1 Classic Server.

Creating a PXE Boot Server Profile

If you have not already created a PXE Boot server profile, you must do so. The PXE Boot Server profile must have only one vNIC that is connected to the PXE server where the ESXi 4.1 boot image will reside.

Step 1 Create the Server profile. For example, to create the server profile “esx41i” for the PXE boot server “gorgon” which is connected through IB port 23 on the Oracle Fabric Director “tuffy:

```
add server profile esx41i gorgon@tuffy:ServerPort23
```

Step 2 Create the vNIC for the PXE Boot Server Profile. For example, to create a vNIC named “vnic1” in server profile “esx41i” and have the vNIC terminated on Gigabit Ethernet port 1 in slot 2:

```
add vnic vnic1.esx41i 2/1
```

Step 3 Create a vHBA for the PXE Boot Server Profile. For example, to create a vHBA named “vhba1” in server profile “esx41i” and have the vHBA terminated on Fibre Channel port 1 in slot 8:

```
add vhba vhba1.esx41i 8/1
```

Step 4 Set the vNIC for PXE Booting:

```
set vhba vhba.esx41i -boot-capable=true
```

Step 5 If the server has an OS, verify that the server profile is configured correctly and in the `up/up` state.

```
show server-profile esx41i
name     state   descr   connection                    def-gw   vnics   vhbas
-------------------------------------------------------------------------------
esx41i    up/up           gorgon@tuffy:ServerPort23              1       1
1 record displayed
```

If the server has no OS installed, the Server Profile state will not be up/up. Instead, it will be `up/unassigned`.

With the modified ISO ready and the PXE Boot server profile created, you will need to edit the PXE config file on the PXE boot server.

Editing the PXE Config File

On your network's PXE server, you will need to move the modified ISO onto the PXE server, and edit the PXE server's config file to add the ESXi 4.1 bulletins provided with the Xsigo host drivers.

Step 1 Move the stock ISO image to the PXE server. For example:

```
cp /tmp/VMware-VMvisor-Installer-4.1.0-171294.x86_64.iso VMware-VMvisor-
Installer-4.1.0-171294.x86_64.iso
sudo mount -o loop VMware-VMvisor-Installer-4.1.0-171294.x86_64.iso iso
```

Step 2 Using **vi** or some other common Linux editor, add the following lines to the PXE server's PXE config file:

```
label esx41i-install

kernel esx41i/mboot.c32
append esx41i/vmkboot.gz --- esx41i/vmkernel.gz --- esx41i/sys.vgz ---
esx41i/cim.vgz --- esx41i/ienviron.tgz --- esx41i/image.tgz --- esx41i/
install.tgz --- esx41i/xsigo.tgz
```


Note

The xsigo.tgz file is the Xsigo depot file renamed.

Step 3 Save the changes and quit the editor.

Making the New PXE Config Available to Booting Servers

After all modifications are made and the new initrd is on the PXE Server, you will need to flag the newly created initrd so that it can be used to boot any server that has received the PXE boot image. For example, the steps in this section set the options in the kickstart script so that a booting server can receive required install-time settings.

Step 1 Edit the kickstart script to include labels that append the initrd. For example

```
label esx41i-pxe3 #3.5.0 drivers
        kernel vmlinuz-esx-260247
        append initrd=XG-3.5.0-initrd-260247.img mem=768M askmedia
```

Step 2 Close the default file, making sure to preserve the changes.

Loading the Image into the ESXi 4.1 Server

After the PXE Boot vNIC is created in the PXE Boot server profile, and the ISO image is modified, and the PXE server's config file points to the correct files to use for PXE booting, you can now load the ISO image into the ESXi 4.1 server, and boot the server.


Note

The following procedure assumes the installation of the PXE boot image through a "lights out" management solution.

To load the PXE Boot image into the ESXi 4.1 server, follow this procedure:

Step 1 From the ESXi 4.1 server, enter the PXE menu and boot from it. The ESXi 4.1 begins loading all pertinent OS files as shown in Figure 2.

When the `XG-VMware-Visor-Installer` is mounted, you will see output similar to the following.

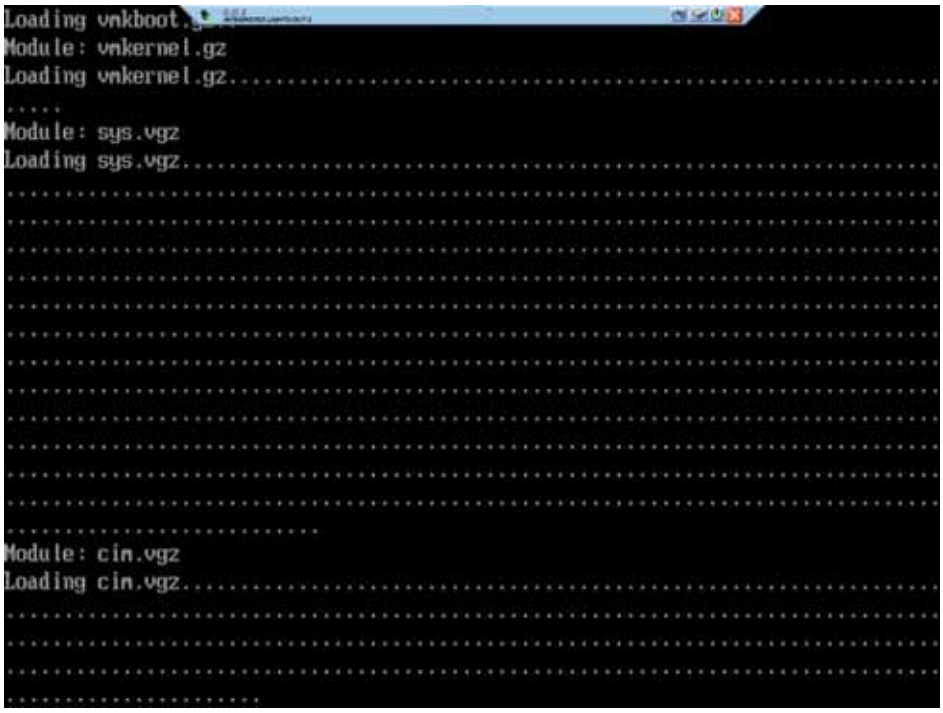


Figure 2 Loading OS Files from Modified ISO Image

Step 2 After the OS files have been loaded, the server boots to the ESXi 4.1 native operating system. See Figure 3 on page 147.

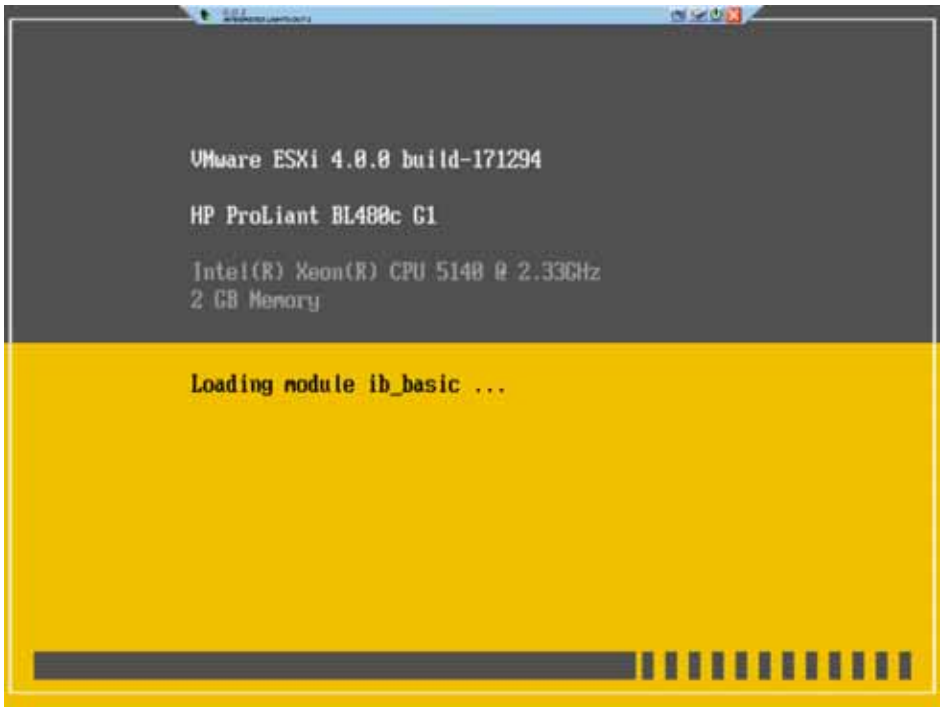


Figure 3 Boot Up

Step 3 As part of the boot sequence, you must read the license agreement.

Step 4 Accept (or Decline) the license agreement as needed. To continue the installation procedure, accept the license agreement and start the ESXi 4.1 installer. Figure 4 on page 148 shows the ESXi 4.1 installer.

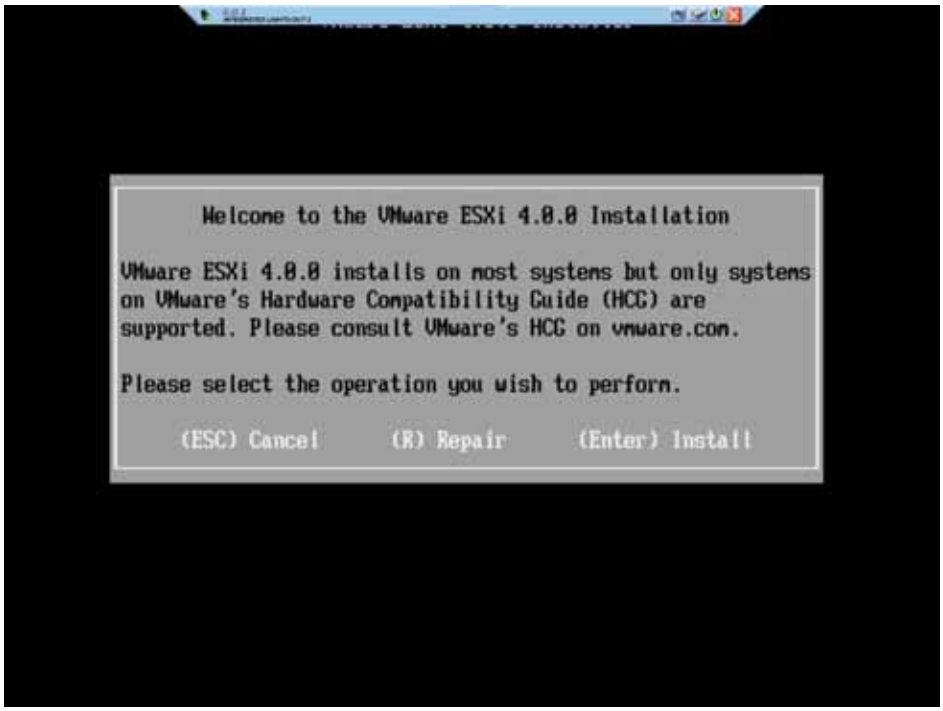


Figure 4 ESXi 4.1 Installer

Follow the installer until you are prompted to specify a boot device on the Select a Disk dialog.

Step 5 On the Select a Disk dialog, select the hard disk on the PXE server. Figure 5 on page 149 shows an example of this dialog.

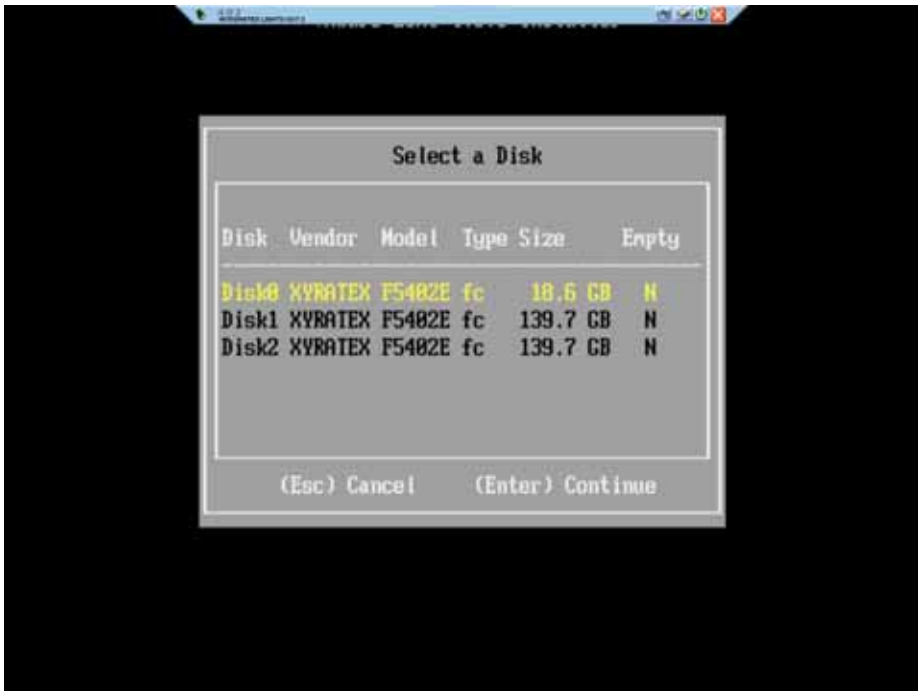


Figure 5 Select a Disk for Booting

Step 6 Press ***Enter*** to continue the installer until you see the Confirm Install dialog. See Figure 6.

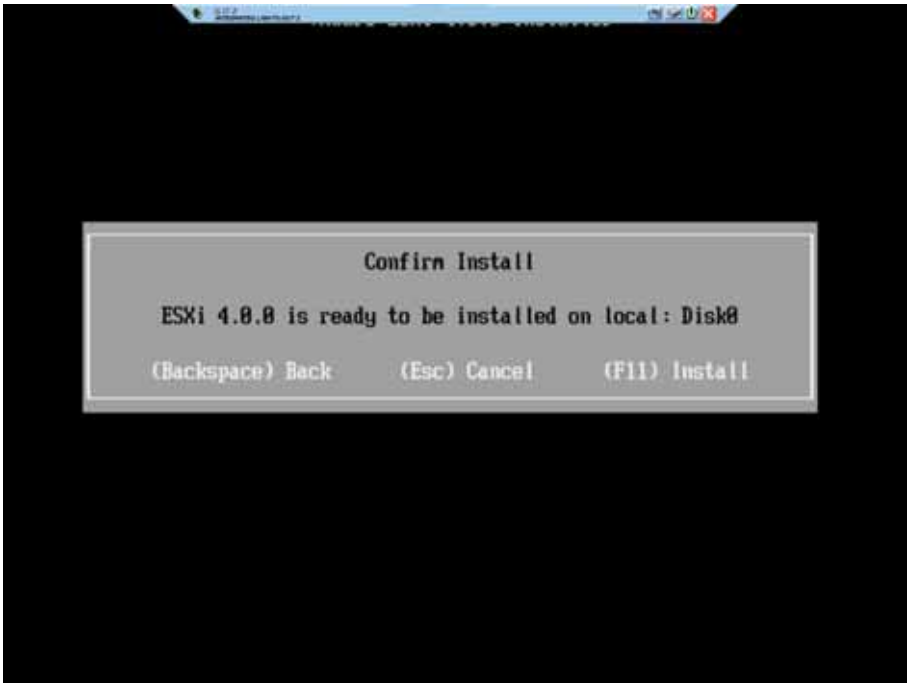


Figure 6 Confirm Install Dialog

Note

Xsigo recommends that the PXE servers hard disk be dedicated to the modified ISO. However, in some cases, this situation might not be possible. If your PXE server's hard disk already contains data, the following dialog will be displayed before the Confirm Install dialog.

If this dialog is displayed, be aware that the data on this disk will be overwritten. You can either overwrite the existing data, or abort the current installation, store the data on a different device, then resume the installation.

Step 7 When the correct boot disk is confirmed, the installation runs to completion. The Installation Complete dialog is displayed. Figure 7 on page 151 shows this dialog.

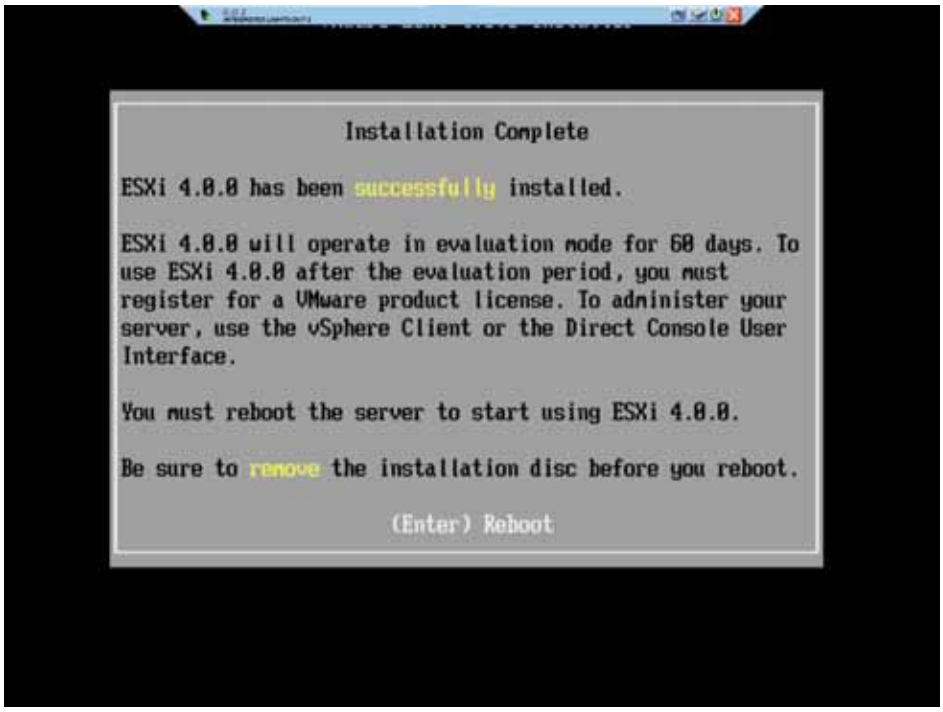


Figure 7 Installation Complete

Step 8 Make sure that any installation medium (CD or DVD) in the ESXi 4.1 server is removed, then allow the server to reboot, as shown in Figure 8 on page 152.

Figure 8 ESXi 4.1 Server Rebooting

Step 9 When the server reboots, it progresses through the boot devices until it locates the PXE server's boot device from which it retrieves the boot image.

Firmware and Option ROM Levels

Whenever you configure a server to boot remotely, you must ensure that the HCA firmware is at the correct version and that the option ROM is installed. This appendix provides instructions for these tasks:

- Linux Firmware and Option ROM
- Windows Firmware and Option ROM

Linux Firmware and Option ROM

To ensure the levels of firmware and option ROM for your Linux server:

Step 1 Log in into the host server as root.

Step 2 Unpack the Xsigo HCA firmware package on the server. For example:

```
rpm -ivh <xsigo-hca-firmware_number.i386.rpm>
```

Note

Replace `xsigo-hca-firmware_2.6.6.i386.rpm` with the xsigo firmware for your server. Supported host drivers for each operating system are listed in the release notes.

This step unpacks the `xg_config` tool, which you can use to update the HCA firmware and Option ROM.

Step 3 Check the firmware and option ROM level:

- Log in as root to the host server.
- Run xg_config to view the firmware and option ROM levels.

```
/opt/xsigo/bin/xg_config
#############################################################################
# Main menu
#############################################################################

Selected card:
   Node GUID          : '0002:c902:0020:4934'
   Board ID           : 'MT_0150000001'
   CA type            : 'MT25208'
   Firmware version   : '5.3.0'
   Hardware version   : 'a0'
   Option ROM version : 'XgBoot Version 2.2.0'
```

Oracle's XgOS supports the following minimum firmware levels:

InfiniHost Single-Port HCA: 1.3.0 or higher

InfiniHost Dual-Port HCA: 5.3.0 or higher

ConnectX Dual-Port HCA: 2.8.0 or higher

ConnectX-2 Single and Dual-Port HCA: Firmware version 2.8.0 or higher

If your firmware and XgBoot versions are as shown above, you can skip Step 4.

Step 4 On your Linux host server, upgrade the HCA firmware and the option ROM if necessary.

- If you haven't already done so, log in as root to the host server.
- Upgrade the Xsigo HCA firmware package on the server. For example:

```
rpm -Uvh <xsigo-hca-firmware_number.i386.rpm>
```

Note

Replace `xsigo-hca-firmware_number.i386.rpm` with the Xsigo host driver for your server. Supported host drivers for each operating system are listed in the release notes.

- Run xg_config to upgrade the firmware and option ROM.

```
/opt/xsigo/bin/xg_config
################################################################################
# Main menu
################################################################################

Selected card:
   Node GUID          : '0002:c902:0020:4934'
   Board ID           : 'MT_0150000001'
   CA type            : 'MT25208'
   Firmware version   : '5.3.0'
   Hardware version   : 'a0'
   Option ROM version : 'XgBoot Version 2.2.11'

1) Flash HCA Firmware
2) Flash HCA Firmware + Option ROM
3) Flash Option ROM
4) Change selected card
0) Quit
Select option>
```

- If you are using SAN Boot or might decide to in the future, select option 2. Otherwise, select option 1. In the following screen output example, option 2 was selected:

```
################################################################################
# Flash HCA Firmware + Option ROM Menu
################################################################################

Selected card:
   Node GUID          : '0002:c902:0020:4934'
   Board ID           : 'MT_0150000001'
   CA type            : 'MT25208'
   Firmware version   : '5.3.0'
   Hardware version   : 'a0'
   Option ROM version : 'XgBoot Version 2.2.11'

1) 5.3.0 (XgBoot Version 1.5)
2) 5.1.400 (XgBoot Version 1.5)
0) Return to previous menu
Select firmware to use>
********************************************************
```

- Select the most recent firmware (the one displayed first).

You will need to reboot for the firmware upgrade to take effect. However, you can wait to reboot until you have upgraded the host drivers.

- For other servers that were not used for remastering the ISO, you can just boot once from the remastered ISO which can be used as a golden master image to boot any number of Citrix Xen 5.6 FP1 servers.

Windows Firmware and Option ROM

Oracle's XgOS supports the following minimum firmware levels:

- Single Port HCA: 1.2.0
- Dual Port HCA: 5.3.0
- Connect-X: 2.6.0

When the `Xg_FWUpdate.vbs` script runs, it first checks the current HCA Device ID and firmware level and determines if an update is required.

Step 1 To run the script, start a command prompt by following ***Start->Run...***.

Step 2 Change directory (**cd**) to `%programfiles%\Xsigo Systems\Support\FirmwareUpdate`, which is the directory where the HCA firmware update script is located. For example:

```
cd %programfiles%\Xsigo Systems\Support\FirmwareUpdate
```

Step 3 From the prompt, run the script by issuing the following command:

```
cscript Xg_FWUpdate.vbs

Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

################################################################################
# (C) 2011 XSIGO SYSTEMS Inc. All rights reserved. This material may not be
# reproduced, displayed, modified or distributed without the express prior
# written permission of the copyright holder.
################################################################################

################################################################################

# Main Menu
################################################################################

Selected HCA Card Number: 0
HCA Device ID : mt25218_pciconf0
Image Type : failsafe
I.S. Version : 1
Device ID :
Chip Revision : a0
GUID Descr : node port1 port2 sys image
GUIDs : 0002c9020021f1f0 0002c9020021f1f1 0002c9020021f1f2 0002c9020021
f1f3
BOARD ID : mt_0370110001
VSD :
PSID : mt_0370110001
FW Version :
HCA mlx FW Ver : 5.1.400
1) Flash HCA Firmware
2) Change selected card
0) Quit
Select option>
```

Step 4 When prompted, enter `1` to enter the Flash HCA Firmware Menu, as shown in the following example:

```
Select option> 1

###############################################################################
# Flash HCA Firmware Menu
###############################################################################
Selected HCA Card Number: 0
HCA Device ID : mt25218_pciconf0
Image Type : failsafe
I.S. Version : 1
Device ID :
Chip Revision : a0
GUID Descr : node             port1           port2            sys image
GUIDs : 0002c902002410a0 0002c902002410a1 0002c902002410a2
0002c902002410a3
BOARD ID : mt_0370110001
VSD :
PSID : mt_0370110001
FW Version :
HCA mlx FW Ver : 5.1.400


1) 5.3.0
2) 5.1.400
0) Return to previous menu

Select Firmware to Burn>
```

Step 5 When prompted, select the firmware version that you want to burn onto the HCA in the Windows server. Do not attempt to abort the firmware upgrade process after it has started. The following example shows updating the HCA with firmware version 5.3.0.

```
Select Firmware to Burn> 1
Upgrading HCA firmware 5.1.400 to 5.3.0
This Will Flash HCA with Firmware file .\Image\fw-25218-5_2_0-mhea28-
xtc_a1-a2.bin
Please do not interrupt the burn process or reboot the machine...
Wait till burn completes ...
.....................................................................
---------------------------------------------------------------------
The firmware on one or more of the HCAs has been upgraded.
It is recommended to reboot the machine in order for changes to take
effect.
---------------------------------------------------------------------
Press Enter key to continue
```

Step 6 Press ***Enter*** to exit the update script.

Step 7 Whenever you run the script and burn firmware on one or more HCAs, shut down the Windows server and then start it to bring HCAs up.

If HCAs have been updated, this cold boot is required bring them online with the new firmware.

Table 1 lists the terms used in this document and other Xsigo documentation. The definitions provided here should only be used in the context of Oracle's Fabric Director and Oracle's Xsigo Operating System (XgOS).

Table 1 Terms and Definitions

Term	Definition
Active Directory	Active Directory (AD) is an implementation of LDAP directory services by Microsoft for use primarily in Windows environments. Its main purpose is to provide central authentication and authorization services for Windows based computers. Active Directory also allows administrators to assign policies, deploy software, and apply critical updates to an organization.
Admin State	Administrative state. The intention of the operator by setting a given resource up or down. See also Oper State.
CPIO	Copy Input Output. A binary file archiver and a file format. CPIO's use by the RPM Package Manager continues to make CPIO an important archive format. See **`man cpio`**.
FC	Fibre Channel. The American National Standards Institute (ANSI) began work on FC in 1988, and since then the X3T11 Task Group (see www.t11.org) has developed 20+ standards. FC has its own stack of protocol levels (layers), ranging from the physical connectors and media (FC-0) to upper-level protocols (FC-4). Each of these levels defines a different and separate part of how the FC equipment communicates. The different FC-4 protocols (FCP, IP, Virtual Interface, and others) are tied directly to different kinds of applications (storage, networking, and clustering) for different uses. For more background information, see www.fibrechannel.org
HA vNIC	High Availability vNIC - A pair of virtual Ethernet interfaces that are both assigned to the same server profile, but bound to different physical interfaces.
HBA	Host Bus Adaptor. A Fibre Channel network interface card used in a SAN fabric. FC HBAs are replacing SCSI HBAs.
HCA	Host Channel Adapter. An InfiniBand network interface card used in an InfiniBand network. An HCA provides high-speed connectivity and virtual interfaces, based on the InfiniBand interface. An HCA can have 1 or 2 ports.
hypervisor	A hypervisor is a virtualization platform that allows multiple guest operating systems to run at the second level above the hardware.
IB	InfiniBand. A switched fabric communications link primarily used in high-performance computing. IB is the result of merging two competing designs, Future I/O, developed by Compaq, IBM, and Hewlett-Packard, with Next Generation I/O (ngio), developed by Intel, Microsoft, and Sun. For more information, see www.infinibandta.org
IDE	Integrated Drive Electronics. Throughout the 1980s, a standard interface for connecting hosts to direct-attached storage devices. Parallel SCSI was another approach.
I/O	Input/Output. In computer architecture, the combination of the CPU and main memory (i.e., memory that the CPU can read and write to directly, with individual instructions) is considered the heart of a computer. Any movement of information to or from that complex, for example to or from a disk drive, is considered I/O.
I/O Module	A physical card that is installed in one of 15 slots in the chassis' card bay. There are two types of I/O module: Ethernet and Host Bus Adapter. The Ethernet and Host Bus Adapter modules provide access to Ethernet and Fibre Channel networks, respectively.

Table 1 (continued) Terms and Definitions

Term	Definition
I/O Port	A single port on an Ethernet module, a Host Bus Adapter module, or one of the 24 InfiniBand server ports.
JBOD	Just A Bunch of Disks. Very large storage arrays, capable of storing terabytes and terabytes of data. Farms of JBODs connect through an FC SAN. In a JBOD each disk is visible to the SAN, assigned an address, and is treated as an autonomous device even though the physical disks are located in the same enclosure.
jitter	For QoS the delta between packets on the receive side. Low jitter is guaranteed by having a low-latency queue mechanism. In this way, a flow is guaranteed service and packets are not held up (delayed) in buffers.
Kerberos	Kerberos is a network authentication protocol. It is designed to provide strong authentication for client/server applications by using secret-key cryptography. Kerberos was developed in the Athena Project at the Massachusetts Institute of Technology (MIT). The name is taken from Greek mythology; Kerberos was a three-headed dog who guarded the gates of Hades. Kerberos lets a user request an encrypted “ticket” from an authentication process that can then be used to request a particular service from a server.
LDAP	The Lightweight Directory Access Protocol (LDAP) is an application protocol for querying and modifying directory services running over TCP/IP. A client starts an LDAP session by connecting to an LDAP server, by default on TCP port 389. The client then sends operation requests to the server, and the server sends responses in turn.
Managed Object	An object-oriented representation of a resource managed in a device. This can be a physical or logical resource.
NAS	Network Attached Storage. NAS uses common client networks, such as Ethernet, to connect client computers to a host file server. Unlike SANs, the client does not directly communicate with the storage. Data exchange occurs at the file level, unlike a SAN where data is operated at the block level over FC.
NPIV	N-Port ID Virtualization, a fibre-channel facility that allows multiple node port IDs to share a single physical node port.
OFED	OpenFabrics Enterprise Distribution. OFED is the driver stack for the InfiniBand Host Channel Adaptor (HCA). For more information, see http://www.openfabrics.org/resources.htm
OpenSM	The default Subnet Manager running on the Xsigo Fabric Director.
Oper State	Operative state. This indicates whether a resources is configured and operating properly. See also Admin State.
Policy	Configuration of automatic system behavior (e.g. stats collection, dB cleanup, etc.).
Quality of Service	The Quality of Service (QoS) object allows the data traffic of individual applications or interfaces to be managed. The performance of a particular application can be guaranteed by raising the priority of its dataflow, relative to the other applications.
RADIUS	Remote Authentication Dial In User Service (RADIUS) is an Authentication, Authorization, and Accounting (AAA) protocol for controlling access to network resources. RADIUS is commonly used by ISPs and corporations managing access to Internet or internal networks across an array of access technologies including modem, DSL, wireless, and VPNs.

Table 1 (continued) Terms and Definitions

Term	Definition
RAID	Redundant Array of Inexpensive Disks.
RDMA	Remote Direct Memory Access. One of the key problems with server I/O is the CPU overhead associated with data movement between memory and I/O devices, such as LAN and SAN interfaces. InfiniBand solves this problem by using RDMA to offload data movement from the server CPU to the InfiniBand HCA. Using RDMA, the sending device either reads data from or writes data to the target devices' user space memory, thereby avoiding CPU interrupts and multiple data copies on the memory bus. This approach enables RDMA to significantly reduce the CPU overhead associated with data movement between nodes.
Role	One of Xsigo's fixed-privilege levels to which a user may be assigned (for example., Operators, Administrators, Storage).
SAN	Fibre Channel Storage Area Network. A SAN is a network of storage and system components, all communicating on a fibre-channel network, that can be used to consolidate and share storage, provide high-performance links to data devices, add redundant links to storage systems, speed up data backup, and support high-availability clustering systems. The advent of SANs has been driven by today's insatiable appetite for storage. See www.snia.org for more background information.
SCSI	Small Computer Systems Interface. In the early 1980s, SCSI was the standard direct-attach storage interface to SCSI-enabled disks. As computer systems increased in speed and data storage needs increased, the parallel bus architecture of SCSI began hitting performance and distance limits. In response to this need, FC was introduced to provide gigabit-speed serial networking capabilities for storage.
Server Profile	One instance of a server I/O configuration that is assignable to a single physical server through an IB port.
State	Displayed in Fabric Manager and the CLI as a pair of statuses, for example: up/up. The first is the Admin State while the second is the Oper State. When using SNMP or the Java APIs, these statuses are returned individually.
User	An internal or external representation of a person. Users either exist locally or remotely via LDAP, Active Directory, or RADIUS. By default, an "admin" user is created locally.
vHBA	Virtual Host Bus Adapter - A Fibre Channel Storage connection, provided without a physical HBA.
VLAN	Virtual Local Area Network - A private, independent, logical networks that are created within a physical network. A VLAN behaves like an ordinary LAN, but connected devices don't have to be physically connected to the same network segment.
VM	Virtual Machine. A VM is a software entity that runs its own operating systems and applications, as if it were a physical computer. A VM behaves exactly like a physical computer and contains its own virtual (software based) CPU, RAM, hard disk, and NIC. An operating system installed on a VM is called a guest operating system.
vNIC	Virtual Network Interface Card - An Ethernet interface, provided without a physical NIC.
WWNN	World Wide Node Name
WWPN	World Wide Port Name

B

C

D

E

F

G

I

K

L

M

N

O

P

R

S

T

U

V

W

X