

vSphere Storage

ESXi 6.0
vCenter Server 6.0

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see <http://www.vmware.com/support/pubs>.

EN-001522-03

vmware[®]

You can find the most up-to-date technical documentation on the VMware Web site at:

<http://www.vmware.com/support/>

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

Copyright © 2009–2015 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

About vSphere Storage	9
Updated Information	11
1 Introduction to Storage	13
Storage Virtualization	13
Types of Physical Storage	14
Target and Device Representations	17
Storage Device Characteristics	18
Supported Storage Adapters	20
Datastore Characteristics	21
How Virtual Machines Access Storage	24
Comparing Types of Storage	24
2 Overview of Using ESXi with a SAN	27
ESXi and SAN Use Cases	28
Specifics of Using SAN Storage with ESXi	28
ESXi Hosts and Multiple Storage Arrays	29
Making LUN Decisions	29
Choosing Virtual Machine Locations	30
Layered Applications	31
Third-Party Management Applications	32
SAN Storage Backup Considerations	32
3 Using ESXi with Fibre Channel SAN	35
Fibre Channel SAN Concepts	35
Using Zoning with Fibre Channel SANs	36
How Virtual Machines Access Data on a Fibre Channel SAN	37
4 Configuring Fibre Channel Storage	39
ESXi Fibre Channel SAN Requirements	39
Installation and Setup Steps	40
N-Port ID Virtualization	41
5 Configuring Fibre Channel over Ethernet	45
Fibre Channel over Ethernet Adapters	45
Configuration Guidelines for Software FCoE	46
Set Up Networking for Software FCoE	46
Add Software FCoE Adapters	47

- 6 Booting ESXi from Fibre Channel SAN 49**
 - Boot from SAN Benefits 49
 - Boot from Fibre Channel SAN Requirements and Considerations 50
 - Getting Ready for Boot from SAN 50
 - Configure Emulex HBA to Boot from SAN 52
 - Configure QLogic HBA to Boot from SAN 53

- 7 Booting ESXi with Software FCoE 55**
 - Requirements and Considerations for Software FCoE Boot 55
 - Best Practices for Software FCoE Boot 56
 - Set Up Software FCoE Boot 56
 - Troubleshooting Installation and Boot from Software FCoE 57

- 8 Best Practices for Fibre Channel Storage 59**
 - Preventing Fibre Channel SAN Problems 59
 - Disable Automatic Host Registration 60
 - Optimizing Fibre Channel SAN Storage Performance 60

- 9 Using ESXi with iSCSI SAN 63**
 - iSCSI SAN Concepts 63
 - How Virtual Machines Access Data on an iSCSI SAN 68

- 10 Configuring iSCSI Adapters and Storage 69**
 - ESXi iSCSI SAN Requirements 70
 - ESXi iSCSI SAN Restrictions 70
 - Setting LUN Allocations for iSCSI 70
 - Network Configuration and Authentication 71
 - Set Up Independent Hardware iSCSI Adapters 71
 - About Dependent Hardware iSCSI Adapters 74
 - About the Software iSCSI Adapter 81
 - Modify General Properties for iSCSI Adapters 87
 - Setting Up iSCSI Network 88
 - Using Jumbo Frames with iSCSI 95
 - Configuring Discovery Addresses for iSCSI Adapters 97
 - Configuring CHAP Parameters for iSCSI Adapters 98
 - Configuring Advanced Parameters for iSCSI 102
 - iSCSI Session Management 103

- 11 Booting from iSCSI SAN 107**
 - General Boot from iSCSI SAN Recommendations 107
 - Prepare the iSCSI SAN 108
 - Configure Independent Hardware iSCSI Adapter for SAN Boot 108
 - iBFT iSCSI Boot Overview 109

- 12 Best Practices for iSCSI Storage 115**
 - Preventing iSCSI SAN Problems 115
 - Optimizing iSCSI SAN Storage Performance 116
 - Checking Ethernet Switch Statistics 119

- 13 Managing Storage Devices 121**
 - Storage Device Characteristics 121
 - Understanding Storage Device Naming 123
 - Storage Refresh and Rescan Operations 124
 - Identifying Device Connectivity Problems 126
 - Edit Configuration File Parameters 131
 - Enable or Disable the Locator LED on Storage Devices 132

- 14 Working with Flash Devices 133**
 - Using Flash Devices 134
 - Marking Storage Devices 134
 - Monitor Flash Devices 136
 - Best Practices for Flash Devices 136
 - About Virtual Flash Resource 137
 - Configuring Host Swap Cache 139

- 15 About VMware vSphere Flash Read Cache 141**
 - DRS Support for Flash Read Cache 142
 - vSphere High Availability Support for Flash Read Cache 142
 - Configure Flash Read Cache for a Virtual Machine 142
 - Migrating Virtual Machines with Flash Read Cache 143

- 16 Working with Datastores 145**
 - Understanding VMFS Datastores 146
 - Understanding Network File System Datastores 152
 - Creating Datastores 160
 - Managing Duplicate VMFS Datastores 162
 - Upgrading VMFS Datastores 164
 - Increasing VMFS Datastore Capacity 165
 - Administrative Operations for Datastores 167
 - Set Up Dynamic Disk Mirroring 173
 - Collecting Diagnostic Information for ESXi Hosts on a Storage Device 174
 - Checking Metadata Consistency with VOMA 177
 - Configuring VMFS Pointer Block Cache 179

- 17 Understanding Multipathing and Failover 181**
 - Failover with Fibre Channel 181
 - Host-Based Failover with iSCSI 182
 - Array-Based Failover with iSCSI 184
 - Path Failover and Virtual Machines 185
 - Managing Multiple Paths 186
 - VMware Multipathing Module 187
 - Path Scanning and Claiming 189
 - Managing Storage Paths and Multipathing Plug-Ins 192
 - Scheduling Queues for Virtual Machine I/Os 200

- 18 Raw Device Mapping 203**
 - About Raw Device Mapping 203
 - Raw Device Mapping Characteristics 206
 - Create Virtual Machines with RDMs 208
 - Manage Paths for a Mapped LUN 209

- 19 Working with Virtual Volumes 211**
 - Virtual Volumes Concepts 212
 - Guidelines when Using Virtual Volumes 215
 - Virtual Volumes and Storage Protocols 215
 - Virtual Volumes Architecture 216
 - Before You Enable Virtual Volumes 217
 - Configure Virtual Volumes 218
 - Provision Virtual Machines on Virtual Datastores 221

- 20 Virtual Machine Storage Policies 225**
 - Upgrading Legacy Storage Profiles 225
 - Understanding Virtual Machine Storage Policies 226
 - Working with Virtual Machine Storage Policies 228
 - Creating and Managing VM Storage Policies 228
 - Storage Policies and Virtual Machines 233

- 21 VMkernel and Storage 239**
 - Storage APIs 240

- 22 Storage Hardware Acceleration 243**
 - Hardware Acceleration Benefits 243
 - Hardware Acceleration Requirements 244
 - Hardware Acceleration Support Status 244
 - Hardware Acceleration for Block Storage Devices 244
 - Hardware Acceleration on NAS Devices 249
 - Hardware Acceleration Considerations 251

- 23 Storage Thick and Thin Provisioning 253**
 - Storage Over-Subscription 253
 - Virtual Disk Thin Provisioning 253
 - Array Thin Provisioning and VMFS Datastores 257

- 24 Using Storage Providers 261**
 - Storage Providers and Storage Data Representation 262
 - Storage Provider Requirements and Considerations 262
 - Storage Status Reporting 263
 - Register Storage Providers 263
 - Securing Communication with Storage Providers 264
 - View Storage Provider Information 264
 - Unregister Storage Providers 264
 - Update Storage Providers 265

25 Using vmkfstools 267
 vmkfstools Command Syntax 267
 vmkfstools Options 268

Index 277

About vSphere Storage

vSphere Storage describes storage options available to VMware[®] ESXi and explains how to configure your ESXi system so that it can use and manage different types of storage. In addition, *vSphere Storage* explicitly concentrates on Fibre Channel[®] and iSCSI storage area networks (SANs) as storage options and discusses specifics of using ESXi in Fibre Channel and iSCSI environments.

Intended Audience

This information is for experienced system administrators who are familiar with virtual machine technology, data center operations, and SAN storage concepts.

Updated Information

This *vSphere Storage* is updated with each release of the product or when necessary.

This table provides the update history of the *vSphere Storage*.

Revision	Description
EN-001522-03	“Use esxcli Commands to Enable or Disable Per File I/O Scheduling,” on page 201 has been added.
EN-001522-02	<ul style="list-style-type: none">Information related to Storage Views and Storage Reports has been removed. This functionality is deprecated in vSphere 6.0.“Set Up a File as Core Dump Location,” on page 175 has been updated to clarify command syntax.“Deactivate and Delete a Core Dump File,” on page 177 has been added.
EN-001522-01	“Change the Number of Scanned Storage Devices,” on page 125 has been updated to include new details.
EN-001522-00	Initial release.

Introduction to Storage

This introduction describes storage options available in vSphere and explains how to configure your host so that it can use and manage different types of storage.

This chapter includes the following topics:

- [“Storage Virtualization,”](#) on page 13
- [“Types of Physical Storage,”](#) on page 14
- [“Target and Device Representations,”](#) on page 17
- [“Storage Device Characteristics,”](#) on page 18
- [“Supported Storage Adapters,”](#) on page 20
- [“Datastore Characteristics,”](#) on page 21
- [“How Virtual Machines Access Storage,”](#) on page 24
- [“Comparing Types of Storage,”](#) on page 24

Storage Virtualization

vSphere storage virtualization supports such capabilities as virtual machines, Virtual SAN, Virtual Volumes, policy-based storage managements, and so on.

ESXi provides host-level storage virtualization, which logically abstracts the physical storage layer from virtual machines. An ESXi virtual machine uses a virtual disk to store its operating system, program files, and other data associated with its activities. A virtual disk is a large physical file, or a set of files, that can be copied, moved, archived, and backed up as easily as any other file. You can configure virtual machines with multiple virtual disks.

To access virtual disks, a virtual machine uses virtual SCSI controllers. These virtual controllers include BusLogic Parallel, LSI Logic Parallel, LSI Logic SAS, and VMware Paravirtual. These controllers are the only types of SCSI controllers that a virtual machine can see and access.

Each virtual disk resides on a datastore that is deployed on physical storage. From the standpoint of the virtual machine, each virtual disk appears as if it were a SCSI drive connected to a SCSI controller. Whether the actual physical storage is being accessed through storage or network adapters on the host is typically transparent to the guest operating system and to applications running on the virtual machine.

In addition to virtual disks, vSphere offers a mechanism called raw device mapping (RDM). RDM is useful when a guest operating system inside a virtual machine requires direct access to a storage device. For information about RDMs, see [Chapter 18, “Raw Device Mapping,”](#) on page 203.

Other storage virtualization capabilities that vSphere provides include Virtual SAN, Virtual Flash, Virtual Volumes, and policy-based storage management. For information about Virtual SAN, see the *Administering VMware Virtual SAN*.

Types of Physical Storage

The ESXi storage management process starts with storage space that your storage administrator preallocates on different storage systems.

ESXi supports the following types of storage:

Local Storage	Stores virtual machine files on internal or directly connected external storage disks.
Networked Storage	Stores virtual machine files on external storage disks or arrays attached to your host through a direct connection or through a high-speed network.

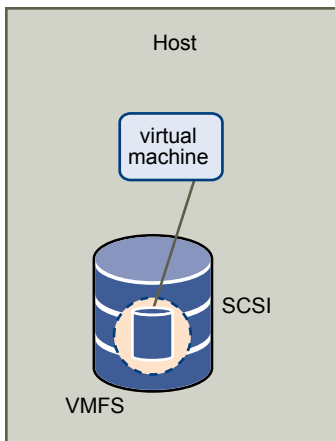
Local Storage

Local storage can be internal hard disks located inside your ESXi host, or it can be external storage systems located outside and connected to the host directly through protocols such as SAS or SATA.

Local storage does not require a storage network to communicate with your host. You need a cable connected to the storage unit and, when required, a compatible HBA in your host.

The following illustration depicts a virtual machine using local SCSI storage.

Figure 1-1. Local Storage



In this example of a local storage topology, the host uses a single connection to a storage disk. On that disk, you can create a VMFS datastore, which you use to store virtual machine disk files.

Although this storage configuration is possible, it is not a recommended topology. Using single connections between storage arrays and hosts creates single points of failure (SPOF) that can cause interruptions when a connection becomes unreliable or fails. However, because the majority of local storage devices do not support multiple connections, you cannot use multiple paths to access local storage.

ESXi supports a variety of local storage devices, including SCSI, IDE, SATA, USB, and SAS storage systems. Regardless of the type of storage you use, your host hides a physical storage layer from virtual machines.

NOTE You cannot use IDE/ATA or USB drives to store virtual machines.

Local storage does not support sharing across multiple hosts. Only one host has access to a datastore on a local storage device. As a result, although you can use local storage to create virtual machines, it prevents you from using VMware features that require shared storage, such as HA and vMotion.

However, if you use a cluster of hosts that have just local storage devices, you can implement Virtual SAN. Virtual SAN transforms local storage resources into software-defined shared storage and allows you to use features that require shared storage. For details, see the *Administering VMware Virtual SAN* documentation.

Networked Storage

Networked storage consists of external storage systems that your ESXi host uses to store virtual machine files remotely. Typically, the host accesses these systems over a high-speed storage network.

Networked storage devices are shared. Datastores on networked storage devices can be accessed by multiple hosts concurrently. ESXi supports multiple networked storage technologies.

In addition to traditional networked storage that this topic covers, VMware supports virtualized shared storage, such as Virtual SAN. Virtual SAN transforms internal storage resources of your ESXi hosts into shared storage that provides such capabilities as High Availability and vMotion for virtual machines. For details, see the *Administering VMware Virtual SAN* documentation.

NOTE The same LUN cannot be presented to an ESXi host or multiple hosts through different storage protocols. To access the LUN, hosts must always use a single protocol, for example, either Fibre Channel only or iSCSI only.

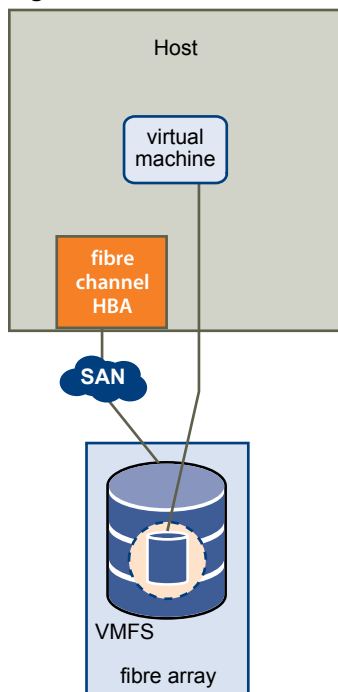
Fibre Channel (FC)

Stores virtual machine files remotely on an FC storage area network (SAN). FC SAN is a specialized high-speed network that connects your hosts to high-performance storage devices. The network uses Fibre Channel protocol to transport SCSI traffic from virtual machines to the FC SAN devices.

To connect to the FC SAN, your host should be equipped with Fibre Channel host bus adapters (HBAs). Unless you use Fibre Channel direct connect storage, you need Fibre Channel switches to route storage traffic. If your host contains FCoE (Fibre Channel over Ethernet) adapters, you can connect to your shared Fibre Channel devices by using an Ethernet network.

Fibre Channel Storage depicts virtual machines using Fibre Channel storage.

Figure 1-2. Fibre Channel Storage



In this configuration, a host connects to a SAN fabric, which consists of Fibre Channel switches and storage arrays, using a Fibre Channel adapter. LUNs from a storage array become available to the host. You can access the LUNs and create datastores for your storage needs. The datastores use the VMFS format.

For specific information on setting up the Fibre Channel SAN, see [Chapter 3, “Using ESXi with Fibre Channel SAN,”](#) on page 35.

Internet SCSI (iSCSI)

Stores virtual machine files on remote iSCSI storage devices. iSCSI packages SCSI storage traffic into the TCP/IP protocol so that it can travel through standard TCP/IP networks instead of the specialized FC network. With an iSCSI connection, your host serves as the initiator that communicates with a target, located in remote iSCSI storage systems.

ESXi offers the following types of iSCSI connections:

Hardware iSCSI

Your host connects to storage through a third-party adapter capable of offloading the iSCSI and network processing. Hardware adapters can be dependent and independent.

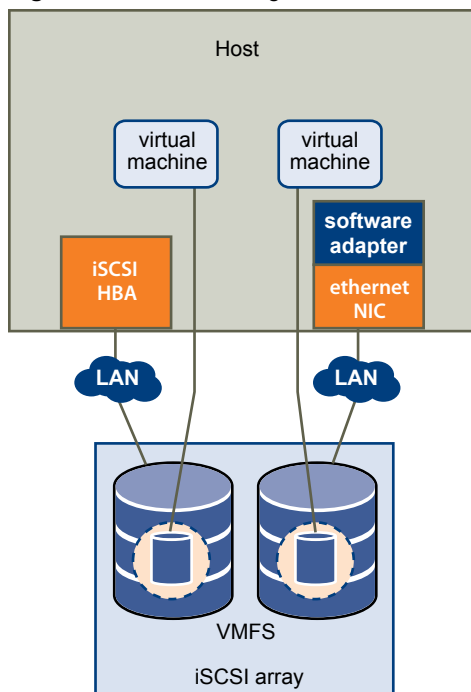
Software iSCSI

Your host uses a software-based iSCSI initiator in the VMkernel to connect to storage. With this type of iSCSI connection, your host needs only a standard network adapter for network connectivity.

You must configure iSCSI initiators for the host to access and display iSCSI storage devices.

iSCSI Storage depicts different types of iSCSI initiators.

Figure 1-3. iSCSI Storage



In the left example, the host uses the hardware iSCSI adapter to connect to the iSCSI storage system.

In the right example, the host uses a software iSCSI adapter and an Ethernet NIC to connect to the iSCSI storage.

iSCSI storage devices from the storage system become available to the host. You can access the storage devices and create VMFS datastores for your storage needs.

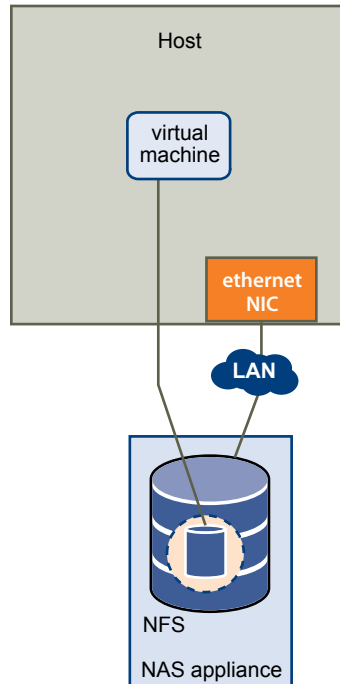
For specific information on setting up the iSCSI SAN, see [Chapter 9, “Using ESXi with iSCSI SAN,”](#) on page 63.

Network-attached Storage (NAS)

Stores virtual machine files on remote file servers accessed over a standard TCP/IP network. The NFS client built into ESXi uses Network File System (NFS) protocol version 3 and 4.1 to communicate with the NAS/NFS servers. For network connectivity, the host requires a standard network adapter.

NFS Storage depicts a virtual machine using the NFS volume to store its files. In this configuration, the host connects to the NFS server, which stores the virtual disk files, through a regular network adapter.

Figure 1-4. NFS Storage



For specific information on setting up NFS storage, see [“Understanding Network File System Datastores,”](#) on page 152.

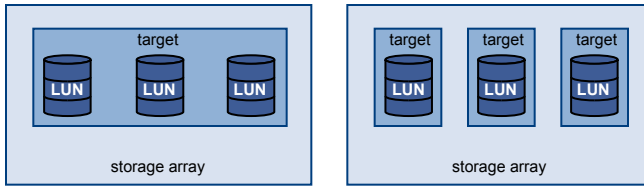
Shared Serial Attached SCSI (SAS)

Stores virtual machines on direct-attached SAS storage systems that offer shared access to multiple hosts. This type of access permits multiple hosts to access the same VMFS datastore on a LUN.

Target and Device Representations

In the ESXi context, the term target identifies a single storage unit that the host can access. The terms device and LUN describe a logical volume that represents storage space on a target. Typically, the terms device and LUN, in the ESXi context, mean a storage volume presented to the host from a storage target and available for formatting.

Different storage vendors present the storage systems to ESXi hosts in different ways. Some vendors present a single target with multiple storage devices or LUNs on it, while others present multiple targets with one LUN each.

Figure 1-5. Target and LUN Representations

In this illustration, three LUNs are available in each configuration. In one case, the host sees one target, but that target has three LUNs that can be used. Each LUN represents an individual storage volume. In the other example, the host sees three different targets, each having one LUN.

Targets that are accessed through the network have unique names that are provided by the storage systems. The iSCSI targets use iSCSI names, while Fibre Channel targets use World Wide Names (WWNs).

NOTE ESXi does not support accessing the same LUN through different transport protocols, such as iSCSI and Fibre Channel.

A device, or LUN, is identified by its UUID name. If a LUN is shared by multiple hosts, it must be presented to all hosts with the same UUID.

Storage Device Characteristics

You can display all storage devices or LUNs available to the host, including all local and networked devices. If you use third-party multipathing plug-ins, the storage devices available through the plug-ins also appear on the list.

For each storage adapter, you can display a separate list of storage devices available for this adapter.

Generally, when you review storage devices, you see the following information.

Table 1-1. Storage Device Information

Storage Device Information	Description
Name	Also called Display Name. It is a name that the ESXi host assigns to the device based on the storage type and manufacturer. You can change this name to a name of your choice.
Identifier	A universally unique identifier that is intrinsic to the device.
Operational State	Indicates whether the device is mounted or unmounted. For details, see “Detach Storage Devices,” on page 128.
LUN	Logical Unit Number (LUN) within the SCSI target. The LUN number is provided by the storage system. If a target has only one LUN, the LUN number is always zero (0).
Type	Type of device, for example, disk or CD-ROM.
Drive Type	Information about whether the device is a flash drive or a regular HDD drive. For information about flash drives, see Chapter 14, “Working with Flash Devices,” on page 133.
Transport	Transportation protocol your host uses to access the device. The protocol depends on the type of storage being used. See “Types of Physical Storage,” on page 14.
Capacity	Total capacity of the storage device.
Owner	The plug-in, such as the NMP or a third-party plug-in, that the host uses to manage paths to the storage device. For details, see “Managing Multiple Paths,” on page 186.
Hardware Acceleration	Information about whether the storage device assists the host with virtual machine management operations. The status can be Supported, Not Supported, or Unknown. For details, see Chapter 22, “Storage Hardware Acceleration,” on page 243.
Location	A path to the storage device in the <code>/vmfs/devices/</code> directory.

Table 1-1. Storage Device Information (Continued)

Storage Device Information	Description
Partition Format	A partition scheme used by the storage device. It could be of a master boot record (MBR) or GUID partition table (GPT) format. The GPT devices can support datastores greater than 2TB. For more information, see “VMFS Datastores and Storage Disk Formats,” on page 147.
Partitions	Primary and logical partitions, including a VMFS datastore, if configured.
Multipathing Policies (VMFS datastores)	Path Selection Policy and Storage Array Type Policy the host uses to manage paths to storage. For more information, see Chapter 17, “Understanding Multipathing and Failover,” on page 181.
Paths (VMFS datastores)	Paths used to access storage and their status.

Display Storage Devices for a Host

Display all storage devices available to a host. If you use any third-party multipathing plug-ins, the storage devices available through the plug-ins also appear on the list.

The Storage Devices view allows you to list the hosts' storage devices, analyze their information, and modify properties.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Devices**.

All storage devices available to the host are listed under Storage Devices.

- 4 To view details for a specific device, select the device from the list.
- 5 Use tabs under Device Details to access additional information and modify properties for the selected device.

Tab	Description
Properties	View device properties and characteristics. View and modify multipathing policies for the device.
Paths	Display paths available for the device. Disable or enable a selected path.

Display Storage Devices for an Adapter

Display a list of storage devices accessible through a specific storage adapter on the host.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click Storage Adapters.

All storage adapters installed on the host are listed under Storage Adapters.

- 4 Select the adapter from the list and click the **Devices** tab.

Storage devices that the host can access through the adapter are displayed.

Supported Storage Adapters

Storage adapters provide connectivity for your ESXi host to a specific storage unit or network.

ESXi supports different classes of adapters, including SCSI, iSCSI, RAID, Fibre Channel, Fibre Channel over Ethernet (FCoE), and Ethernet. ESXi accesses the adapters directly through device drivers in the VMkernel.

Depending on the type of storage you use, you might need to enable and configure a storage adapter on your host.

For information on setting up software FCoE adapters, see [Chapter 5, “Configuring Fibre Channel over Ethernet,”](#) on page 45.

For information on configuring different types of iSCSI adapters, see [Chapter 10, “Configuring iSCSI Adapters and Storage,”](#) on page 69.

Storage Adapter Characteristics

The host uses storage adapters to access different storage devices. You can display details for the available storage adapters and review their information.

You must enable certain adapters, for example software iSCSI or FCoE, before you can view their information.

Table 1-2. Storage Adapter Information

Adapter Information	Description
Model	Model of the adapter.
Targets (Fibre Channel and SCSI)	Number of targets accessed through the adapter.
Connected Targets (iSCSI)	Number of connected targets on an iSCSI adapter.
WWN (Fibre Channel)	World Wide Name formed according to Fibre Channel standards that uniquely identifies the FC adapter.
iSCSI Name (iSCSI)	Unique name formed according to iSCSI standards that identifies the iSCSI adapter.
iSCSI Alias (iSCSI)	A friendly name used instead of the iSCSI name.
IP Address (independent hardware iSCSI)	Address assigned to the iSCSI HBA.
Devices	All storage devices or LUNs the adapter can access.
Paths	All paths the adapter uses to access storage devices.
Properties	Link that indicates that the adapter requires additional configuration. iSCSI and FCoE adapters display this link.

View Storage Adapters Information

Display storage adapters that your host uses and review their information.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters**.
- 4 To view details for a specific adapter, select the adapter from the list.

Datastore Characteristics

Datastores are logical containers, analogous to file systems, that hide specifics of each storage device and provide a uniform model for storing virtual machine files. You can display all datastores available to your hosts and analyze their properties.

Datastores are added to vCenter Server in the following ways:

- You can create a VMFS5 datastore, an NFS version 3 or 4.1 datastore, or a virtual datastore using the New Datastore wizard. A Virtual SAN datastore is automatically created when you enable Virtual SAN.
- When you add a host to vCenter Server, all datastores on the host are added to vCenter Server.

The following table describes datastore details that you can see when you review datastores through the vSphere Web Client. Certain characteristic might not be available or applicable to all types of datastores.

Table 1-3. Datastore Information

Datastore Information	Applicable Datastore Type	Description
Name	VMFS NFS Virtual SAN VVOL	Editable name that you assign to a datastore. For information on renaming a datastore, see “Change Datastore Name,” on page 167.
File System Type	VMFS NFS Virtual SAN VVOL	File system that the datastore uses. For information about VMFS and NFS datastores and how to manage them, see Chapter 16, “Working with Datastores,” on page 145. For information about Virtual SAN datastores, see the <i>Administering VMware Virtual SAN</i> documentation. For information about Virtual Volumes, see Chapter 19, “Working with Virtual Volumes,” on page 211.
Device Backing	VMFS NFS Virtual SAN	Information about underlying storage, such as a storage device on which the datastore is deployed (VMFS), server and folder (NFS), or disk groups (Virtual SAN) .
Protocol Endpoints	VVOL	Information about corresponding protocol endpoints. See “Protocol Endpoints,” on page 214.
Extents	VMFS	Individual extents that the datastore spans and their capacity.
Drive Type	VMFS	Type of underlying storage device, a flash drive or a regular HHD drive. For details, see Chapter 14, “Working with Flash Devices,” on page 133.
Capacity	VMFS NFS Virtual SAN VVOL	Includes total capacity, provisioned space, and free space.
Mount Point	VMFS NFS Virtual SAN VVOL	A path to the datastore in the host's /vmfs/volumes/ directory.

Table 1-3. Datastore Information (Continued)

Datastore Information	Applicable Datastore Type	Description
Capability Sets	VMFS NOTE A multi-extent VMFS datastore assumes capabilities of only one of its extents. NFS Virtual SAN	Information about storage data services that the underlying storage entity provides. You cannot modify them.
Storage I/O Control	VMFS NFS	Information on whether cluster-wide storage I/O prioritization is enabled. See the <i>vSphere Resource Management</i> documentation.
Hardware Acceleration	VMFS NFS Virtual SAN VVOL	Information on whether the underlying storage entity supports hardware acceleration. The status can be Supported, Not Supported, or Unknown. For details, see Chapter 22, “Storage Hardware Acceleration,” on page 243. NOTE NFS 4.1 does not support Hardware Acceleration.
Tags	VMFS NFS Virtual SAN VVOL	Datastore capabilities that you define and associate with datastores in a form of tags. For information, see “Storage Policies and Rules,” on page 226.
Connectivity with Hosts	VMFS NFS VVOL	Hosts where the datastore is mounted.
Multipathing	VMFS VVOL	Path selection policy the host uses to access storage. For more information, see Chapter 17, “Understanding Multipathing and Failover,” on page 181.




Display Datastore Information



Access the Datastores view with the vSphere Web Client navigator. The Datastores view that you display through the navigator lets you list all datastores available in the vSphere infrastructure inventory, analyze the information, and modify properties. You can also use the view to create datastores.

To list datastores for a particular parent object, such as a data center, cluster, or host, see [“List Datastores for an Infrastructure Object,”](#) on page 23.

Procedure

- 1 In the vSphere Web Client navigator, select **vCenter Inventory Lists > Datastores**.
Datastores that are available in the inventory appear in the center Datastores panel.
- 2 Use the icons to create a datastore or to perform basic tasks for a selected datastore.

Icon	Description
	Create a datastore.
	Increase datastore capacity.
	Mount a datastore to certain hosts.

Icon	Description
	Remove a datastore.
	Unmount a datastore from certain hosts.

- To view specific datastore details, click a selected datastore.
- Use tabs to access additional information and modify datastore properties.

Tab	Description
Getting Started	View introductory information and access basic actions.
Summary	View statistics and configuration for the selected datastore.
Monitor	View alarms, performance data, resource allocation, events, and other status information for the datastore.
Manage	View and modify datastore properties, alarm definitions, tags, and permissions. Use this tab to access storage devices that back the datastore, and to view and edit multipathing details for the datastore devices.
Related Objects	View objects related to the datastore. The objects include virtual machines that reside on the datastore and hosts where the datastore is mounted.

List Datastores for an Infrastructure Object






Display datastores for a specific parent object, such as a data center, cluster, or host.

Procedure

- Use the vSphere Web Client object navigator to browse to an object that is a valid parent object of a datastore, such as a data center, cluster, or host.
- Click the **Related Objects** tab and click **Datastores**.

If any datastores are configured for this object, they appear in the center Datastores panel.

- Use the icons to create a datastore or to perform basic tasks for a selected datastore.

Icon	Description
	Create a datastore.
	Increase datastore capacity.
	Mount a datastore to certain hosts.
	Remove a datastore.
	Unmount a datastore from certain hosts.

- Use tabs to access additional information and modify datastore properties.

Tab	Description
Getting Started	View introductory information and access basic actions.
Summary	View statistics and configuration for the selected datastore.
Monitor	View alarms, performance data, resource allocation, events, and other status information for the datastore.

Tab	Description
Manage	View and modify datastore properties, alarm definitions, tags, and permissions. Use this tab to access storage devices that back the datastore, and to view and edit multipathing details for the datastore devices.
Related Objects	View objects related to the datastore. The objects include virtual machines that reside on the datastore and hosts where the datastore is mounted.

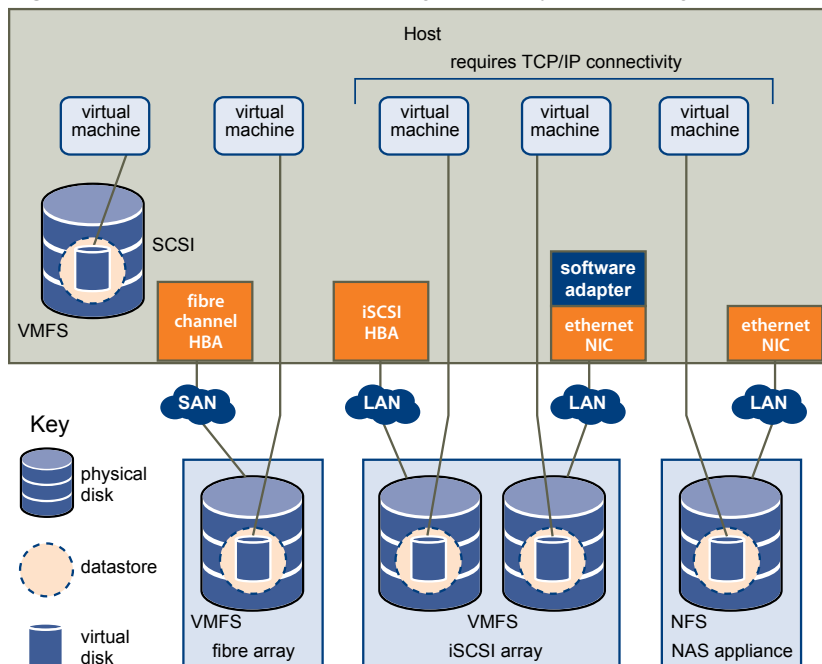
How Virtual Machines Access Storage

When a virtual machine communicates with its virtual disk stored on a datastore, it issues SCSI commands. Because datastores can exist on various types of physical storage, these commands are encapsulated into other forms, depending on the protocol that the ESXi host uses to connect to a storage device.

ESXi supports Fibre Channel (FC), Internet SCSI (iSCSI), Fibre Channel over Ethernet (FCoE), and NFS protocols. Regardless of the type of storage device your host uses, the virtual disk always appears to the virtual machine as a mounted SCSI device. The virtual disk hides a physical storage layer from the virtual machine's operating system. This allows you to run operating systems that are not certified for specific storage equipment, such as SAN, inside the virtual machine.

The following graphic depicts five virtual machines using different types of storage to illustrate the differences between each type.

Figure 1-6. Virtual machines accessing different types of storage



NOTE This diagram is for conceptual purposes only. It is not a recommended configuration.

Comparing Types of Storage

Whether certain vSphere functionality is supported might depend on the storage technology that you use.

The following table compares networked storage technologies that ESXi supports.

Table 1-4. Networked Storage that ESXi Supports

Technology	Protocols	Transfers	Interface
Fibre Channel	FC/SCSI	Block access of data/LUN	FC HBA
Fibre Channel over Ethernet	FCoE/SCSI	Block access of data/LUN	<ul style="list-style-type: none"> ■ Converged Network Adapter (hardware FCoE) ■ NIC with FCoE support (software FCoE)
iSCSI	IP/SCSI	Block access of data/LUN	<ul style="list-style-type: none"> ■ iSCSI HBA or iSCSI-enabled NIC (hardware iSCSI) ■ Network adapter (software iSCSI)
NAS	IP/NFS	File (no direct LUN access)	Network adapter

The following table compares the vSphere features that different types of storage support.

Table 1-5. vSphere Features Supported by Storage

Storage Type	Boot VM	vMotion	Datastore	RDM	VM Cluster	VMware HA and DRS	Storage APIs - Data Protection
Local Storage	Yes	No	VMFS	No	Yes	No	Yes
Fibre Channel	Yes	Yes	VMFS	Yes	Yes	Yes	Yes
iSCSI	Yes	Yes	VMFS	Yes	Yes	Yes	Yes
NAS over NFS	Yes	Yes	NFS 3 and NFS 4.1	No	No	Yes	Yes

NOTE Local storage supports a cluster of virtual machines on a single host (also known as a cluster in a box). A shared virtual disk is required. For more information about this configuration, see the *vSphere Resource Management* documentation.

Overview of Using ESXi with a SAN

Using ESXi with a SAN improves flexibility, efficiency, and reliability. Using ESXi with a SAN also supports centralized management, failover, and load balancing technologies.

The following are benefits of using ESXi with a SAN:

- You can store data securely and configure multiple paths to your storage, eliminating a single point of failure.
- Using a SAN with ESXi systems extends failure resistance to the server. When you use SAN storage, all applications can instantly be restarted on another host after the failure of the original host.
- You can perform live migration of virtual machines using VMware vMotion.
- Use VMware High Availability (HA) in conjunction with a SAN to restart virtual machines in their last known state on a different server if their host fails.
- Use VMware Fault Tolerance (FT) to replicate protected virtual machines on two different hosts. Virtual machines continue to function without interruption on the secondary host if the primary one fails.
- Use VMware Distributed Resource Scheduler (DRS) to migrate virtual machines from one host to another for load balancing. Because storage is on a shared SAN array, applications continue running seamlessly.
- If you use VMware DRS clusters, put an ESXi host into maintenance mode to have the system migrate all running virtual machines to other ESXi hosts. You can then perform upgrades or other maintenance operations on the original host.

The portability and encapsulation of VMware virtual machines complements the shared nature of this storage. When virtual machines are located on SAN-based storage, you can quickly shut down a virtual machine on one server and power it up on another server, or suspend it on one server and resume operation on another server on the same network. This ability allows you to migrate computing resources while maintaining consistent shared access.

This chapter includes the following topics:

- [“ESXi and SAN Use Cases,”](#) on page 28
- [“Specifics of Using SAN Storage with ESXi,”](#) on page 28
- [“ESXi Hosts and Multiple Storage Arrays,”](#) on page 29
- [“Making LUN Decisions,”](#) on page 29
- [“Choosing Virtual Machine Locations,”](#) on page 30
- [“Layered Applications,”](#) on page 31
- [“Third-Party Management Applications,”](#) on page 32

- [“SAN Storage Backup Considerations,”](#) on page 32

ESXi and SAN Use Cases

When used with a SAN, ESXi can benefit from multiple vSphere features, including Storage vMotion, Distributed Resource Scheduler (DRS), High Availability, and so on.

Using ESXi in conjunction with a SAN is effective for the following tasks:

Storage consolidation and simplification of storage layout	If you are working with multiple hosts, and each host is running multiple virtual machines, the storage on the hosts is no longer sufficient and external storage is required. Choose a SAN for external storage to provide a simpler system architecture along with other benefits.
Maintenance with zero downtime	When performing ESXi host or infrastructure maintenance, use vMotion to migrate virtual machines to other host. If shared storage is on the SAN, you can perform maintenance without interruptions to the users of the virtual machines. Virtual machine working processes continue throughout a migration.
Load balancing	You can add a host to a DRS cluster, and the host's resources become part of the cluster's resources. The distribution and usage of CPU and memory resources for all hosts and virtual machines in the cluster are continuously monitored. DRS compares these metrics to an ideal resource utilization. Ideal utilization takes into account the attributes of the cluster's resource pools and virtual machines, the current demand, and the imbalance target. It then performs (or recommends) virtual machine migrations accordingly.
Disaster recovery	You can use VMware High Availability to configure multiple ESXi hosts as a cluster to provide rapid recovery from outages and cost-effective high availability for applications running in virtual machines.
Simplified array migrations and storage upgrades	When you purchase new storage systems or arrays, use Storage vMotion to perform live automated migration of virtual machine disk files from existing storage to their new destination without interruptions to the users of the virtual machines.

Specifics of Using SAN Storage with ESXi

Using a SAN in conjunction with an ESXi host differs from traditional SAN usage in a variety of ways.

When you use SAN storage with ESXi, keep in mind the following considerations:

- You cannot use SAN administration tools to directly access operating systems of virtual machines that use the storage. With traditional tools, you can monitor only the VMware ESXi operating system. You use the vSphere Web Client to monitor virtual machines.
- The HBA visible to the SAN administration tools is part of the ESXi system, not part of the virtual machine.
- Typically, your ESXi system performs multipathing for you.

ESXi Hosts and Multiple Storage Arrays

An ESXi host can access storage devices presented from multiple storage arrays, including arrays from different vendors.

When you use multiple arrays from different vendors, the following considerations apply:

- If your host uses the same Storage Array Type Plugin (SATP) for multiple arrays, be careful when you need to change the default Path Selection Policy (PSP) for that SATP. The change will apply to all arrays. For information on SATPs and PSPs, see [Chapter 17, “Understanding Multipathing and Failover,”](#) on page 181.
- Some storage arrays make recommendations on queue depth and other settings. Typically, these settings are configured globally at the ESXi host level. Making a change for one array impacts other arrays that present LUNs to the host. For information on changing queue depth, see the VMware knowledge base article at <http://kb.vmware.com/kb/1267>.
- Use single-initiator-single-target zoning when zoning ESXi hosts to Fibre Channel arrays. With this type of configuration, fabric related events that occur on one array do not impact other arrays. For more information about zoning, see [“Using Zoning with Fibre Channel SANs,”](#) on page 36.

Making LUN Decisions

You must plan how to set up storage for your ESXi systems before you format LUNs with VMFS datastores.

When you make your LUN decision, keep in mind the following considerations:

- Each LUN should have the correct RAID level and storage characteristic for the applications running in virtual machines that use the LUN.
- Each LUN must contain only one VMFS datastore.
- If multiple virtual machines access the same VMFS, use disk shares to prioritize virtual machines.

You might want fewer, larger LUNs for the following reasons:

- More flexibility to create virtual machines without asking the storage administrator for more space.
- More flexibility for resizing virtual disks, doing snapshots, and so on.
- Fewer VMFS datastores to manage.

You might want more, smaller LUNs for the following reasons:

- Less wasted storage space.
- Different applications might need different RAID characteristics.
- More flexibility, as the multipathing policy and disk shares are set per LUN.
- Use of Microsoft Cluster Service requires that each cluster disk resource is in its own LUN.
- Better performance because there is less contention for a single volume.

When the storage characterization for a virtual machine is not available, there is often no simple method to determine the number and size of LUNs to provision. You can experiment using either a predictive or adaptive scheme.

Use the Predictive Scheme to Make LUN Decisions

When setting up storage for ESXi systems, before creating VMFS datastores, you must decide on the size and number of LUNs to provision. You can experiment using the predictive scheme.

Procedure

- 1 Provision several LUNs with different storage characteristics.
- 2 Create a VMFS datastore on each LUN, labeling each datastore according to its characteristics.
- 3 Create virtual disks to contain the data for virtual machine applications in the VMFS datastores created on LUNs with the appropriate RAID level for the applications' requirements.
- 4 Use disk shares to distinguish high-priority from low-priority virtual machines.

NOTE Disk shares are relevant only within a given host. The shares assigned to virtual machines on one host have no effect on virtual machines on other hosts.

- 5 Run the applications to determine whether virtual machine performance is acceptable.

Use the Adaptive Scheme to Make LUN Decisions

When setting up storage for ESXi hosts, before creating VMFS datastores, you must decide on the number and size of LUNs to provision. You can experiment using the adaptive scheme.

Procedure

- 1 Provision a large LUN (RAID 1+0 or RAID 5), with write caching enabled.
- 2 Create a VMFS on that LUN.
- 3 Create four or five virtual disks on the VMFS.
- 4 Run the applications to determine whether disk performance is acceptable.

If performance is acceptable, you can place additional virtual disks on the VMFS. If performance is not acceptable, create a new, large LUN, possibly with a different RAID level, and repeat the process. Use migration so that you do not lose virtual machines data when you recreate the LUN.

Choosing Virtual Machine Locations

When you're working on optimizing performance for your virtual machines, storage location is an important factor. A trade-off always exists between expensive storage that offers high performance and high availability and storage with lower cost and lower performance.

Storage can be divided into different tiers depending on a number of factors:

- **High Tier.** Offers high performance and high availability. Might offer built-in snapshots to facilitate backups and point-in-time (PiT) restorations. Supports replication, full storage processor redundancy, and SAS drives. Uses high-cost spindles.
- **Mid Tier.** Offers mid-range performance, lower availability, some storage processor redundancy, and SCSI or SAS drives. May offer snapshots. Uses medium-cost spindles.
- **Lower Tier.** Offers low performance, little internal storage redundancy. Uses low end SCSI drives or SATA (serial low-cost spindles).

Not all applications need to be on the highest-performance, most-available storage—at least not throughout their entire life cycle.

NOTE If you need some of the functionality of the high tier, such as snapshots, but do not want to pay for it, you might be able to achieve some of the high-performance characteristics in software. For example, you can create snapshots in software.

When you decide where to place a virtual machine, ask yourself these questions:

- How critical is the virtual machine?
- What are its performance and availability requirements?
- What are its PiT restoration requirements?
- What are its backup requirements?
- What are its replication requirements?

A virtual machine might change tiers throughout its life cycle because of changes in criticality or changes in technology that push higher-tier features to a lower tier. Criticality is relative and might change for a variety of reasons, including changes in the organization, operational processes, regulatory requirements, disaster planning, and so on.

Layered Applications

SAN administrators customarily use specialized array-based software for backup, disaster recovery, data mining, forensics, and configuration testing.

Storage providers typically supply two types of advanced services for their LUNs: snapshotting and replication.

- Snapshotting creates space with efficient copies of LUNs that share common blocks of data. In general, snapshotting is used locally on the same storage systems as the primary LUN for quick backups, application testing, forensics, or data mining.
- Replication creates full copies of LUNs. Replicas are usually made to separate storage systems, possibly separate sites to protect against major outages that incapacitate or destroy an entire array or site.

When you use an ESXi system in conjunction with a SAN, you must decide whether array-based or host-based tools are more suitable for your particular situation.

Array-Based (Third-Party) Solution

When you use an ESXi system in conjunction with a SAN, you must decide whether array-based tools are more suitable for your particular situation.

When you consider an array-based solution, keep in mind the following points:

- Array-based solutions usually result in more comprehensive statistics. With RDMs, data always takes the same path, which results in easier performance management.
- Security is more transparent to the storage administrator when you use an RDM and an array-based solution because with RDMs, virtual machines more closely resemble physical machines.
- If you use an array-based solution, physical compatibility RDMs are often used for the storage of virtual machines. If you do not intend to use RDMs, check the storage vendor documentation to see if operations on LUNs with VMFS volumes are supported. If you use array operations on VMFS LUNs, carefully read the section on resignaturing.

File-Based (VMFS) Solution

When you use an ESXi system in conjunction with a SAN, you must decide whether file-based tools are more suitable for your particular situation.

When you consider a file-based solution that uses VMware tools and VMFS instead of the array tools, be aware of the following points:

- Using VMware tools and VMFS is better for provisioning. One large LUN is allocated and multiple .vmdk files can be placed on that LUN. With an RDM, a new LUN is required for each virtual machine.
- Snapshotting is included with your ESXi host at no extra cost.
- Using VMFS is easier for ESXi administrators.
- ESXi administrators who use the file-based solution are more independent from the SAN administrator.

Third-Party Management Applications

You can use third-party management applications in conjunction with your ESXi host.

Most SAN hardware is packaged with storage management software. In many cases, this software is a web application that can be used with any web browser connected to your network. In other cases, this software typically runs on the storage system or on a single server, independent of the servers that use the SAN for storage.

Use this third-party management software for the following tasks:

- Storage array management, including LUN creation, array cache management, LUN mapping, and LUN security.
- Setting up replication, check points, snapshots, or mirroring.

If you decide to run the SAN management software on a virtual machine, you gain the benefits of running a virtual machine, including failover using vMotion and VMware HA. Because of the additional level of indirection, however, the management software might not be able to see the SAN. In this case, you can use an RDM.

NOTE Whether a virtual machine can run management software successfully depends on the particular storage system.

SAN Storage Backup Considerations

Having a proper backup strategy is one of the most important aspects of SAN management. In the SAN environment, backups have two goals. The first goal is to archive online data to offline media. This process is repeated periodically for all online data on a time schedule. The second goal is to provide access to offline data for recovery from a problem. For example, database recovery often requires retrieval of archived log files that are not currently online.

Scheduling a backup depends on a number of factors:

- Identification of critical applications that require more frequent backup cycles within a given period of time.
- Recovery point and recovery time goals. Consider how precise your recovery point needs to be, and how long you are willing to wait for it.
- The rate of change (RoC) associated with the data. For example, if you are using synchronous/asynchronous replication, the RoC affects the amount of bandwidth required between the primary and secondary storage devices.

- Overall impact on SAN environment, storage performance (while backing up), and other applications.
- Identification of peak traffic periods on the SAN (backups scheduled during those peak periods can slow the applications and the backup process).
- Time to schedule all backups within the data center.
- Time it takes to back up an individual application.
- Resource availability for archiving data; usually offline media access (tape).

Include a recovery-time objective for each application when you design your backup strategy. That is, consider the time and resources necessary to perform a backup. For example, if a scheduled backup stores so much data that recovery requires a considerable amount of time, examine the scheduled backup. Perform the backup more frequently, so that less data is backed up at a time and the recovery time decreases.

If a particular application requires recovery within a certain time frame, the backup process needs to provide a time schedule and specific data processing to meet this requirement. Fast recovery can require the use of recovery volumes that reside on online storage to minimize or eliminate the need to access slow offline media for missing data components.

Using Third-Party Backup Packages

You can use third-party backup solutions to protect system, application, and user data in your virtual machines.

VMware offers the Storage APIs - Data Protection to work in conjunction with third-party products. When using the APIs, third-party software can perform backups without loading ESXi hosts with the processing of backup tasks.

The third-party products using the Storage APIs - Data Protection can perform the following backup tasks:

- Perform full, differential, and incremental image backup and restore of virtual machines.
- Perform file-level backup of virtual machines that use supported Windows and Linux operating systems.
- Ensure data consistency by using Microsoft Volume Shadow Copy Services (VSS) for virtual machines that run supported Microsoft Windows operating systems.

Because the Storage APIs - Data Protection leverage the snapshot capabilities of VMFS, backups that you can perform do not require downtime for virtual machines. These backups are nondisruptive, can be performed at any time, and do not need extended backup windows.

For information about the Storage APIs - Data Protection and integration with backup products, see the VMware Web site or contact your backup vendor.

Using ESXi with Fibre Channel SAN

When you set up ESXi hosts to use FC SAN storage arrays, special considerations are necessary. This section provides introductory information about how to use ESXi with a FC SAN array.

This chapter includes the following topics:

- [“Fibre Channel SAN Concepts,”](#) on page 35
- [“Using Zoning with Fibre Channel SANs,”](#) on page 36
- [“How Virtual Machines Access Data on a Fibre Channel SAN,”](#) on page 37

Fibre Channel SAN Concepts

If you are an ESXi administrator planning to set up hosts to work with SANs, you must have a working knowledge of SAN concepts. You can find information about SANs in print and on the Internet. Because this industry changes constantly, check these resources frequently.

If you are new to SAN technology, familiarize yourself with the basic terminology.

A storage area network (SAN) is a specialized high-speed network that connects computer systems, or host servers, to high performance storage subsystems. The SAN components include host bus adapters (HBAs) in the host servers, switches that help route storage traffic, cables, storage processors (SPs), and storage disk arrays.

A SAN topology with at least one switch present on the network forms a SAN fabric.

To transfer traffic from host servers to shared storage, the SAN uses the Fibre Channel (FC) protocol that packages SCSI commands into Fibre Channel frames.

To restrict server access to storage arrays not allocated to that server, the SAN uses zoning. Typically, zones are created for each group of servers that access a shared group of storage devices and LUNs. Zones define which HBAs can connect to which SPs. Devices outside a zone are not visible to the devices inside the zone.

Zoning is similar to LUN masking, which is commonly used for permission management. LUN masking is a process that makes a LUN available to some hosts and unavailable to other hosts.

When transferring data between the host server and storage, the SAN uses a technique known as multipathing. Multipathing allows you to have more than one physical path from the ESXi host to a LUN on a storage system.

Generally, a single path from a host to a LUN consists of an HBA, switch ports, connecting cables, and the storage controller port. If any component of the path fails, the host selects another available path for I/O. The process of detecting a failed path and switching to another is called path failover.

Ports in Fibre Channel SAN

In the context of this document, a port is the connection from a device into the SAN. Each node in the SAN, such as a host, a storage device, or a fabric component has one or more ports that connect it to the SAN. Ports are identified in a number of ways.

WWPN (World Wide Port Name)	A globally unique identifier for a port that allows certain applications to access the port. The FC switches discover the WWPN of a device or host and assign a port address to the device.
Port_ID (or port address)	Within a SAN, each port has a unique port ID that serves as the FC address for the port. This unique ID enables routing of data through the SAN to that port. The FC switches assign the port ID when the device logs in to the fabric. The port ID is valid only while the device is logged on.

When N-Port ID Virtualization (NPIV) is used, a single FC HBA port (N-port) can register with the fabric by using several WWPNs. This method allows an N-port to claim multiple fabric addresses, each of which appears as a unique entity. When ESXi hosts use a SAN, these multiple, unique identifiers allow the assignment of WWNs to individual virtual machines as part of their configuration.

Fibre Channel Storage Array Types

ESXi supports different storage systems and arrays.

The types of storage that your host supports include active-active, active-passive, and ALUA-compliant.

Active-active storage system	Allows access to the LUNs simultaneously through all the storage ports that are available without significant performance degradation. All the paths are active at all times, unless a path fails.
Active-passive storage system	A system in which one storage processor is actively providing access to a given LUN. The other processors act as backup for the LUN and can be actively providing access to other LUN I/O. I/O can be successfully sent only to an active port for a given LUN. If access through the active storage port fails, one of the passive storage processors can be activated by the servers accessing it.
Asymmetrical storage system	Supports Asymmetric Logical Unit Access (ALUA). ALUA-complaint storage systems provide different levels of access per port. ALUA allows hosts to determine the states of target ports and prioritize paths. The host uses some of the active paths as primary while others as secondary.

Using Zoning with Fibre Channel SANs

Zoning provides access control in the SAN topology. Zoning defines which HBAs can connect to which targets. When you configure a SAN by using zoning, the devices outside a zone are not visible to the devices inside the zone.

Zoning has the following effects:

- Reduces the number of targets and LUNs presented to a host.
- Controls and isolates paths in a fabric.
- Can prevent non-ESXi systems from accessing a particular storage system, and from possibly destroying VMFS data.
- Can be used to separate different environments, for example, a test from a production environment.

With ESXi hosts, use a single-initiator zoning or a single-initiator-single-target zoning. The latter is a preferred zoning practice. Using the more restrictive zoning prevents problems and misconfigurations that can occur on the SAN.

For detailed instructions and best zoning practices, contact storage array or switch vendors.

How Virtual Machines Access Data on a Fibre Channel SAN

ESXi stores a virtual machine's disk files within a VMFS datastore that resides on a SAN storage device. When virtual machine guest operating systems issue SCSI commands to their virtual disks, the SCSI virtualization layer translates these commands to VMFS file operations.

When a virtual machine interacts with its virtual disk stored on a SAN, the following process takes place:

- 1 When the guest operating system in a virtual machine reads or writes to a SCSI disk, it issues SCSI commands to the virtual disk.
- 2 Device drivers in the virtual machine's operating system communicate with the virtual SCSI controllers.
- 3 The virtual SCSI controller forwards the command to the VMkernel.
- 4 The VMkernel performs the following tasks.
 - a Locates the file in the VMFS volume that corresponds to the guest virtual machine disk.
 - b Maps the requests for the blocks on the virtual disk to blocks on the appropriate physical device.
 - c Sends the modified I/O request from the device driver in the VMkernel to the physical HBA.
- 5 The physical HBA performs the following tasks.
 - a Packages the I/O request according to the rules of the FC protocol.
 - b Transmits the request to the SAN.
- 6 Depending on a port the HBA uses to connect to the fabric, one of the SAN switches receives the request and routes it to the storage device that the host wants to access.

Configuring Fibre Channel Storage

When you use ESXi systems with SAN storage, specific hardware and system requirements exist.

This chapter includes the following topics:

- [“ESXi Fibre Channel SAN Requirements,”](#) on page 39
- [“Installation and Setup Steps,”](#) on page 40
- [“N-Port ID Virtualization,”](#) on page 41

ESXi Fibre Channel SAN Requirements

In preparation for configuring your SAN and setting up your ESXi system to use SAN storage, review the requirements and recommendations.

- Make sure that the SAN storage hardware and firmware combinations you use are supported in conjunction with ESXi systems. For an up-to-date list, see the *VMware Compatibility Guide*.
- Configure your system to have only one VMFS volume per LUN.
- Unless you are using diskless servers, do not set up the diagnostic partition on a SAN LUN.
In the case of diskless servers that boot from a SAN, a shared diagnostic partition is appropriate.
- Use RDMs to access raw disks. For information, see [Chapter 18, “Raw Device Mapping,”](#) on page 203.
- For multipathing to work properly, each LUN must present the same LUN ID number to all ESXi hosts.
- Make sure the storage device driver specifies a large enough queue. You can set the queue depth for the physical HBA during system setup. For information on changing queue depth for HBAs and virtual machines, see the *vSphere Troubleshooting* documentation.
- On virtual machines running Microsoft Windows, increase the value of the `SCSI TimeoutValue` parameter to 60. This increase allows Windows to better tolerate delayed I/O resulting from path failover. For information, see [“Set Timeout on Windows Guest OS,”](#) on page 185.

ESXi Fibre Channel SAN Restrictions

When you use ESXi with a SAN, certain restrictions apply.

- ESXi does not support FC connected tape devices.
- You cannot use multipathing software inside a virtual machine to perform I/O load balancing to a single physical LUN. However, when your Microsoft Windows virtual machine uses dynamic disks, this restriction does not apply. For information about configuring dynamic disks, see [“Set Up Dynamic Disk Mirroring,”](#) on page 173.

Setting LUN Allocations

This topic provides general information about how to allocate LUNs when your ESXi works in conjunction with SAN.

When you set LUN allocations, be aware of the following points:

Storage provisioning

To ensure that the ESXi system recognizes the LUNs at startup time, provision all LUNs to the appropriate HBAs before you connect the SAN to the ESXi system.

VMware recommends that you provision all LUNs to all ESXi HBAs at the same time. HBA failover works only if all HBAs see the same LUNs.

For LUNs that will be shared among multiple hosts, make sure that LUN IDs are consistent across all hosts. For example, LUN 5 should be mapped to host 1, host 2, and host 3 as LUN 5.

vMotion and VMware DRS

When you use vCenter Server and vMotion or DRS, make sure that the LUNs for the virtual machines are provisioned to all ESXi hosts. This provides the most ability to move virtual machines.

Active-active compared to active-passive arrays

When you use vMotion or DRS with an active-passive SAN storage device, make sure that all ESXi systems have consistent paths to all storage processors. Not doing so can cause path thrashing when a vMotion migration occurs.

For active-passive storage arrays not listed in Storage/SAN Compatibility, VMware does not support storage port failover. In those cases, you must connect the server to the active port on the storage array. This configuration ensures that the LUNs are presented to the ESXi host.

Setting Fibre Channel HBAs

Typically, FC HBAs that you use on your ESXi host work correctly with the default configuration settings.

You should follow the configuration guidelines provided by your storage array vendor. During FC HBA setup, consider the following issues.

- Do not mix FC HBAs from different vendors in a single host. Having different models of the same HBA is supported, but a single LUN cannot be accessed through two different HBA types, only through the same type.
- Ensure that the firmware level on each HBA is the same.
- Set the timeout value for detecting a failover. To ensure optimal performance, do not change the default value.
- ESXi supports 16 GB end-to-end Fibre Channel connectivity.

Installation and Setup Steps

This topic provides an overview of installation and setup steps that you need to follow when configuring your SAN environment to work with ESXi.

Follow these steps to configure your ESXi SAN environment.

- 1 Design your SAN if it is not already configured. Most existing SANs require only minor modification to work with ESXi.
- 2 Check that all SAN components meet requirements.

- 3 Perform any necessary storage array modification.
Most vendors have vendor-specific documentation for setting up a SAN to work with VMware ESXi.
- 4 Set up the HBAs for the hosts you have connected to the SAN.
- 5 Install ESXi on the hosts.
- 6 Create virtual machines and install guest operating systems.
- 7 (Optional) Set up your system for VMware HA failover or for using Microsoft Clustering Services.
- 8 Upgrade or modify your environment as needed.

N-Port ID Virtualization

N-Port ID Virtualization (NPIV) is an ANSI T11 standard that describes how a single Fibre Channel HBA port can register with the fabric using several worldwide port names (WWPNs). This allows a fabric-attached N-port to claim multiple fabric addresses. Each address appears as a unique entity on the Fibre Channel fabric.

How NPIV-Based LUN Access Works

NPIV enables a single FC HBA port to register several unique WWNs with the fabric, each of which can be assigned to an individual virtual machine.

SAN objects, such as switches, HBAs, storage devices, or virtual machines can be assigned World Wide Name (WWN) identifiers. WWNs uniquely identify such objects in the Fibre Channel fabric. When virtual machines have WWN assignments, they use them for all RDM traffic, so the LUNs pointed to by any of the RDMs on the virtual machine must not be masked against its WWNs. When virtual machines do not have WWN assignments, they access storage LUNs with the WWNs of their host's physical HBAs. By using NPIV, however, a SAN administrator can monitor and route storage access on a per virtual machine basis. The following section describes how this works.

When a virtual machine has a WWN assigned to it, the virtual machine's configuration file (.vmx) is updated to include a WWN pair (consisting of a World Wide Port Name, WWPN, and a World Wide Node Name, WWNN). As that virtual machine is powered on, the VMkernel instantiates a virtual port (VPORT) on the physical HBA which is used to access the LUN. The VPORT is a virtual HBA that appears to the FC fabric as a physical HBA, that is, it has its own unique identifier, the WWN pair that was assigned to the virtual machine. Each VPORT is specific to the virtual machine, and the VPORT is destroyed on the host and it no longer appears to the FC fabric when the virtual machine is powered off. When a virtual machine is migrated from one host to another, the VPORT is closed on the first host and opened on the destination host.

If NPIV is enabled, WWN pairs (WWPN & WWNN) are specified for each virtual machine at creation time. When a virtual machine using NPIV is powered on, it uses each of these WWN pairs in sequence to try to discover an access path to the storage. The number of VPORTs that are instantiated equals the number of physical HBAs present on the host. A VPORT is created on each physical HBA that a physical path is found on. Each physical path is used to determine the virtual path that will be used to access the LUN. Note that HBAs that are not NPIV-aware are skipped in this discovery process because VPORTs cannot be instantiated on them.

Requirements for Using NPIV

If you plan to enable NPIV on your virtual machines, you should be aware of certain requirements.

The following requirements exist:

- NPIV can be used only for virtual machines with RDM disks. Virtual machines with regular virtual disks use the WWNs of the host's physical HBAs.

- HBAs on your host must support NPIV.

For information, see the *VMware Compatibility Guide* and refer to your vendor documentation.

- Use HBAs of the same type, either all QLogic or all Emulex. VMware does not support heterogeneous HBAs on the same host accessing the same LUNs.
- If a host uses multiple physical HBAs as paths to the storage, zone all physical paths to the virtual machine. This is required to support multipathing even though only one path at a time will be active.
- Make sure that physical HBAs on the host have access to all LUNs that are to be accessed by NPIV-enabled virtual machines running on that host.
- The switches in the fabric must be NPIV-aware.
- When configuring a LUN for NPIV access at the storage level, make sure that the NPIV LUN number and NPIV target ID match the physical LUN and Target ID.

NPIV Capabilities and Limitations

Learn about specific capabilities and limitations of the use of NPIV with ESXi.

ESXi with NPIV supports the following items:

- NPIV supports vMotion. When you use vMotion to migrate a virtual machine it retains the assigned WWN.

If you migrate an NPIV-enabled virtual machine to a host that does not support NPIV, VMkernel reverts to using a physical HBA to route the I/O.

- If your FC SAN environment supports concurrent I/O on the disks from an active-active array, the concurrent I/O to two different NPIV ports is also supported.

When you use ESXi with NPIV, the following limitations apply:

- Because the NPIV technology is an extension to the FC protocol, it requires an FC switch and does not work on the direct attached FC disks.
- When you clone a virtual machine or template with a WWN assigned to it, the clones do not retain the WWN.
- NPIV does not support Storage vMotion.
- Disabling and then re-enabling the NPIV capability on an FC switch while virtual machines are running can cause an FC link to fail and I/O to stop.

Assign WWNs to Virtual Machines

Assign WWN settings to virtual machine with an RDM disk.

You can create from 1 to 16 WWN pairs, which can be mapped to the first 1 to 16 physical FC HBAs on the host.

Prerequisites

Create a virtual machine with an RDM disk. See [“Create Virtual Machines with RDMs,”](#) on page 208.

Procedure

- 1 In the vSphere Web Client, browse to the virtual machine.
- 2 Right-click the virtual machine and select **Edit Settings**.
- 3 Click **VM Options**.

- 4 Click the Fibre Channel NPIV triangle to expand the NPIV options.
- 5 Deselect the **Temporarily Disable NPIV for this virtual machine** check box.
- 6 Select **Generate new WWNs**.
- 7 Specify the number of WWNNs and WWPNS.

A minimum of 2 WWPNS are needed to support failover with NPIV. Typically only 1 WWNN is created for each virtual machine.

The host creates WWN assignments for the virtual machine.

What to do next

Register newly created WWNs in the fabric so that the virtual machine is able to log in to the switch, and assign storage LUNs to the WWNs.

Modify WWN Assignments

You can modify WWN assignments for a virtual machine with an RDM.

Typically, you do not need to change existing WWN assignments on your virtual machine. In certain circumstances, for example, when manually assigned WWNs are causing conflicts on the SAN, you might need to change or remove WWNs.

Prerequisites

Make sure to power off the virtual machine if you want to edit the existing WWNs.

Before you begin, ensure that your SAN administrator has provisioned the storage LUN ACL to allow the virtual machine's ESXi host to access it.

Procedure

- 1 Open the Virtual Machine Properties dialog box by clicking the **Edit Settings** link for the selected virtual machine.
- 2 Click the **Options** tab and select **Fibre Channel NPIV**.
The Virtual Machine Properties dialog box opens.
- 3 Edit the WWN assignments by selecting one of the following options:

Option	Description
Temporarily disable NPIV for this virtual machine	Disable the WWN assignments for the virtual machine.
Leave unchanged	The existing WWN assignments are retained. The read-only WWN Assignments section of this dialog box displays the node and port values of any existing WWN assignments.
Generate new WWNs	New WWNs are generated and assigned to the virtual machine, overwriting any existing WWNs (those of the HBA itself are unaffected).
Remove WWN assignment	The WWNs assigned to the virtual machine are removed and it uses the HBA WWNs to access the storage LUN. This option is not available if you are creating a new virtual machine.

- 4 Click **OK** to save your changes.

Configuring Fibre Channel over Ethernet

5

To access Fibre Channel storage, an ESXi host can use the Fibre Channel over Ethernet (FCoE) protocol.

The FCoE protocol encapsulates Fibre Channel frames into Ethernet frames. As a result, your host does not need special Fibre Channel links to connect to Fibre Channel storage, but can use 10Gbit lossless Ethernet to deliver Fibre Channel traffic.

This chapter includes the following topics:

- [“Fibre Channel over Ethernet Adapters,”](#) on page 45
- [“Configuration Guidelines for Software FCoE,”](#) on page 46
- [“Set Up Networking for Software FCoE,”](#) on page 46
- [“Add Software FCoE Adapters,”](#) on page 47

Fibre Channel over Ethernet Adapters

To use Fibre Channel over Ethernet (FCoE), you need to install FCoE adapters on your host.

The adapters that VMware supports generally fall into two categories, hardware FCoE adapters and software FCoE adapters that use the native FCoE stack in ESXi.

Hardware FCoE Adapters

This category includes completely offloaded specialized Converged Network Adapters (CNAs) that contain network and Fibre Channel functionalities on the same card.

When such adapter is installed, your host detects and can use both CNA components. In the client, the networking component appears as a standard network adapter (vmnic) and the Fibre Channel component as a FCoE adapter (vmhba). You do not need to configure the hardware FCoE adapter to be able to use it.

Software FCoE Adapters

A software FCoE adapter uses the native FCoE protocol stack in ESXi for the protocol processing. The software FCoE adapter is used with a NIC that offers Data Center Bridging (DCB) and I/O offload capabilities. Intel X520 is an example of such NIC. For information on NICs supporting software FCoE, see the *VMware Compatibility Guide*.

For the software FCoE adapter, you must properly configure networking and then activate the adapter.

NOTE The number of software FCoE adapters you activate corresponds to the number of physical NIC ports. ESXi supports a maximum of four software FCoE adapters on one host.

Configuration Guidelines for Software FCoE

When setting up your network environment to work with ESXi software FCoE, follow the guidelines and best practices that VMware offers.

Network Switch Guidelines

Follow these guidelines when you configure a network switch for software FCoE environment:

- On the ports that communicate with your ESXi host, disable the Spanning Tree Protocol (STP). Having the STP enabled might delay the FCoE Initialization Protocol (FIP) response at the switch and cause an all paths down (APD) condition.

The FIP is a protocol that FCoE uses to discover and initialize FCoE entities on the Ethernet.

- Turn on Priority-based Flow Control (PFC) and set it to AUTO.
- Make sure that you have a compatible firmware version on the FCoE switch.

Network Adapter Best Practices

If you plan to enable software FCoE adapters to work with network adapters, specific considerations apply.

- Make sure that the latest microcode is installed on the FCoE network adapter.
- If the network adapter has multiple ports, when configuring networking, add each port to a separate vSwitch. This practice helps you to avoid an APD condition when a disruptive event, such as an MTU change, occurs.
- Do not move a network adapter port from one vSwitch to another when FCoE traffic is active. If you need to make this change, reboot your host afterwards.
- If you changed the vSwitch for a network adapter port and caused a failure, moving the port back to the original vSwitch resolves the problem.

Set Up Networking for Software FCoE

Before you activate the software FCoE adapters, you need to create VMkernel network adapters for all physical FCoE NICs installed on your host.

This procedure explains how to create a single VMkernel network adapter connected to a single FCoE physical network adapter through a vSphere standard switch. If your host has multiple network adapters or multiple ports on the adapter, connect each FCoE NIC to a separate standard switch. For more information, see the *vSphere Networking* documentation.

NOTE ESXi supports the maximum of four network adapter ports used for software FCoE.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click **Actions > Add Networking**.
- 3 Select **VMkernel Network Adapter**, and click **Next**.
- 4 Select **New standard switch** to create a vSphere standard switch.
- 5 Under Unclaimed Adapters, select the network adapter (vmnic#) that supports FCoE and click **Assign**.
Make sure to assign the adapter to Active Adapters.

- 6 Enter a network label.

Network label is a friendly name that identifies the VMkernel adapter that you are creating, for example, FCoE.

- 7 Specify a VLAN ID and click **Next**.

Because FCoE traffic requires an isolated network, make sure that the VLAN ID you enter is different from the one used for regular networking on your host. For more information, see the *vSphere Networking* documentation.

- 8 After completing configuration, review the information and click **Finish**.

You have created the virtual VMkernel adapter for the physical FCoE network adapter installed on your host.

NOTE To avoid FCoE traffic disruptions, do not remove the FCoE network adapter (vmnic#) from the vSphere standard switch after you set up FCoE networking.

Add Software FCoE Adapters

You must activate software FCoE adapters so that your host can use them to access Fibre Channel storage.

The number of software FCoE adapters you can activate corresponds to the number of physical FCoE NIC ports on your host. ESXi supports the maximum of four software FCoE adapters on one host.

Prerequisites

Set up networking for the software FCoE adapter.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters**, and click **Add**.
- 4 Select **Software FCoE Adapter**.
- 5 On the Add Software FCoE Adapter dialog box, select an appropriate vmnic from the drop-down list of physical network adapters.

Only those adapters that are not yet used for FCoE traffic are listed.

- 6 Click **OK**.

The software FCoE adapter appears on the list of storage adapters.

After you activate the software FCoE adapter, you can view its properties. If you do not use the adapter, you can remove it from the list of adapters.

Booting ESXi from Fibre Channel SAN

6

When you set up your host to boot from a SAN, your host's boot image is stored on one or more LUNs in the SAN storage system. When the host starts, it boots from the LUN on the SAN rather than from its local disk.

ESXi supports booting through a Fibre Channel host bus adapter (HBA) or a Fibre Channel over Ethernet (FCoE) converged network adapter (CNA).

This chapter includes the following topics:

- [“Boot from SAN Benefits,”](#) on page 49
- [“Boot from Fibre Channel SAN Requirements and Considerations,”](#) on page 50
- [“Getting Ready for Boot from SAN,”](#) on page 50
- [“Configure Emulex HBA to Boot from SAN,”](#) on page 52
- [“Configure QLogic HBA to Boot from SAN,”](#) on page 53

Boot from SAN Benefits

Boot from SAN can provide numerous benefits to your environment. However, in certain cases, you should not use boot from SAN for ESXi hosts. Before you set up your system for boot from SAN, decide whether it is appropriate for your environment.



CAUTION When you use boot from SAN with multiple ESXi hosts, each host must have its own boot LUN. If you configure multiple hosts to share the same boot LUN, ESXi image corruption is likely to occur.

If you use boot from SAN, the benefits for your environment will include the following:

- Cheaper servers. Servers can be more dense and run cooler without internal storage.
- Easier server replacement. You can replace servers and have the new server point to the old boot location.
- Less wasted space. Servers without local disks often take up less space.
- Easier backup processes. You can backup the system boot images in the SAN as part of the overall SAN backup procedures. Also, you can use advanced array features such as snapshots on the boot image.
- Improved management. Creating and managing the operating system image is easier and more efficient.
- Better reliability. You can access the boot disk through multiple paths, which protects the disk from being a single point of failure.

Boot from Fibre Channel SAN Requirements and Considerations

Your ESXi boot configuration must meet specific requirements.

Table 6-1. Boot from SAN Requirements

Requirement	Description
ESXi system requirements	Follow vendor recommendation for the server booting from a SAN.
Adapter requirements	Enable and correctly configure the adapter, so it can access the boot LUN. See your vendor documentation.
Access control	<ul style="list-style-type: none"> ■ Each host must have access to its own boot LUN only, not the boot LUNs of other hosts. Use storage system software to make sure that the host accesses only the designated LUNs. ■ Multiple servers can share a diagnostic partition. You can use array specific LUN masking to achieve this.
Multipathing support	Multipathing to a boot LUN on active-passive arrays is not supported because the BIOS does not support multipathing and is unable to activate a standby path.
SAN considerations	SAN connections must be through a switched topology if the array is not certified for direct connect topology. If the array is certified for direct connect topology, the SAN connections can be made directly to the array. Boot from SAN is supported for both switched topology and direct connect topology if these topologies for the specific array are certified.
Hardware- specific considerations	If you are running an IBM eServer BladeCenter and use boot from SAN, you must disable IDE drives on the blades.

Getting Ready for Boot from SAN

When you set up your boot from SAN environment, you perform a number of tasks.

This section describes the generic boot-from-SAN enablement process on the rack mounted servers. For information on enabling boot from SAN on Cisco Unified Computing System FCoE blade servers, refer to Cisco documentation.

- 1 [Configure SAN Components and Storage System](#) on page 50
Before you set up your ESXi host to boot from a SAN LUN, configure SAN components and a storage system.
- 2 [Configure Storage Adapter to Boot from SAN](#) on page 51
When you set up your host to boot from SAN, you enable the boot adapter in the host BIOS. You then configure the boot adapter to initiate a primitive connection to the target boot LUN.
- 3 [Set Up Your System to Boot from Installation Media](#) on page 51
When setting up your host to boot from SAN, you first boot the host from the VMware installation media. To achieve this, you need to change the system boot sequence in the BIOS setup.

Configure SAN Components and Storage System

Before you set up your ESXi host to boot from a SAN LUN, configure SAN components and a storage system.

Because configuring the SAN components is vendor specific, refer to the product documentation for each item.

Procedure

- 1 Connect network cable, referring to any cabling guide that applies to your setup.
Check the switch wiring, if there is any.

- 2 Configure the storage array.
 - a From the SAN storage array, make the ESXi host visible to the SAN. This process is often referred to as creating an object.
 - b From the SAN storage array, set up the host to have the WWPNs of the host's adapters as port names or node names.
 - c Create LUNs.
 - d Assign LUNs.
 - e Record the IP addresses of the switches and storage arrays.
 - f Record the WWPN for each SP.



CAUTION If you use scripted installation to install ESXi in boot from SAN mode, you need to take special steps to avoid unintended data loss.

Configure Storage Adapter to Boot from SAN

When you set up your host to boot from SAN, you enable the boot adapter in the host BIOS. You then configure the boot adapter to initiate a primitive connection to the target boot LUN.

Prerequisites

Determine the WWPN for the storage adapter.

Procedure

- ◆ Configure the storage adapter to boot from SAN.

Because configuring boot adapters is vendor specific, refer to your vendor documentation.

Set Up Your System to Boot from Installation Media

When setting up your host to boot from SAN, you first boot the host from the VMware installation media. To achieve this, you need to change the system boot sequence in the BIOS setup.

Because changing the boot sequence in the BIOS is vendor specific, refer to vendor documentation for instructions. The following procedure explains how to change the boot sequence on an IBM host.

Procedure

- 1 During your system power up, enter the system BIOS Configuration/Setup Utility.
- 2 Select **Startup Options** and press Enter.
- 3 Select **Startup Sequence Options** and press Enter.
- 4 Change the **First Startup Device** to [CD-ROM].

You can now install ESXi.

Configure Emulex HBA to Boot from SAN

Configuring the Emulex HBA BIOS to boot from SAN includes enabling the BootBIOS prompt and enabling BIOS.

Procedure

- 1 [Enable the BootBIOS Prompt](#) on page 52
When you configure the Emulex HBA BIOS to boot ESXi from SAN, you need to enable the BootBIOS prompt.
- 2 [Enable the BIOS](#) on page 52
When you configure the Emulex HBA BIOS to boot ESXi from SAN, you need to enable BIOS.

Enable the BootBIOS Prompt

When you configure the Emulex HBA BIOS to boot ESXi from SAN, you need to enable the BootBIOS prompt.

Procedure

- 1 Run `lputil`.
- 2 Select **3. Firmware Maintenance**.
- 3 Select an adapter.
- 4 Select **6. Boot BIOS Maintenance**.
- 5 Select **1. Enable Boot BIOS**.

Enable the BIOS

When you configure the Emulex HBA BIOS to boot ESXi from SAN, you need to enable BIOS.

Procedure

- 1 Reboot the host.
- 2 To configure the adapter parameters, press ALT+E at the Emulex prompt and follow these steps.
 - a Select an adapter (with BIOS support).
 - b Select **2. Configure This Adapter's Parameters**.
 - c Select **1. Enable or Disable BIOS**.
 - d Select **1** to enable BIOS.
 - e Select **x** to exit and **Esc** to return to the previous menu.
- 3 To configure the boot device, follow these steps from the Emulex main menu.
 - a Select the same adapter.
 - b Select **1. Configure Boot Devices**.
 - c Select the location for the Boot Entry.
 - d Enter the two-digit boot device.
 - e Enter the two-digit (HEX) starting LUN (for example, **08**).
 - f Select the boot LUN.

- g Select **1. WWPN**. (Boot this device using WWPN, not DID).
 - h Select **x** to exit and **Y** to reboot.
- 4 Boot into the system BIOS and move Emulex first in the boot controller sequence.
 - 5 Reboot and install on a SAN LUN.

Configure QLogic HBA to Boot from SAN

This sample procedure explains how to configure the QLogic HBA to boot ESXi from SAN. The procedure involves enabling the QLogic HBA BIOS, enabling the selectable boot, and selecting the boot LUN.

Procedure

- 1 While booting the server, press **Ctrl+Q** to enter the Fast!UTIL configuration utility.
- 2 Perform the appropriate action depending on the number of HBAs.

Option	Description
One HBA	If you have only one host bus adapter (HBA), the Fast!UTIL Options page appears. Skip to Step 3 .
Multiple HBAs	If you have more than one HBA, select the HBA manually. <ol style="list-style-type: none"> a In the Select Host Adapter page, use the arrow keys to position the cursor on the appropriate HBA. b Press Enter.

- 3 In the Fast!UTIL Options page, select **Configuration Settings** and press **Enter**.
- 4 In the Configuration Settings page, select **Adapter Settings** and press **Enter**.
- 5 Set the BIOS to search for SCSI devices.
 - a In the Host Adapter Settings page, select **Host Adapter BIOS**.
 - b Press **Enter** to toggle the value to Enabled.
 - c Press **Esc** to exit.
- 6 Enable the selectable boot.
 - a Select **Selectable Boot Settings** and press **Enter**.
 - b In the Selectable Boot Settings page, select **Selectable Boot**.
 - c Press **Enter** to toggle the value to **Enabled**.
- 7 Use the cursor keys to select the Boot Port Name entry in the list of storage processors (SPs) and press **Enter** to open the Select Fibre Channel Device screen.
- 8 Use the cursor keys to select the specific SP and press **Enter**.

If you are using an active-passive storage array, the selected SP must be on the preferred (active) path to the boot LUN. If you are not sure which SP is on the active path, use your storage array management software to find out. The target IDs are created by the BIOS and might change with each reboot.

- 9 Perform the appropriate action depending on the number of LUNs attached to the SP.

Option	Description
One LUN	The LUN is selected as the boot LUN. You do not need to enter the Select LUN screen.
Multiple LUNs	Select LUN screen opens. Use the cursor to select the boot LUN, then press Enter .

- 10 If any remaining storage processors show in the list, press **C** to clear the data.
- 11 Press **Esc** twice to exit and press **Enter** to save the setting.

Booting ESXi with Software FCoE

ESXi supports boot from FCoE capable network adapters.

When you install and boot ESXi from an FCoE LUN, the host can use a VMware software FCoE adapter and a network adapter with FCoE capabilities. The host does not require a dedicated FCoE HBA.

You perform most configurations through the option ROM of your network adapter. The network adapters must support one of the following formats, which communicate parameters about an FCoE boot device to VMkernel.

- FCoE Boot Firmware Table (FBFT). FBFT is Intel proprietary.
- FCoE Boot Parameter Table (FBPT). FBPT is defined by VMware for third-party vendors to implement software FCoE boot.

The configuration parameters are set in the option ROM of your adapter. During an ESXi installation or a subsequent boot, these parameters are exported in to system memory in either FBFT format or FBPT format. The VMkernel can read the configuration settings and use them to access the boot LUN.

This chapter includes the following topics:

- [“Requirements and Considerations for Software FCoE Boot,”](#) on page 55
- [“Best Practices for Software FCoE Boot,”](#) on page 56
- [“Set Up Software FCoE Boot,”](#) on page 56
- [“Troubleshooting Installation and Boot from Software FCoE,”](#) on page 57

Requirements and Considerations for Software FCoE Boot

When you boot the ESXi host from SAN using software FCoE, certain requirements and considerations apply.

Requirements

- ESXi 5.1 or later.
- The network adapter must have the following capabilities:
 - Be FCoE capable.
 - Support ESXi open FCoE stack.
 - Contain FCoE boot firmware which can export boot information in FBFT format or FBPT format.

Considerations

- You cannot change software FCoE boot configuration from within ESXi.

- Coredump is not supported on any software FCoE LUNs, including the boot LUN.
- Multipathing is not supported at pre-boot.
- Boot LUN cannot be shared with other hosts even on shared storage.

Best Practices for Software FCoE Boot

VMware recommends several best practices when you boot your system from a software FCoE LUN.

- Make sure that the host has access to the entire boot LUN. The boot LUN cannot be shared with other hosts even on shared storage.
- If you use Intel 10 Gigabit Ethernet Controller (Niantec) with a Cisco switch, configure the switch port in the following way:
 - Enable the Spanning Tree Protocol (STP).
 - Turn off `switchport trunk native vlan` for the VLAN used for FCoE.

Set Up Software FCoE Boot

Your ESXi host can boot from a FCoE LUN using the software FCoE adapter a network adapter.

When you configure your host for a software FCoE boot, you perform a number of tasks.

Prerequisites

The network adapter has the following capabilities:

- Support partial FCoE offload (software FCoE).
- Contain either a FCoE Boot Firmware Table (FBFT) or a FCoE Boot Parameter Table (FBPT).

For information about network adapters that support software FCoE boot, see the *VMware Compatibility Guide*.

Procedure

- 1 [Configure Software FCoE Boot Parameters](#) on page 56

To support a software FCoE boot process, a network adapter on your host must have a specially configured FCoE boot firmware. When you configure the firmware, you enable the adapter for the software FCoE boot and specify the boot LUN parameters.

- 2 [Install and Boot ESXi from Software FCoE LUN](#) on page 57

When you set up your system to boot from a software FCoE LUN, you install the ESXi image to the target LUN. You can then boot your host from that LUN.

Configure Software FCoE Boot Parameters

To support a software FCoE boot process, a network adapter on your host must have a specially configured FCoE boot firmware. When you configure the firmware, you enable the adapter for the software FCoE boot and specify the boot LUN parameters.

Procedure

- ◆ In the option ROM of the network adapter, specify software FCoE boot parameters.

These parameters include boot target, boot LUN, VLAN ID, and so on.

Because configuring the network adapter is vendor specific, review your vendor documentation for instructions.

Install and Boot ESXi from Software FCoE LUN

When you set up your system to boot from a software FCoE LUN, you install the ESXi image to the target LUN. You can then boot your host from that LUN.

Prerequisites

- Configure the option ROM of the network adapter to point to a target LUN that you want to use as the boot LUN. Make sure that you have information about the bootable LUN.
- Change the boot order in the system BIOS to the following sequence:
 - a The network adapter that you use for the software FCoE boot.
 - b The ESXi installation media.

See the vendor documentation for your system.

Procedure

- 1 Start an interactive installation from the ESXi installation CD/DVD.

The ESXi installer verifies that FCoE boot is enabled in the BIOS and, if needed, creates a standard virtual switch for the FCoE capable network adapter. The name of the vSwitch is `VMware_FCoE_vSwitch`. The installer then uses preconfigured FCoE boot parameters to discover and display all available FCoE LUNs.

- 2 On the Select a Disk screen, select the software FCoE LUN that you specified in the boot parameter setting.

If the boot LUN does not appear in this menu, make sure that you correctly configured boot parameters in the option ROM of the network adapter.

- 3 Follow the prompts to complete the installation.
- 4 Reboot the host.
- 5 Change the boot order in the system BIOS so that the FCoE boot LUN is the first bootable device.

ESXi continues booting from the software FCoE LUN until it is ready to be used.

What to do next

If needed, you can rename and modify the `VMware_FCoE_vSwitch` that the installer automatically created. Make sure that the Cisco Discovery Protocol (CDP) mode is set to Listen or Both.

Troubleshooting Installation and Boot from Software FCoE

If the installation or boot of ESXi from a software FCoE LUN fails, you can use several troubleshooting methods.

Problem

When you install or boot ESXi from FCoE storage using a VMware software FCoE adapter and a network adapter with partial FCoE offload capabilities, the installation or the boot process fails.

Solution

- Make sure that you correctly configured boot parameters in the option ROM of the FCoE network adapter.
- During installation, monitor the BIOS of the FCoE network adapter for any errors.
- If possible, check the VMkernel log for errors.

- Use the `esxcli` command to verify whether the boot LUN is present.

```
esxcli conn_options hardware bootdevice list
```

Best Practices for Fibre Channel Storage

8

When using ESXi with Fibre Channel SAN, follow best practices that VMware offers to avoid performance problems.

The vSphere Web Client offers extensive facilities for collecting performance information. The information is graphically displayed and frequently updated.

You can also use the `resxtop` or `esxtop` command-line utilities. The utilities provide a detailed look at how ESXi uses resources in real time. For more information, see the *vSphere Resource Management* documentation.

Check with your storage representative if your storage system supports Storage API - Array Integration hardware acceleration features. If it does, refer to your vendor documentation for information on how to enable hardware acceleration support on the storage system side. For more information, see [Chapter 22, “Storage Hardware Acceleration,”](#) on page 243.

This chapter includes the following topics:

- [“Preventing Fibre Channel SAN Problems,”](#) on page 59
- [“Disable Automatic Host Registration,”](#) on page 60
- [“Optimizing Fibre Channel SAN Storage Performance,”](#) on page 60

Preventing Fibre Channel SAN Problems

When using ESXi in conjunction with a Fibre Channel SAN, you must follow specific guidelines to avoid SAN problems.

You should observe these tips for preventing problems with your SAN configuration:

- Place only one VMFS datastore on each LUN.
- Do not change the path policy the system sets for you unless you understand the implications of making such a change.
- Document everything. Include information about zoning, access control, storage, switch, server and FC HBA configuration, software and firmware versions, and storage cable plan.
- Plan for failure:
 - Make several copies of your topology maps. For each element, consider what happens to your SAN if the element fails.
 - Cross off different links, switches, HBAs and other elements to ensure you did not miss a critical failure point in your design.
- Ensure that the Fibre Channel HBAs are installed in the correct slots in the host, based on slot and bus speed. Balance PCI bus load among the available busses in the server.

- Become familiar with the various monitor points in your storage network, at all visibility points, including host's performance charts, FC switch statistics, and storage performance statistics.
- Be cautious when changing IDs of the LUNs that have VMFS datastores being used by your ESXi host. If you change the ID, the datastore becomes inactive and its virtual machines fail. You can resignature the datastore to make it active again. See [“Managing Duplicate VMFS Datastores,”](#) on page 162.

If there are no running virtual machines on the VMFS datastore, after you change the ID of the LUN, you must use rescan to reset the ID on your host. For information on using rescan, see [“Storage Refresh and Rescan Operations,”](#) on page 124.

Disable Automatic Host Registration

Certain storage arrays require that ESXi hosts register with the arrays. ESXi performs automatic host registration by sending the host's name and IP address to the array. If you prefer to perform manual registration using storage management software, disable the ESXi auto-registration feature.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Settings**.
- 3 Under System, click **Advanced System Settings**.
- 4 Under Advanced System Settings, select the **Disk.EnableNaviReg** parameter and click the **Edit** icon.
- 5 Change the value to 0.

This disables the automatic host registration enabled by default.

Optimizing Fibre Channel SAN Storage Performance

Several factors contribute to optimizing a typical SAN environment.

If the environment is properly configured, the SAN fabric components (particularly the SAN switches) are only minor contributors because of their low latencies relative to servers and storage arrays. Make sure that the paths through the switch fabric are not saturated, that is, that the switch fabric is running at the highest throughput.

Storage Array Performance

Storage array performance is one of the major factors contributing to the performance of the entire SAN environment.

If there are issues with storage array performance, be sure to consult your storage array vendor's documentation for any relevant information.

Follow these general guidelines to improve the array performance in the vSphere environment:

- When assigning LUNs, remember that each LUN is accessed by a number of hosts, and that a number of virtual machines can run on each host. One LUN used by a host can service I/O from many different applications running on different operating systems. Because of this diverse workload, the RAID group containing the ESXi LUNs should not include LUNs used by other servers that are not running ESXi.
- Make sure read/write caching is enabled.
- SAN storage arrays require continual redesign and tuning to ensure that I/O is load balanced across all storage array paths. To meet this requirement, distribute the paths to the LUNs among all the SPs to provide optimal load balancing. Close monitoring indicates when it is necessary to rebalance the LUN distribution.

Tuning statically balanced storage arrays is a matter of monitoring the specific performance statistics (such as I/O operations per second, blocks per second, and response time) and distributing the LUN workload to spread the workload across all the SPs.

NOTE Dynamic load balancing is not currently supported with ESXi.

Server Performance with Fibre Channel

You must consider several factors to ensure optimal server performance.

Each server application must have access to its designated storage with the following conditions:

- High I/O rate (number of I/O operations per second)
- High throughput (megabytes per second)
- Minimal latency (response times)

Because each application has different requirements, you can meet these goals by choosing an appropriate RAID group on the storage array. To achieve performance goals:

- Place each LUN on a RAID group that provides the necessary performance levels. Pay attention to the activities and resource utilization of other LUNS in the assigned RAID group. A high-performance RAID group that has too many applications doing I/O to it might not meet performance goals required by an application running on the ESXi host.
- Make sure that each server has a sufficient number of HBAs to allow maximum throughput for all the applications hosted on the server for the peak period. I/O spread across multiple HBAs provide higher throughput and less latency for each application.
- To provide redundancy in the event of HBA failure, make sure the server is connected to a dual redundant fabric.
- When allocating LUNs or RAID groups for ESXi systems, multiple operating systems use and share that resource. As a result, the performance required from each LUN in the storage subsystem can be much higher if you are working with ESXi systems than if you are using physical machines. For example, if you expect to run four I/O intensive applications, allocate four times the performance capacity for the ESXi LUNs.
- When using multiple ESXi systems in conjunction with vCenter Server, the performance needed from the storage subsystem increases correspondingly.
- The number of outstanding I/Os needed by applications running on an ESXi system should match the number of I/Os the HBA and storage array can handle.

Using ESXi with iSCSI SAN

You can use ESXi in conjunction with a storage area network (SAN), a specialized high-speed network that connects computer systems to high-performance storage subsystems. Using ESXi together with a SAN provides storage consolidation, improves reliability, and helps with disaster recovery.

To use ESXi effectively with a SAN, you must have a working knowledge of ESXi systems and SAN concepts. Also, when you set up ESXi hosts to use Internet SCSI (iSCSI) SAN storage systems, you must be aware of certain special considerations that exist.

This chapter includes the following topics:

- [“iSCSI SAN Concepts,”](#) on page 63
- [“How Virtual Machines Access Data on an iSCSI SAN,”](#) on page 68

iSCSI SAN Concepts

If you are an administrator who plans to set up ESXi hosts to work with iSCSI SANs, you must have a working knowledge of iSCSI concepts.

iSCSI SANs use Ethernet connections between computer systems, or host servers, and high performance storage subsystems. The SAN components include iSCSI host bus adapters (HBAs) or Network Interface Cards (NICs) in the host servers, switches and routers that transport the storage traffic, cables, storage processors (SPs), and storage disk systems.

iSCSI SAN uses a client-server architecture. The client, called iSCSI initiator, operates on your host. It initiates iSCSI sessions by issuing SCSI commands and transmitting them, encapsulated into iSCSI protocol, to a server. The server is known as an iSCSI target. The iSCSI target represents a physical storage system on the network. It can also be provided by a virtual iSCSI SAN, for example, an iSCSI target emulator running in a virtual machine. The iSCSI target responds to the initiator's commands by transmitting required iSCSI data.

iSCSI Multipathing

When transferring data between the host server and storage, the SAN uses a technique known as multipathing. Multipathing allows you to have more than one physical path from the ESXi host to a LUN on a storage system.

Generally, a single path from a host to a LUN consists of an iSCSI adapter or NIC, switch ports, connecting cables, and the storage controller port. If any component of the path fails, the host selects another available path for I/O. The process of detecting a failed path and switching to another is called path failover.

For more information on multipathing, see [Chapter 17, “Understanding Multipathing and Failover,”](#) on page 181.

Ports in the iSCSI SAN

A single discoverable entity on the iSCSI SAN, such as an initiator or a target, represents an iSCSI node. Each node has one or more ports that connect it to the SAN.

iSCSI ports are end-points of an iSCSI session. Each node can be identified in a number of ways.

IP Address	Each iSCSI node can have an IP address associated with it so that routing and switching equipment on your network can establish the connection between the server and storage. This address is just like the IP address that you assign to your computer to get access to your company's network or the Internet.
iSCSI Name	<p>A worldwide unique name for identifying the node. iSCSI uses the iSCSI Qualified Name (IQN) and Extended Unique Identifier (EUI).</p> <p>By default, ESXi generates unique iSCSI names for your iSCSI initiators, for example, <code>iqn.1998-01.com.vmware:iscsitestox-68158ef2</code>. Usually, you do not have to change the default value, but if you do, make sure that the new iSCSI name you enter is worldwide unique.</p>
iSCSI Alias	A more manageable name for an iSCSI device or port used instead of the iSCSI name. iSCSI aliases are not unique and are intended to be just a friendly name to associate with a port.

iSCSI Naming Conventions

iSCSI uses a special unique name to identify an iSCSI node, either target or initiator. This name is similar to the WorldWide Name (WWN) associated with Fibre Channel devices and is used as a way to universally identify the node.

iSCSI names are formatted in two different ways. The most common is the IQN format.

For more details on iSCSI naming requirements and string profiles, see RFC 3721 and RFC 3722 on the IETF Web site.

iSCSI Qualified Name (IQN) Format

The IQN format takes the form `iqn.yyyy-mm.naming-authority:unique name`, where:

- *yyyy-mm* is the year and month when the naming authority was established.
- *naming-authority* is usually reverse syntax of the Internet domain name of the naming authority. For example, the `iscsi.vmware.com` naming authority could have the iSCSI qualified name form of `iqn.1998-01.com.vmware.iscsi`. The name indicates that the `vmware.com` domain name was registered in January of 1998, and `iscsi` is a subdomain, maintained by `vmware.com`.
- *unique name* is any name you want to use, for example, the name of your host. The naming authority must make sure that any names assigned following the colon are unique, such as:
 - `iqn.1998-01.com.vmware.iscsi:name1`
 - `iqn.1998-01.com.vmware.iscsi:name2`
 - `iqn.1998-01.com.vmware.iscsi:name999`

Enterprise Unique Identifier (EUI) Format

The EUI format takes the form `eui.16 hex digits`.

For example, `eui.0123456789ABCDEF`.

The 16-hexadecimal digits are text representations of a 64-bit number of an IEEE EUI (extended unique identifier) format. The top 24 bits are a company ID that IEEE registers with a particular company. The lower 40 bits are assigned by the entity holding that company ID and must be unique.

iSCSI Initiators

To access iSCSI targets, your host uses iSCSI initiators. The initiators transport SCSI requests and responses, encapsulated into the iSCSI protocol, between the host and the iSCSI target.

Your host supports different types of initiators.

For information on configuring and using iSCSI adapters, see [Chapter 10, “Configuring iSCSI Adapters and Storage,”](#) on page 69.

Software iSCSI Adapter

A software iSCSI adapter is a VMware code built into the VMkernel. It allows your host to connect to the iSCSI storage device through standard network adapters. The software iSCSI adapter handles iSCSI processing while communicating with the network adapter. With the software iSCSI adapter, you can use iSCSI technology without purchasing specialized hardware.

Hardware iSCSI Adapter

A hardware iSCSI adapter is a third-party adapter that offloads iSCSI and network processing from your host. Hardware iSCSI adapters are divided into categories.

Dependent Hardware iSCSI Adapter

Depends on VMware networking, and iSCSI configuration and management interfaces provided by VMware.

This type of adapter can be a card that presents a standard network adapter and iSCSI offload functionality for the same port. The iSCSI offload functionality depends on the host's network configuration to obtain the IP, MAC, and other parameters used for iSCSI sessions. An example of a dependent adapter is the iSCSI licensed Broadcom 5709 NIC.

Independent Hardware iSCSI Adapter

Implements its own networking and iSCSI configuration and management interfaces.

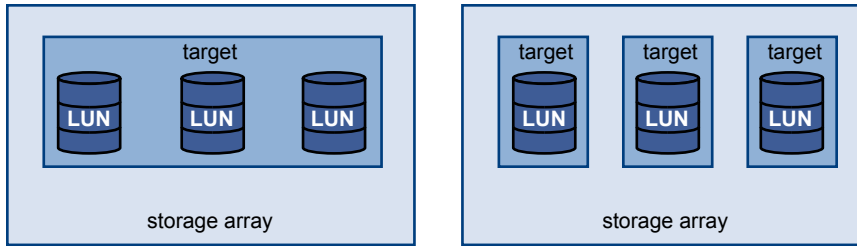
An example of an independent hardware iSCSI adapter is a card that either presents only iSCSI offload functionality or iSCSI offload functionality and standard NIC functionality. The iSCSI offload functionality has independent configuration management that assigns the IP, MAC, and other parameters used for the iSCSI sessions. An example of an independent adapter is the QLogic QLA4052 adapter.

Hardware iSCSI adapters might need to be licensed. Otherwise, they will not appear in the client or vSphere CLI. Contact your vendor for licensing information.

Establishing iSCSI Connections

In the ESXi context, the term target identifies a single storage unit that your host can access. The terms storage device and LUN describe a logical volume that represents storage space on a target. Typically, the terms device and LUN, in the ESXi context, mean a SCSI volume presented to your host from a storage target and available for formatting.

Different iSCSI storage vendors present storage to servers in different ways. Some vendors present multiple LUNs on a single target, while others present multiple targets with one LUN each. While the way the storage is used by ESXi is similar, the way the information is presented through administrative tools is different.

Figure 9-1. Target Compared to LUN Representations

Three LUNs are available in each of these configurations. In the first case, the host detects one target but that target has three LUNs that can be used. Each of the LUNs represents individual storage volume. In the second case, the host detects three different targets, each having one LUN.

Host-based iSCSI initiators establish connections to each target. Storage systems with a single target containing multiple LUNs have traffic to all the LUNs on a single connection. With a system that has three targets with one LUN each, a host uses separate connections to the three LUNs. This information is useful when you are trying to aggregate storage traffic on multiple connections from the host with multiple iSCSI HBAs, where traffic for one target can be set to a particular HBA, while traffic for another target can use a different HBA.

iSCSI Storage System Types

ESXi supports different storage systems and arrays.

The types of storage that your host supports include active-active, active-passive, and ALUA-compliant.

Active-active storage system

Allows access to the LUNs simultaneously through all the storage ports that are available without significant performance degradation. All the paths are active at all times, unless a path fails.

Active-passive storage system

A system in which one storage processor is actively providing access to a given LUN. The other processors act as backup for the LUN and can be actively providing access to other LUN I/O. I/O can be successfully sent only to an active port for a given LUN. If access through the active storage port fails, one of the passive storage processors can be activated by the servers accessing it.

Asymmetrical storage system

Supports Asymmetric Logical Unit Access (ALUA). ALUA-compliant storage systems provide different levels of access per port. ALUA allows hosts to determine the states of target ports and prioritize paths. The host uses some of the active paths as primary while others as secondary.

Virtual port storage system

Allows access to all available LUNs through a single virtual port. These are active-active storage devices, but hide their multiple connections through a single port. ESXi multipathing does not make multiple connections from a specific port to the storage by default. Some storage vendors supply session managers to establish and manage multiple connections to their storage. These storage systems handle port failover and connection balancing transparently. This is often referred to as transparent failover.

Discovery, Authentication, and Access Control

You can use several mechanisms to discover your storage and to limit access to it.

You must configure your host and the iSCSI storage system to support your storage access control policy.

Discovery

A discovery session is part of the iSCSI protocol, and it returns the set of targets you can access on an iSCSI storage system. The two types of discovery available on ESXi are dynamic and static. Dynamic discovery obtains a list of accessible targets from the iSCSI storage system, while static discovery can only try to access one particular target by target name and address.

For more information, see [“Configuring Discovery Addresses for iSCSI Adapters,”](#) on page 97.

Authentication

iSCSI storage systems authenticate an initiator by a name and key pair. ESXi supports the CHAP protocol, which VMware recommends for your SAN implementation. To use CHAP authentication, the ESXi host and the iSCSI storage system must have CHAP enabled and have common credentials.

For information on enabling CHAP, see [“Configuring CHAP Parameters for iSCSI Adapters,”](#) on page 98.

Access Control

Access control is a policy set up on the iSCSI storage system. Most implementations support one or more of three types of access control:

- By initiator name
- By IP address
- By the CHAP protocol

Only initiators that meet all rules can access the iSCSI volume.

Using only CHAP for access control can slow down rescans because the ESXi host can discover all targets, but then fails at the authentication step. iSCSI rescans work faster if the host discovers only the targets it can authenticate.

Error Correction

To protect the integrity of iSCSI headers and data, the iSCSI protocol defines error correction methods known as header digests and data digests.

Both parameters are disabled by default, but you can enable them. These digests pertain to, respectively, the header and SCSI data being transferred between iSCSI initiators and targets, in both directions.

Header and data digests check the end-to-end, noncryptographic data integrity beyond the integrity checks that other networking layers provide, such as TCP and Ethernet. They check the entire communication path, including all elements that can change the network-level traffic, such as routers, switches, and proxies.

The existence and type of the digests are negotiated when an iSCSI connection is established. When the initiator and target agree on a digest configuration, this digest must be used for all traffic between them.

Enabling header and data digests does require additional processing for both the initiator and the target and can affect throughput and CPU use performance.

NOTE Systems that use Intel Nehalem processors offload the iSCSI digest calculations, thus reducing the impact on performance.

For information on enabling header and data digests, see [“Configuring Advanced Parameters for iSCSI,”](#) on page 102.

How Virtual Machines Access Data on an iSCSI SAN

ESXi stores a virtual machine's disk files within a VMFS datastore that resides on a SAN storage device. When virtual machine guest operating systems issue SCSI commands to their virtual disks, the SCSI virtualization layer translates these commands to VMFS file operations.

When a virtual machine interacts with its virtual disk stored on a SAN, the following process takes place:

- 1 When the guest operating system in a virtual machine reads or writes to SCSI disk, it issues SCSI commands to the virtual disk.
- 2 Device drivers in the virtual machine's operating system communicate with the virtual SCSI controllers.
- 3 The virtual SCSI controller forwards the command to the VMkernel.
- 4 The VMkernel performs the following tasks.
 - a Locates the file, which corresponds to the guest virtual machine disk, in the VMFS volume.
 - b Maps the requests for the blocks on the virtual disk to blocks on the appropriate physical device.
 - c Sends the modified I/O request from the device driver in the VMkernel to the iSCSI initiator (hardware or software).
- 5 If the iSCSI initiator is a hardware iSCSI adapter (both independent or dependent), the adapter performs the following tasks.
 - a Encapsulates I/O requests into iSCSI Protocol Data Units (PDUs).
 - b Encapsulates iSCSI PDUs into TCP/IP packets.
 - c Sends IP packets over Ethernet to the iSCSI storage system.
- 6 If the iSCSI initiator is a software iSCSI adapter, the following takes place.
 - a The iSCSI initiator encapsulates I/O requests into iSCSI PDUs.
 - b The initiator sends iSCSI PDUs through TCP/IP connections.
 - c The VMkernel TCP/IP stack relays TCP/IP packets to a physical NIC.
 - d The physical NIC sends IP packets over Ethernet to the iSCSI storage system.
- 7 Depending on which port the iSCSI initiator uses to connect to the network, Ethernet switches and routers carry the request to the storage device that the host wants to access.

Configuring iSCSI Adapters and Storage

10

Before ESXi can work with a SAN, you must set up your iSCSI adapters and storage.

To do this, you must first observe certain basic requirements and then follow best practices for installing and setting up hardware or software iSCSI adapters to access the SAN.

The following table lists the iSCSI adapters (vmhbas) that ESXi supports and indicates whether VMkernel networking configuration is required.

Table 10-1. Supported iSCSI adapters

iSCSI Adapter (vmhba)	Description	VMkernel Networking
Software	Uses standard NICs to connect your host to a remote iSCSI target on the IP network .	Required
Independent Hardware	Third-party adapter that offloads the iSCSI and network processing and management from your host.	Not required
Dependent Hardware	Third-party adapter that depends on VMware networking and iSCSI configuration and management interfaces.	Required

After you set up the iSCSI adapters, you can create a datastore on iSCSI storage. For details on how to create and manage datastores, see [“Creating Datastores,”](#) on page 160.

This chapter includes the following topics:

- [“ESXi iSCSI SAN Requirements,”](#) on page 70
- [“ESXi iSCSI SAN Restrictions,”](#) on page 70
- [“Setting LUN Allocations for iSCSI,”](#) on page 70
- [“Network Configuration and Authentication,”](#) on page 71
- [“Set Up Independent Hardware iSCSI Adapters,”](#) on page 71
- [“About Dependent Hardware iSCSI Adapters,”](#) on page 74
- [“About the Software iSCSI Adapter,”](#) on page 81
- [“Modify General Properties for iSCSI Adapters,”](#) on page 87
- [“Setting Up iSCSI Network,”](#) on page 88
- [“Using Jumbo Frames with iSCSI,”](#) on page 95
- [“Configuring Discovery Addresses for iSCSI Adapters,”](#) on page 97

- [“Configuring CHAP Parameters for iSCSI Adapters,”](#) on page 98
- [“Configuring Advanced Parameters for iSCSI,”](#) on page 102
- [“iSCSI Session Management,”](#) on page 103

ESXi iSCSI SAN Requirements

You must meet several requirements for your ESXi host to work properly with a SAN.

- Verify that your SAN storage hardware and firmware combinations are supported in conjunction with ESXi systems. For an up-to-date list, see *VMware Compatibility Guide*.
- Configure your system to have only one VMFS datastore for each LUN.
- Unless you are using diskless servers, set up a diagnostic partition on a local storage. If you have diskless servers that boot from iSCSI SAN, see [“General Boot from iSCSI SAN Recommendations,”](#) on page 107 for information about diagnostic partitions with iSCSI.
- Use RDMS for access to any raw disk. For information, see [Chapter 18, “Raw Device Mapping,”](#) on page 203.
- Set the SCSI controller driver in the guest operating system to a large enough queue. For information on changing queue depth for iSCSI adapters and virtual machines, see *vSphere Troubleshooting*.
- On virtual machines running Microsoft Windows, increase the value of the SCSI TimeoutValue parameter to allow Windows to better tolerate delayed I/O resulting from path failover. For information, see [“Set Timeout on Windows Guest OS,”](#) on page 185.

ESXi iSCSI SAN Restrictions

A number of restrictions exist when you use ESXi with an iSCSI SAN.

- ESXi does not support iSCSI-connected tape devices.
- You cannot use virtual-machine multipathing software to perform I/O load balancing to a single physical LUN.
- ESXi does not support multipathing when you combine independent hardware adapters with either software or dependent hardware adapters.

Setting LUN Allocations for iSCSI

When preparing your ESXi system to use iSCSI SAN storage you need to set LUN allocations.

Note the following points:

- **Storage Provisioning.** To ensure that the host recognizes LUNs at startup time, configure all iSCSI storage targets so that your host can access them and use them. Also, configure your host so that it can discover all available iSCSI targets.
- **vMotion and VMware DRS.** When you use vCenter Server and vMotion or DRS, make sure that the LUNs for the virtual machines are provisioned to all hosts. This configuration provides the greatest freedom in moving virtual machines.
- **Active-active versus active-passive arrays.** When you use vMotion or DRS with an active-passive SAN storage device, make sure that all hosts have consistent paths to all storage processors. Not doing so can cause path thrashing when a vMotion migration occurs.

For active-passive storage arrays not listed in Storage/SAN Compatibility, VMware does not support storage-port failover. You must connect the server to the active port on the storage system. This configuration ensures that the LUNs are presented to the host.

Network Configuration and Authentication

Before your ESXi host can discover iSCSI storage, the iSCSI initiators must be configured and authentication might have to be set up.

- For software iSCSI and dependent hardware iSCSI, networking for the VMkernel must be configured. You can verify the network configuration by using the `vmkping` utility. With software iSCSI and dependent iSCSI, IPv4 and IPv6 protocols are supported.
- For independent hardware iSCSI, network parameters, such as IP address, subnet mask, and default gateway must be configured on the HBA. You can also specify a network protocol, IPv4 or IPv6 for the adapter.
- Check and change the default initiator name if necessary.
- The dynamic discovery address or static discovery address and target name of the storage system must be set. For software iSCSI and dependent hardware iSCSI, the address should be pingable using `vmkping`.
- For CHAP authentication, enable it on the initiator and the storage system side. After authentication is enabled, it applies for all of the targets that are not yet discovered, but does not apply to targets that are already discovered. After the discovery address is set, the new targets discovered are exposed and can be used at that point.

For details on how to use the `vmkping` command, search the VMware Knowledge Base.

Set Up Independent Hardware iSCSI Adapters

An independent hardware iSCSI adapter is a specialized third-party adapter capable of accessing iSCSI storage over TCP/IP. This iSCSI adapter handles all iSCSI and network processing and management for your ESXi system.

Prerequisites

- Check whether the adapter needs to be licensed.
- Install the adapter.

For information about licencing, installation, and firmware updates, see vendor documentation.

Procedure

- 1 [View Independent Hardware iSCSI Adapters](#) on page 72
View an independent hardware iSCSI adapter to verify that it is correctly installed and ready for configuration.
- 2 [Modify General Properties for iSCSI Adapters](#) on page 72
You can change the default iSCSI name and alias assigned to your iSCSI adapters. For the for independent hardware iSCSI adapters, you can also change the default IP settings.
- 3 [Edit Network Settings for Hardware iSCSI](#) on page 73
After you install an independent hardware iSCSI adapter, you might need to change its default network settings so that the adapter is configured properly for the iSCSI SAN.
- 4 [Set Up Dynamic or Static Discovery for iSCSI](#) on page 73
With dynamic discovery, each time the initiator contacts a specified iSCSI storage system, it sends the `SendTargets` request to the system. The iSCSI system responds by supplying a list of available targets to the initiator. In addition to the dynamic discovery method, you can use static discovery and manually enter information for the targets.

What to do next

If required, configure CHAP parameters and jumbo frames.

View Independent Hardware iSCSI Adapters

View an independent hardware iSCSI adapter to verify that it is correctly installed and ready for configuration.

After you install an independent hardware iSCSI adapter on a host, it appears on the list of storage adapters available for configuration. You can view its properties.

Prerequisites

Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters**.

If installed, the hardware iSCSI adapter appears on the list of storage adapters.

- 4 Select the adapter to view.

The default details for the adapter appear, including the model, iSCSI name, iSCSI alias, IP address, and target and paths information.

Modify General Properties for iSCSI Adapters

You can change the default iSCSI name and alias assigned to your iSCSI adapters. For the for independent hardware iSCSI adapters, you can also change the default IP settings.

IMPORTANT When you modify any default properties for your iSCSI adapters, make sure to correctly format their names and IP addresses.

Prerequisites

Required privilege: **Host .Configuration.Storage Partition Configuration**

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters**, and select the adapter (vmhba#) to configure.
- 4 Under Adapter Details, click the **Properties** tab and click **Edit** in the General panel.
- 5 To change the default iSCSI name for your adapter, enter the new name.

Make sure the name you enter is worldwide unique and properly formatted or some storage devices might not recognize the iSCSI adapter.

- 6 (Optional) Enter the iSCSI alias.

The alias is a name that you use to identify the iSCSI adapter.

If you change the iSCSI name, it will be used for new iSCSI sessions. For existing sessions, new settings will not be used until logout and re-login.

Edit Network Settings for Hardware iSCSI

After you install an independent hardware iSCSI adapter, you might need to change its default network settings so that the adapter is configured properly for the iSCSI SAN.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters**, and select the adapter (vmhba#) to configure.
- 4 Under Adapter Details, click the **Network Settings** tab and click **Edit**.
- 5 In the IPv4 settings section, disable IPv6 or select the method used to obtain IP addresses.

NOTE The automatic DHCP option and static option are mutually exclusive.

Option	Description
No IPv4 settings	Disable IPv4.
Obtain IPv4 settings automatically	Use DHCP to obtain IP settings.
Use static IPv4 settings	Enter the IPv4 IP address, subnet mask, and default gateway for the iSCSI adapter.

- 6 In the IPv6 settings section, disable IPv6 or select an appropriate option for obtaining IPv6 addresses.

NOTE Automatic options and the static option are mutually exclusive.

Option	Description
No IPv6 settings	Disable IPv6.
Enable IPv6	Select an option for obtaining IPv6 addresses.
Obtain IPv6 addresses automatically through DHCP	Use DHCP to obtain IPv6 addresses.
Obtain IPv6 addresses automatically through Router Advertisement	Use router advertisement to obtain IPv6 addresses.
Override Link-local address for IPv6	Override the link-local IP address by configuring a static IP address.
Static IPv6 addresses	<ol style="list-style-type: none"> a Click Add to add a new IPv6 address. b Enter the IPv6 address and subnet prefix length, and click OK.

- 7 In the DNS settings section, provide IP addresses for a preferred DNS server and an alternate DNS server.

You must provide both values.

Set Up Dynamic or Static Discovery for iSCSI

With dynamic discovery, each time the initiator contacts a specified iSCSI storage system, it sends the SendTargets request to the system. The iSCSI system responds by supplying a list of available targets to the initiator. In addition to the dynamic discovery method, you can use static discovery and manually enter information for the targets.

When you set up static or dynamic discovery, you can only add new iSCSI targets. You cannot change any parameters of an existing target. To make changes, remove the existing target and add a new one.

Prerequisites

Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters** and select the iSCSI adapter to configure from the list.
- 4 Under Adapter Details, click the **Targets** tab.
- 5 Configure the discovery method.

Option	Description
Dynamic Discovery	<ol style="list-style-type: none"> a Click Dynamic Discovery and click Add. b Type the IP address or DNS name of the storage system and click OK. c Rescan the iSCSI adapter. <p>After establishing the SendTargets session with the iSCSI system, you host populates the Static Discovery list with all newly discovered targets.</p>
Static Discovery	<ol style="list-style-type: none"> a Click Static Discovery and click Add. b Enter the target's information and click OK. c Rescan the iSCSI adapter.

About Dependent Hardware iSCSI Adapters

A dependent hardware iSCSI adapter is a third-party adapter that depends on VMware networking, and iSCSI configuration and management interfaces provided by VMware.

An example of a dependent iSCSI adapter is a Broadcom 5709 NIC. When installed on a host, it presents its two components, a standard network adapter and an iSCSI engine, to the same port. The iSCSI engine appears on the list of storage adapters as an iSCSI adapter (vmhba). Although the iSCSI adapter is enabled by default, to make it functional, you must first connect it, through a virtual VMkernel adapter (vmk), to a physical network adapter (vmnic) associated with it. You can then configure the iSCSI adapter.

After you configure the dependent hardware iSCSI adapter, the discovery and authentication data are passed through the network connection, while the iSCSI traffic goes through the iSCSI engine, bypassing the network.

Dependent Hardware iSCSI Considerations

When you use dependent hardware iSCSI adapters with ESXi, certain considerations apply.

- When you use any dependent hardware iSCSI adapter, performance reporting for a NIC associated with the adapter might show little or no activity, even when iSCSI traffic is heavy. This behavior occurs because the iSCSI traffic bypasses the regular networking stack.
- If you use a third-party virtual switch, for example Cisco Nexus 1000V DVS, disable automatic pinning. Use manual pinning instead, making sure to connect a VMkernel adapter (vmk) to an appropriate physical NIC (vmnic). For information, refer to your virtual switch vendor documentation.
- The Broadcom iSCSI adapter performs data reassembly in hardware, which has a limited buffer space. When you use the Broadcom iSCSI adapter in a congested network or under heavy load, enable flow control to avoid performance degradation.

Flow control manages the rate of data transmission between two nodes to prevent a fast sender from overrunning a slow receiver. For best results, enable flow control at the end points of the I/O path, at the hosts and iSCSI storage systems.

To enable flow control for the host, use the `esxcli system module parameters` command. For details, see the VMware knowledge base article at <http://kb.vmware.com/kb/1013413>

- Dependent hardware adapters support IPv4 and IPv6.

Configure Dependent Hardware iSCSI Adapters

The entire setup and configuration process for the dependent hardware iSCSI adapters involves several steps. After you set up your adapter, you might need to configure CHAP parameters and Jumbo Frames.

Procedure

- 1 [View Dependent Hardware iSCSI Adapters](#) on page 75
View a dependent hardware iSCSI adapter to verify that it is correctly loaded.
- 2 [Modify General Properties for iSCSI Adapters](#) on page 76
You can change the default iSCSI name and alias assigned to your iSCSI adapters. For the for independent hardware iSCSI adapters, you can also change the default IP settings.
- 3 [Determine Association Between iSCSI and Network Adapters](#) on page 76
You create network connections to bind dependent iSCSI and physical network adapters. To create the connections correctly, you must determine the name of the physical NIC with which the dependent hardware iSCSI adapter is associated.
- 4 [Create Network Connections for iSCSI](#) on page 77
Configure connections for the traffic between the software or dependent hardware iSCSI adapters and the physical network adapters.
- 5 [Set Up Dynamic or Static Discovery for iSCSI](#) on page 80
With dynamic discovery, each time the initiator contacts a specified iSCSI storage system, it sends the SendTargets request to the system. The iSCSI system responds by supplying a list of available targets to the initiator. In addition to the dynamic discovery method, you can use static discovery and manually enter information for the targets.

What to do next

If required, configure CHAP parameters and jumbo frames.

View Dependent Hardware iSCSI Adapters

View a dependent hardware iSCSI adapter to verify that it is correctly loaded.

If installed, the dependent hardware iSCSI adapter (vmhba#) appears on the list of storage adapters under such category as, for example, Broadcom iSCSI Adapter. If the dependent hardware adapter does not appear on the list of storage adapters, check whether it needs to be licensed. See your vendor documentation.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters**.
- 4 Select the adapter (vmhba#) to view.

The default details for the adapter appear, including the iSCSI name, iSCSI alias, and the status.

What to do next

Although the dependent iSCSI adapter is enabled by default, to make it functional, you must set up networking for the iSCSI traffic and bind the adapter to the appropriate VMkernel iSCSI port. You then configure discovery addresses and CHAP parameters.

Modify General Properties for iSCSI Adapters

You can change the default iSCSI name and alias assigned to your iSCSI adapters. For the for independent hardware iSCSI adapters, you can also change the default IP settings.

IMPORTANT When you modify any default properties for your iSCSI adapters, make sure to correctly format their names and IP addresses.

Prerequisites

Required privilege: **Host .Configuration.Storage Partition Configuration**

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters**, and select the adapter (vmhba#) to configure.
- 4 Under Adapter Details, click the **Properties** tab and click **Edit** in the General panel.
- 5 To change the default iSCSI name for your adapter, enter the new name.

Make sure the name you enter is worldwide unique and properly formatted or some storage devices might not recognize the iSCSI adapter.

- 6 (Optional) Enter the iSCSI alias.

The alias is a name that you use to identify the iSCSI adapter.

If you change the iSCSI name, it will be used for new iSCSI sessions. For existing sessions, new settings will not be used until logout and re-login.

Determine Association Between iSCSI and Network Adapters

You create network connections to bind dependent iSCSI and physical network adapters. To create the connections correctly, you must determine the name of the physical NIC with which the dependent hardware iSCSI adapter is associated.

Prerequisites

In the vSphere Web Client, browse to the dependent hardware iSCSI adapter (vmhba#). See [“View Dependent Hardware iSCSI Adapters,”](#) on page 75.

Procedure

- 1 Select the iSCSI adapter (vmhba#) and click the **Network Port Binding** tab under Adapter Details.
- 2 Click **Add**.

The network adapter (vmnic#) that corresponds to the dependent iSCSI adapter is listed in the Physical Network Adapter column.

What to do next

If the VMkernel Adapter column is empty, create a VMkernel adapter (vmk#) for the physical network adapter (vmnic#) and then bind them to the associated dependent hardware iSCSI. See [“Setting Up iSCSI Network,”](#) on page 88.

Create Network Connections for iSCSI

Configure connections for the traffic between the software or dependent hardware iSCSI adapters and the physical network adapters.

The following tasks discuss the iSCSI network configuration with a vSphere standard switch.

If you use a vSphere distributed switch with multiple uplink ports, for port binding, create a separate distributed port group per each physical NIC. Then set the team policy so that each distributed port group has only one active uplink port. For detailed information on vSphere distributed switches, see the *vSphere Networking* documentation.

Procedure

- 1 [Create a Single VMkernel Adapter for iSCSI](#) on page 77
Connect the VMkernel, which runs services for iSCSI storage, to a physical network adapter.
- 2 [Create Additional VMkernel Adapters for iSCSI](#) on page 78
Use this task if you have two or more physical network adapters for iSCSI and you want to connect all of your physical adapters to a single vSphere standard switch. In this task, you add the physical adapters and VMkernel adapters to an existing vSphere standard switch.
- 3 [Change Network Policy for iSCSI](#) on page 79
If you use a single vSphere standard switch to connect multiple VMkernel adapters to multiple network adapters, set up network policy so that only one physical network adapter is active for each VMkernel adapter.
- 4 [Bind iSCSI and VMkernel Adapters](#) on page 79
Bind an iSCSI adapter with a VMkernel adapter.
- 5 [Review Port Binding Details](#) on page 80
Review networking details of the VMkernel adapter that is bound to the iSCSI adapter.

Create a Single VMkernel Adapter for iSCSI

Connect the VMkernel, which runs services for iSCSI storage, to a physical network adapter.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click **Actions > Add Networking**.
- 3 Select **VMkernel Network Adapter**, and click **Next**.
- 4 Select **New standard switch** to create a vSphere standard switch.
- 5 Click the **Add adapters** icon, and select the network adapter (vmnic#) to use for iSCSI.
Make sure to assign the adapter to Active Adapters.

IMPORTANT If you are creating a VMkernel adapter for dependent hardware iSCSI, select the network adapter that corresponds to the iSCSI component. See [“Determine Association Between iSCSI and Network Adapters,”](#) on page 76.

- 6 Enter a network label.
A network label is a friendly name that identifies the VMkernel adapter that you are creating, for example, iSCSI.
- 7 Specify the IP settings.
- 8 Review the information and click **Finish**.

You created the virtual VMkernel adapter (vmk#) for a physical network adapter (vmnic#) on your host.

What to do next

If your host has one physical network adapter for iSCSI traffic, you must bind the virtual adapter that you created to the iSCSI adapter.

If you have multiple network adapters, create additional VMkernel adapters and then perform iSCSI binding. The number of virtual adapters must correspond to the number of physical adapters on the host.

Create Additional VMkernel Adapters for iSCSI

Use this task if you have two or more physical network adapters for iSCSI and you want to connect all of your physical adapters to a single vSphere standard switch. In this task, you add the physical adapters and VMkernel adapters to an existing vSphere standard switch.

Prerequisites

Create a vSphere standard switch that maps an iSCSI VMkernel adapter to a single physical network adapter designated for iSCSI traffic.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Networking**.
- 3 Click **Virtual Switches**, and select the vSphere switch that you want to modify from the list.
- 4 Connect additional network adapters to the switch.
 - a Click the **Add host networking** icon.
 - b Select **Physical Network Adapters**, and click **Next**.
 - c Make sure that you are using the existing switch, and click **Next**.
 - d Click the **Add adapters** icon, and select one or more network adapters (vmnic#) to use for iSCSI.

With dependent hardware iSCSI adapters, select only those NICs that have a corresponding iSCSI component.
 - e Complete configuration, and click **Finish**.
- 5 Create iSCSI VMkernel adapters for all physical network adapters that you added.

The number of VMkernel interfaces must correspond to the number of physical network adapters on the vSphere standard switch.

 - a Click the **Add host networking** icon.
 - b Select **VMkernel Network Adapter**, and click **Next**.
 - c Make sure that you are using the existing switch, and click **Next**.
 - d Complete configuration, and click **Finish**.

What to do next

Change the network policy for all VMkernel adapters, so that only one physical network adapter is active for each VMkernel adapter. You can then bind the iSCSI VMkernel adapters to the software iSCSI or dependent hardware iSCSI adapters.

Change Network Policy for iSCSI

If you use a single vSphere standard switch to connect multiple VMkernel adapters to multiple network adapters, set up network policy so that only one physical network adapter is active for each VMkernel adapter.

By default, for each VMkernel adapter on the vSphere standard switch, all network adapters appear as active. You must override this setup, so that each VMkernel adapter maps to only one corresponding active physical. For example, vmk1 maps to vmnic1, vmk2 maps to vmnic2, and so on.

Prerequisites

Create a vSphere standard switch that connects VMkernel with physical network adapters designated for iSCSI traffic. The number of VMkernel adapters must correspond to the number of physical adapters on the vSphere standard switch.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Networking**.
- 3 Click **Virtual Switches**, and select the vSphere switch that you want to modify from the list.
- 4 On the vSwitch diagram, select the VMkernel adapter and click the **Edit Settings** icon.
- 5 On the Edit Settings wizard, click **Teaming and Failover** and click **Override** under Failover Order.
- 6 Designate only one physical adapter as active and move all remaining adapters to the **Unused Adapters** category.
- 7 Repeat [Step 4](#) through [Step 6](#) for each iSCSI VMkernel interface on the vSphere standard switch.

Example: iSCSI Network Policy

The following table illustrates the proper iSCSI mapping where only one physical network adapter is active for each VMkernel adapter.

VMkernel Adapter (vmk#)	Physical Network Adapter (vmnic#)
vmk1	Active Adapters
	vmnic1
	Unused Adapters
	vmnic2
vmk2	Active Adapters
	vmnic2
	Unused Adapters
	vmnic1

What to do next

After you perform this task, bind the virtual VMkernel adapters to the software iSCSI or dependent hardware iSCSI adapters.

Bind iSCSI and VMkernel Adapters

Bind an iSCSI adapter with a VMkernel adapter.

Prerequisites

Create a virtual VMkernel adapter for each physical network adapter on your host. If you use multiple VMkernel adapters, set up the correct network policy.

Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters** and select the software or dependent iSCSI adapter to configure from the list.
- 4 Under Adapter Details, click the **Network Port Binding** tab and click **Add**.
- 5 Select a VMkernel adapter to bind with the iSCSI adapter.

NOTE Make sure that the network policy for the VMkernel adapter is compliant with the binding requirements.

You can bind the software iSCSI adapter to one or more VMkernel adapters. For a dependent hardware iSCSI adapter, only one VMkernel adapter associated with the correct physical NIC is available.

- 6 Click **OK**.

The network connection appears on the list of VMkernel port bindings for the iSCSI adapter.

Review Port Binding Details

Review networking details of the VMkernel adapter that is bound to the iSCSI adapter.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters** and select the software or dependent iSCSI adapter from the list.
- 4 Under Adapter Details, click the **Network Port Binding** tab and click **View Details**.
- 5 Review the VMkernel adapter information by switching between available tabs.

Set Up Dynamic or Static Discovery for iSCSI

With dynamic discovery, each time the initiator contacts a specified iSCSI storage system, it sends the SendTargets request to the system. The iSCSI system responds by supplying a list of available targets to the initiator. In addition to the dynamic discovery method, you can use static discovery and manually enter information for the targets.

When you set up static or dynamic discovery, you can only add new iSCSI targets. You cannot change any parameters of an existing target. To make changes, remove the existing target and add a new one.

Prerequisites

Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters** and select the iSCSI adapter to configure from the list.
- 4 Under Adapter Details, click the **Targets** tab.

- 5 Configure the discovery method.

Option	Description
Dynamic Discovery	<ol style="list-style-type: none"> Click Dynamic Discovery and click Add. Type the IP address or DNS name of the storage system and click OK. Rescan the iSCSI adapter. <p>After establishing the SendTargets session with the iSCSI system, you host populates the Static Discovery list with all newly discovered targets.</p>
Static Discovery	<ol style="list-style-type: none"> Click Static Discovery and click Add. Enter the target's information and click OK Rescan the iSCSI adapter.

About the Software iSCSI Adapter

With the software-based iSCSI implementation, you can use standard NICs to connect your host to a remote iSCSI target on the IP network. The software iSCSI adapter that is built into ESXi facilitates this connection by communicating with the physical NICs through the network stack.

Before you can use the software iSCSI adapter, you must set up networking, activate the adapter, and configure parameters such as discovery addresses and CHAP.

NOTE Designate a separate network adapter for iSCSI. Do not use iSCSI on 100Mbps or slower adapters.

Configure the Software iSCSI Adapter

The software iSCSI adapter configuration workflow includes these steps.

Procedure

- 1 [Activate the Software iSCSI Adapter](#) on page 81
You must activate your software iSCSI adapter so that your host can use it to access iSCSI storage.
- 2 [Modify General Properties for iSCSI Adapters](#) on page 82
You can change the default iSCSI name and alias assigned to your iSCSI adapters. For the for independent hardware iSCSI adapters, you can also change the default IP settings.
- 3 [Create Network Connections for iSCSI](#) on page 83
Configure connections for the traffic between the software or dependent hardware iSCSI adapters and the physical network adapters.
- 4 [Set Up Dynamic or Static Discovery for iSCSI](#) on page 86
With dynamic discovery, each time the initiator contacts a specified iSCSI storage system, it sends the SendTargets request to the system. The iSCSI system responds by supplying a list of available targets to the initiator. In addition to the dynamic discovery method, you can use static discovery and manually enter information for the targets.

What to do next

If required, configure CHAP parameters and jumbo frames.

Activate the Software iSCSI Adapter

You must activate your software iSCSI adapter so that your host can use it to access iSCSI storage.

You can activate only one software iSCSI adapter.

Prerequisites

Required privilege: **Host.Configuration.Storage Partition Configuration**

NOTE If you boot from iSCSI using the software iSCSI adapter, the adapter is enabled and the network configuration is created at the first boot. If you disable the adapter, it is reenabled each time you boot the host.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters**, and click **Add**.
- 4 Select **Software iSCSI Adapter** and confirm that you want to add the adapter.

The software iSCSI adapter (vmhba#) is enabled and appears on the list of storage adapters. After enabling the adapter, the host assigns the default iSCSI name to it. If you need to change the default name, follow iSCSI naming conventions.

What to do next

Select the adapter and use the Adapter Details section to complete configuration.

Modify General Properties for iSCSI Adapters

You can change the default iSCSI name and alias assigned to your iSCSI adapters. For the for independent hardware iSCSI adapters, you can also change the default IP settings.

IMPORTANT When you modify any default properties for your iSCSI adapters, make sure to correctly format their names and IP addresses.

Prerequisites

Required privilege: **Host .Configuration.Storage Partition Configuration**

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters**, and select the adapter (vmhba#) to configure.
- 4 Under Adapter Details, click the **Properties** tab and click **Edit** in the General panel.
- 5 To change the default iSCSI name for your adapter, enter the new name.

Make sure the name you enter is worldwide unique and properly formatted or some storage devices might not recognize the iSCSI adapter.

- 6 (Optional) Enter the iSCSI alias.

The alias is a name that you use to identify the iSCSI adapter.

If you change the iSCSI name, it will be used for new iSCSI sessions. For existing sessions, new settings will not be used until logout and re-login.

Create Network Connections for iSCSI

Configure connections for the traffic between the software or dependent hardware iSCSI adapters and the physical network adapters.

The following tasks discuss the iSCSI network configuration with a vSphere standard switch.

If you use a vSphere distributed switch with multiple uplink ports, for port binding, create a separate distributed port group per each physical NIC. Then set the team policy so that each distributed port group has only one active uplink port. For detailed information on vSphere distributed switches, see the *vSphere Networking* documentation.

Procedure

- 1 [Create a Single VMkernel Adapter for iSCSI](#) on page 83
Connect the VMkernel, which runs services for iSCSI storage, to a physical network adapter.
- 2 [Create Additional VMkernel Adapters for iSCSI](#) on page 84
Use this task if you have two or more physical network adapters for iSCSI and you want to connect all of your physical adapters to a single vSphere standard switch. In this task, you add the physical adapters and VMkernel adapters to an existing vSphere standard switch.
- 3 [Change Network Policy for iSCSI](#) on page 85
If you use a single vSphere standard switch to connect multiple VMkernel adapters to multiple network adapters, set up network policy so that only one physical network adapter is active for each VMkernel adapter.
- 4 [Bind iSCSI and VMkernel Adapters](#) on page 85
Bind an iSCSI adapter with a VMkernel adapter.
- 5 [Review Port Binding Details](#) on page 86
Review networking details of the VMkernel adapter that is bound to the iSCSI adapter.

Create a Single VMkernel Adapter for iSCSI

Connect the VMkernel, which runs services for iSCSI storage, to a physical network adapter.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click **Actions > Add Networking**.
- 3 Select **VMkernel Network Adapter**, and click **Next**.
- 4 Select **New standard switch** to create a vSphere standard switch.
- 5 Click the **Add adapters** icon, and select the network adapter (vmnic#) to use for iSCSI.
Make sure to assign the adapter to Active Adapters.

IMPORTANT If you are creating a VMkernel adapter for dependent hardware iSCSI, select the network adapter that corresponds to the iSCSI component. See [“Determine Association Between iSCSI and Network Adapters,”](#) on page 76.

- 6 Enter a network label.
A network label is a friendly name that identifies the VMkernel adapter that you are creating, for example, iSCSI.
- 7 Specify the IP settings.
- 8 Review the information and click **Finish**.

You created the virtual VMkernel adapter (vmk#) for a physical network adapter (vmnic#) on your host.

What to do next

If your host has one physical network adapter for iSCSI traffic, you must bind the virtual adapter that you created to the iSCSI adapter.

If you have multiple network adapters, create additional VMkernel adapters and then perform iSCSI binding. The number of virtual adapters must correspond to the number of physical adapters on the host.

Create Additional VMkernel Adapters for iSCSI

Use this task if you have two or more physical network adapters for iSCSI and you want to connect all of your physical adapters to a single vSphere standard switch. In this task, you add the physical adapters and VMkernel adapters to an existing vSphere standard switch.

Prerequisites

Create a vSphere standard switch that maps an iSCSI VMkernel adapter to a single physical network adapter designated for iSCSI traffic.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Networking**.
- 3 Click **Virtual Switches**, and select the vSphere switch that you want to modify from the list.
- 4 Connect additional network adapters to the switch.
 - a Click the **Add host networking** icon.
 - b Select **Physical Network Adapters**, and click **Next**.
 - c Make sure that you are using the existing switch, and click **Next**.
 - d Click the **Add adapters** icon, and select one or more network adapters (vmnic#) to use for iSCSI.

With dependent hardware iSCSI adapters, select only those NICs that have a corresponding iSCSI component.
 - e Complete configuration, and click **Finish**.
- 5 Create iSCSI VMkernel adapters for all physical network adapters that you added.

The number of VMkernel interfaces must correspond to the number of physical network adapters on the vSphere standard switch.

 - a Click the **Add host networking** icon.
 - b Select **VMkernel Network Adapter**, and click **Next**.
 - c Make sure that you are using the existing switch, and click **Next**.
 - d Complete configuration, and click **Finish**.

What to do next

Change the network policy for all VMkernel adapters, so that only one physical network adapter is active for each VMkernel adapter. You can then bind the iSCSI VMkernel adapters to the software iSCSI or dependent hardware iSCSI adapters.

Change Network Policy for iSCSI

If you use a single vSphere standard switch to connect multiple VMkernel adapters to multiple network adapters, set up network policy so that only one physical network adapter is active for each VMkernel adapter.

By default, for each VMkernel adapter on the vSphere standard switch, all network adapters appear as active. You must override this setup, so that each VMkernel adapter maps to only one corresponding active physical. For example, vmk1 maps to vmnic1, vmk2 maps to vmnic2, and so on.

Prerequisites

Create a vSphere standard switch that connects VMkernel with physical network adapters designated for iSCSI traffic. The number of VMkernel adapters must correspond to the number of physical adapters on the vSphere standard switch.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Networking**.
- 3 Click **Virtual Switches**, and select the vSphere switch that you want to modify from the list.
- 4 On the vSwitch diagram, select the VMkernel adapter and click the **Edit Settings** icon.
- 5 On the Edit Settings wizard, click **Teaming and Failover** and click **Override** under Failover Order.
- 6 Designate only one physical adapter as active and move all remaining adapters to the **Unused Adapters** category.
- 7 Repeat [Step 4](#) through [Step 6](#) for each iSCSI VMkernel interface on the vSphere standard switch.

Example: iSCSI Network Policy

The following table illustrates the proper iSCSI mapping where only one physical network adapter is active for each VMkernel adapter.

VMkernel Adapter (vmk#)	Physical Network Adapter (vmnic#)
vmk1	Active Adapters
	vmnic1
	Unused Adapters
	vmnic2
vmk2	Active Adapters
	vmnic2
	Unused Adapters
	vmnic1

What to do next

After you perform this task, bind the virtual VMkernel adapters to the software iSCSI or dependent hardware iSCSI adapters.

Bind iSCSI and VMkernel Adapters

Bind an iSCSI adapter with a VMkernel adapter.

Prerequisites

Create a virtual VMkernel adapter for each physical network adapter on your host. If you use multiple VMkernel adapters, set up the correct network policy.

Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters** and select the software or dependent iSCSI adapter to configure from the list.
- 4 Under Adapter Details, click the **Network Port Binding** tab and click **Add**.
- 5 Select a VMkernel adapter to bind with the iSCSI adapter.

NOTE Make sure that the network policy for the VMkernel adapter is compliant with the binding requirements.

You can bind the software iSCSI adapter to one or more VMkernel adapters. For a dependent hardware iSCSI adapter, only one VMkernel adapter associated with the correct physical NIC is available.

- 6 Click **OK**.

The network connection appears on the list of VMkernel port bindings for the iSCSI adapter.

Review Port Binding Details

Review networking details of the VMkernel adapter that is bound to the iSCSI adapter.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters** and select the software or dependent iSCSI adapter from the list.
- 4 Under Adapter Details, click the **Network Port Binding** tab and click **View Details**.
- 5 Review the VMkernel adapter information by switching between available tabs.

Set Up Dynamic or Static Discovery for iSCSI

With dynamic discovery, each time the initiator contacts a specified iSCSI storage system, it sends the SendTargets request to the system. The iSCSI system responds by supplying a list of available targets to the initiator. In addition to the dynamic discovery method, you can use static discovery and manually enter information for the targets.

When you set up static or dynamic discovery, you can only add new iSCSI targets. You cannot change any parameters of an existing target. To make changes, remove the existing target and add a new one.

Prerequisites

Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters** and select the iSCSI adapter to configure from the list.
- 4 Under Adapter Details, click the **Targets** tab.

- 5 Configure the discovery method.

Option	Description
Dynamic Discovery	<ol style="list-style-type: none"> a Click Dynamic Discovery and click Add. b Type the IP address or DNS name of the storage system and click OK. c Rescan the iSCSI adapter. <p>After establishing the SendTargets session with the iSCSI system, you host populates the Static Discovery list with all newly discovered targets.</p>
Static Discovery	<ol style="list-style-type: none"> a Click Static Discovery and click Add. b Enter the target's information and click OK c Rescan the iSCSI adapter.

Disable Software iSCSI Adapter

If you do not need the software iSCSI adapter, you can disable it.

Disabling the software iSCSI adapter marks it for removal. The adapter is removed from the host on the next host reboot. After removal, all virtual machines and other data on the storage devices associated with this adapter become inaccessible to the host.

Prerequisites

Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters** and select the software iSCSI adapter from the list.
- 4 Under Adapter Details, click the **Properties** tab.
- 5 Click **Disable** and confirm that you want to disable the adapter.

The status indicates that the adapter is disabled.

- 6 Reboot the host.

After reboot, the adapter no longer appears on the list of storage adapters.

The iSCSI software adapter is no longer available and storage devices associated with it are inaccessible. You can later activate the adapter.

Modify General Properties for iSCSI Adapters

You can change the default iSCSI name and alias assigned to your iSCSI adapters. For the for independent hardware iSCSI adapters, you can also change the default IP settings.

IMPORTANT When you modify any default properties for your iSCSI adapters, make sure to correctly format their names and IP addresses.

Prerequisites

Required privilege: **Host .Configuration.Storage Partition Configuration**

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.

- 3 Click **Storage Adapters**, and select the adapter (vmhba#) to configure.
- 4 Under Adapter Details, click the **Properties** tab and click **Edit** in the General panel.
- 5 To change the default iSCSI name for your adapter, enter the new name.

Make sure the name you enter is worldwide unique and properly formatted or some storage devices might not recognize the iSCSI adapter.

- 6 (Optional) Enter the iSCSI alias.

The alias is a name that you use to identify the iSCSI adapter.

If you change the iSCSI name, it will be used for new iSCSI sessions. For existing sessions, new settings will not be used until logout and re-login.

Setting Up iSCSI Network

Software and dependent hardware iSCSI adapters depend on VMkernel networking. If you use the software or dependent hardware iSCSI adapters, you must configure connections for the traffic between the iSCSI component and the physical network adapters.

Configuring the network connection involves creating a virtual VMkernel adapter for each physical network adapter. You then associate the VMkernel adapter with an appropriate iSCSI adapter. This process is called port binding.

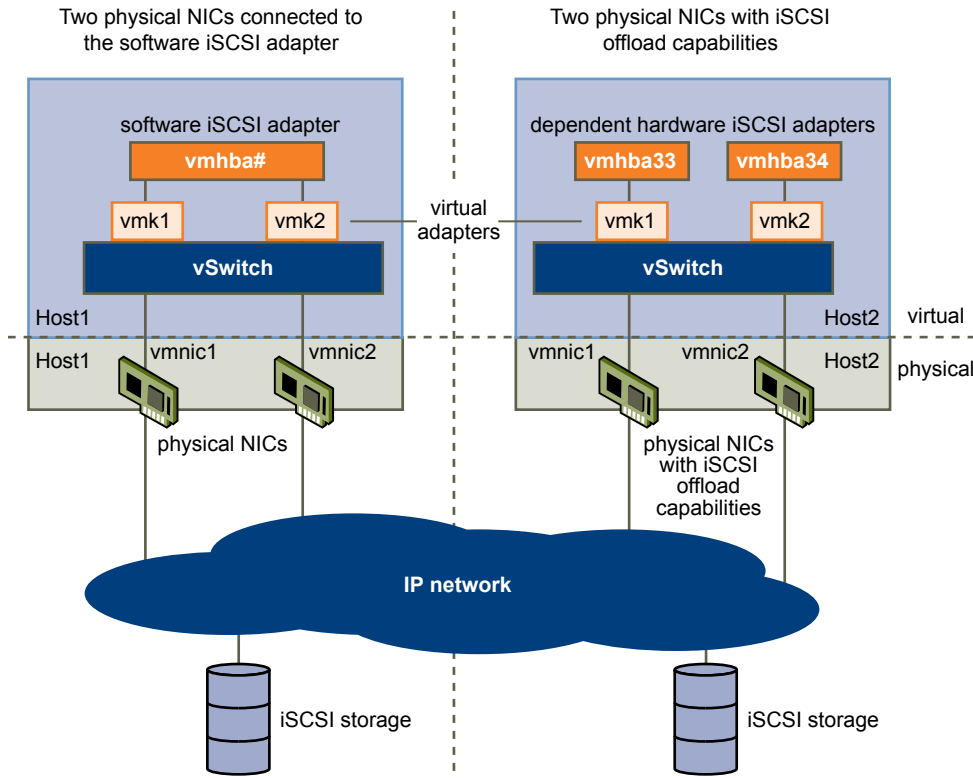
For specific considerations on when and how to use network connections with software iSCSI, see the VMware knowledge base article at <http://kb.vmware.com/kb/2038869>.

Multiple Network Adapters in iSCSI Configuration

If your host has more than one physical network adapter for software and dependent hardware iSCSI, use the adapters for multipathing.

You can connect the software iSCSI adapter with any physical NICs available on your host. The dependent iSCSI adapters must be connected only to their own physical NICs.

NOTE Physical NICs must be on the same subnet as the iSCSI storage system they connect to.

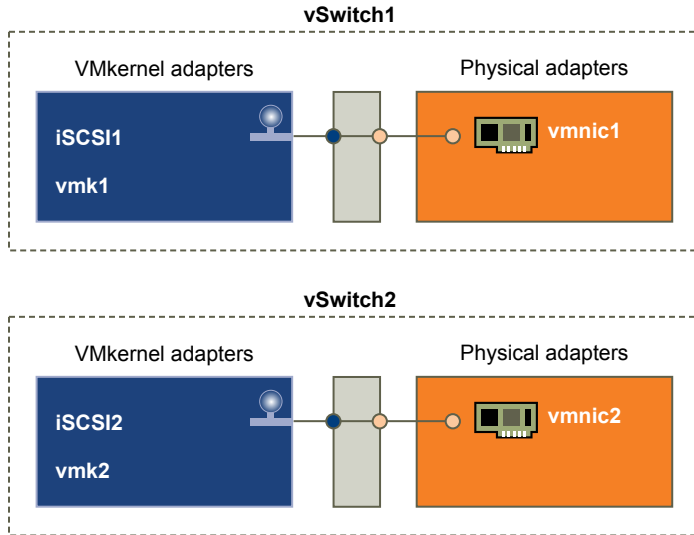
Figure 10-1. Networking with iSCSI

The iSCSI adapter and physical NIC connect through a virtual VMkernel adapter, also called virtual network adapter or VMkernel port. You create a VMkernel adapter (vmk) on a vSphere switch (vSwitch) using 1:1 mapping between each virtual and physical network adapter.

One way to achieve the 1:1 mapping when you have multiple NICs, is to designate a separate vSphere switch for each virtual-to-physical adapter pair.

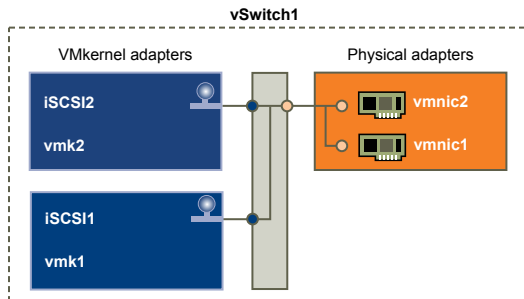
NOTE If you use separate vSphere switches, you must connect them to different IP subnets. Otherwise, VMkernel adapters might experience connectivity problems and the host will fail to discover iSCSI LUNs.

The following examples show configurations that use vSphere standard switches, but you can use distributed switches as well. For more information about vSphere distributed switches, see the *vSphere Networking* documentation.

Figure 10-2. 1:1 adapter mapping on separate vSphere standard switches

An alternative is to add all NICs and VMkernel adapters to a single vSphere standard switch. In this case, you must override the default network setup and make sure that each VMkernel adapter maps to only one corresponding active physical adapter.

NOTE You must use the single vSwitch configuration if VMkernel adapters are on the same subnet.

Figure 10-3. 1:1 adapter mapping on a single vSphere standard switch

The following table summarises the iSCSI networking configuration discussed in this topic.

Table 10-2. Networking configuration for iSCSI

iSCSI Adapters	VMkernel Adapters (Ports)	Physical Adapters (NICs)
Software iSCSI		
vmhba32	vmk1	vmnic1
	vmk2	vmnic2
Dependent Hardware iSCSI		
vmhba33	vmk1	vmnic1
vmhba34	vmk2	vmnic2

Guidelines for Using iSCSI Port Binding in ESXi

You can use multiple VMkernel adapters bound to iSCSI to have multiple paths to an iSCSI array that broadcasts a single IP address.

When you use port binding for multipathing, follow these guidelines:

- iSCSI ports of the array target must reside in the same broadcast domain and IP subnet as the VMkernel adapters.
- All VMkernel adapters used for iSCSI port binding must reside in the same broadcast domain and IP subnet.
- All VMkernel adapters used for iSCSI connectivity must reside in the same virtual switch.
- Port binding does not support network routing.

Do not use port binding when any of the following conditions exist:

- Array target iSCSI ports are in a different broadcast domain and IP subnet.
- VMkernel adapters used for iSCSI connectivity exist in different broadcast domains, IP subnets, or use different virtual switches.
- Routing is required to reach the iSCSI array.

Create Network Connections for iSCSI

Configure connections for the traffic between the software or dependent hardware iSCSI adapters and the physical network adapters.

The following tasks discuss the iSCSI network configuration with a vSphere standard switch.

If you use a vSphere distributed switch with multiple uplink ports, for port binding, create a separate distributed port group per each physical NIC. Then set the team policy so that each distributed port group has only one active uplink port. For detailed information on vSphere distributed switches, see the *vSphere Networking* documentation.

Procedure

- 1 [Create a Single VMkernel Adapter for iSCSI](#) on page 92
Connect the VMkernel, which runs services for iSCSI storage, to a physical network adapter.
- 2 [Create Additional VMkernel Adapters for iSCSI](#) on page 92
Use this task if you have two or more physical network adapters for iSCSI and you want to connect all of your physical adapters to a single vSphere standard switch. In this task, you add the physical adapters and VMkernel adapters to an existing vSphere standard switch.
- 3 [Change Network Policy for iSCSI](#) on page 93
If you use a single vSphere standard switch to connect multiple VMkernel adapters to multiple network adapters, set up network policy so that only one physical network adapter is active for each VMkernel adapter.
- 4 [Bind iSCSI and VMkernel Adapters](#) on page 94
Bind an iSCSI adapter with a VMkernel adapter.
- 5 [Review Port Binding Details](#) on page 94
Review networking details of the VMkernel adapter that is bound to the iSCSI adapter.

Create a Single VMkernel Adapter for iSCSI

Connect the VMkernel, which runs services for iSCSI storage, to a physical network adapter.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click **Actions > Add Networking**.
- 3 Select **VMkernel Network Adapter**, and click **Next**.
- 4 Select **New standard switch** to create a vSphere standard switch.
- 5 Click the **Add adapters** icon, and select the network adapter (vmnic#) to use for iSCSI.

Make sure to assign the adapter to Active Adapters.

IMPORTANT If you are creating a VMkernel adapter for dependent hardware iSCSI, select the network adapter that corresponds to the iSCSI component. See [“Determine Association Between iSCSI and Network Adapters,”](#) on page 76.

- 6 Enter a network label.

A network label is a friendly name that identifies the VMkernel adapter that you are creating, for example, iSCSI.

- 7 Specify the IP settings.
- 8 Review the information and click **Finish**.

You created the virtual VMkernel adapter (vmk#) for a physical network adapter (vmnic#) on your host.

What to do next

If your host has one physical network adapter for iSCSI traffic, you must bind the virtual adapter that you created to the iSCSI adapter.

If you have multiple network adapters, create additional VMkernel adapters and then perform iSCSI binding. The number of virtual adapters must correspond to the number of physical adapters on the host.

Create Additional VMkernel Adapters for iSCSI

Use this task if you have two or more physical network adapters for iSCSI and you want to connect all of your physical adapters to a single vSphere standard switch. In this task, you add the physical adapters and VMkernel adapters to an existing vSphere standard switch.

Prerequisites

Create a vSphere standard switch that maps an iSCSI VMkernel adapter to a single physical network adapter designated for iSCSI traffic.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Networking**.
- 3 Click **Virtual Switches**, and select the vSphere switch that you want to modify from the list.
- 4 Connect additional network adapters to the switch.
 - a Click the **Add host networking** icon.
 - b Select **Physical Network Adapters**, and click **Next**.

- c Make sure that you are using the existing switch, and click **Next**.
 - d Click the **Add adapters** icon, and select one or more network adapters (vmnic#) to use for iSCSI. With dependent hardware iSCSI adapters, select only those NICs that have a corresponding iSCSI component.
 - e Complete configuration, and click **Finish**.
- 5 Create iSCSI VMkernel adapters for all physical network adapters that you added.
- The number of VMkernel interfaces must correspond to the number of physical network adapters on the vSphere standard switch.
- a Click the **Add host networking** icon.
 - b Select **VMkernel Network Adapter**, and click **Next**.
 - c Make sure that you are using the existing switch, and click **Next**.
 - d Complete configuration, and click **Finish**.

What to do next

Change the network policy for all VMkernel adapters, so that only one physical network adapter is active for each VMkernel adapter. You can then bind the iSCSI VMkernel adapters to the software iSCSI or dependent hardware iSCSI adapters.

Change Network Policy for iSCSI

If you use a single vSphere standard switch to connect multiple VMkernel adapters to multiple network adapters, set up network policy so that only one physical network adapter is active for each VMkernel adapter.

By default, for each VMkernel adapter on the vSphere standard switch, all network adapters appear as active. You must override this setup, so that each VMkernel adapter maps to only one corresponding active physical. For example, vmk1 maps to vmnic1, vmk2 maps to vmnic2, and so on.

Prerequisites

Create a vSphere standard switch that connects VMkernel with physical network adapters designated for iSCSI traffic. The number of VMkernel adapters must correspond to the number of physical adapters on the vSphere standard switch.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Networking**.
- 3 Click **Virtual Switches**, and select the vSphere switch that you want to modify from the list.
- 4 On the vSwitch diagram, select the VMkernel adapter and click the **Edit Settings** icon.
- 5 On the Edit Settings wizard, click **Teaming and Failover** and click **Override** under Failover Order.
- 6 Designate only one physical adapter as active and move all remaining adapters to the **Unused Adapters** category.
- 7 Repeat [Step 4](#) through [Step 6](#) for each iSCSI VMkernel interface on the vSphere standard switch.

Example: iSCSI Network Policy

The following table illustrates the proper iSCSI mapping where only one physical network adapter is active for each VMkernel adapter.

VMkernel Adapter (vmk#)	Physical Network Adapter (vmnic#)
vmk1	Active Adapters vmnic1 Unused Adapters vmnic2
vmk2	Active Adapters vmnic2 Unused Adapters vmnic1

What to do next

After you perform this task, bind the virtual VMkernel adapters to the software iSCSI or dependent hardware iSCSI adapters.

Bind iSCSI and VMkernel Adapters

Bind an iSCSI adapter with a VMkernel adapter.

Prerequisites

Create a virtual VMkernel adapter for each physical network adapter on your host. If you use multiple VMkernel adapters, set up the correct network policy.

Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters** and select the software or dependent iSCSI adapter to configure from the list.
- 4 Under Adapter Details, click the **Network Port Binding** tab and click **Add**.
- 5 Select a VMkernel adapter to bind with the iSCSI adapter.

NOTE Make sure that the network policy for the VMkernel adapter is compliant with the binding requirements.

You can bind the software iSCSI adapter to one or more VMkernel adapters. For a dependent hardware iSCSI adapter, only one VMkernel adapter associated with the correct physical NIC is available.

- 6 Click **OK**.

The network connection appears on the list of VMkernel port bindings for the iSCSI adapter.

Review Port Binding Details

Review networking details of the VMkernel adapter that is bound to the iSCSI adapter.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters** and select the software or dependent iSCSI adapter from the list.
- 4 Under Adapter Details, click the **Network Port Binding** tab and click **View Details**.
- 5 Review the VMkernel adapter information by switching between available tabs.

Managing iSCSI Network

Special consideration apply to network adapters, both physical and VMkernel, that are associated with an iSCSI adapter.

After you create network connections for iSCSI, an iSCSI indicator on a number of Networking dialog boxes becomes enabled. This indicator shows that a particular virtual or physical network adapter is iSCSI-bound. To avoid disruptions in iSCSI traffic, follow these guidelines and considerations when managing iSCSI-bound virtual and physical network adapters:

- Make sure that the VMkernel network adapters are assigned addresses on the same subnet as the iSCSI storage portal they connect to.
- iSCSI adapters using VMkernel adapters are not able to connect to iSCSI ports on different subnets, even if those ports are discovered by the iSCSI adapters.
- When using separate vSphere switches to connect physical network adapters and VMkernel adapters, make sure that the vSphere switches connect to different IP subnets.
- If VMkernel adapters are on the same subnet, they must connect to a single vSwitch.
- If you migrate VMkernel adapters to a different vSphere switch, move associated physical adapters.
- Do not make configuration changes to iSCSI-bound VMkernel adapters or physical network adapters.
- Do not make changes that might break association of VMkernel adapters and physical network adapters. You can break the association if you remove one of the adapters or the vSphere switch that connects them, or change the 1:1 network policy for their connection.

iSCSI Network Troubleshooting

A warning sign indicates non-compliant port group policy for an iSCSI-bound VMkernel adapter.

Problem

The VMkernel adapter's port group policy is considered non-compliant in the following cases:

- The VMkernel adapter is not connected to an active physical network adapter.
- The VMkernel adapter is connected to more than one physical network adapter.
- The VMkernel adapter is connected to one or more standby physical adapters.
- The active physical adapter is changed.

Solution

Follow the steps in [“Change Network Policy for iSCSI,”](#) on page 93 to set up the correct network policy for the iSCSI-bound VMkernel adapter.

Using Jumbo Frames with iSCSI

ESXi supports the use of Jumbo Frames with iSCSI.

Jumbo Frames are Ethernet frames with the size that exceeds 1500 Bytes. The maximum transmission unit (MTU) parameter is typically used to measure the size of Jumbo Frames. ESXi allows Jumbo Frames with the MTU up to 9000 Bytes.

When you use Jumbo Frames for iSCSI traffic, the following considerations apply:

- The network must support Jumbo Frames end-to-end for Jumbo Frames to be effective.
- Check with your vendors to ensure your physical NICs and iSCSI HBAs support Jumbo Frames.

- To set up and verify physical network switches for Jumbo Frames, consult your vendor documentation.

The following table explains the level of support that ESXi provides to Jumbo Frames.

Table 10-3. Support of Jumbo Frames

Type of iSCSI Adapters	Jumbo Frames Support
Software iSCSI	Supported
Dependent Hardware iSCSI	Supported. Check with vendor.
Independent Hardware iSCSI	Supported. Check with vendor.

Enable Jumbo Frames for Software and Dependent Hardware iSCSI

To enable Jumbo Frames for software and dependent hardware iSCSI adapters in the vSphere Web Client, change the default value of the maximum transmission units (MTU) parameter.

You change the MTU parameter on the vSphere switch that you use for iSCSI traffic. For more information, see the *vSphere Networking* documentation.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Networking**.
- 3 Click **Virtual Switches**, and select the vSphere switch that you want to modify from the list.
- 4 Click the **Edit Settings** icon.
- 5 On the Properties page, change the MTU parameter.

This step sets the MTU for all physical NICs on that standard switch. The MTU value should be set to the largest MTU size among all NICs connected to the standard switch. ESXi supports the MTU size up to 9000 Bytes.

Enable Jumbo Frames for Independent Hardware iSCSI

To enable Jumbo Frames for independent hardware iSCSI adapters in the vSphere Web Client, change the default value of the maximum transmission units (MTU) parameter.

Use the Advanced Options settings to change the MTU parameter for the iSCSI HBA.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters**, and select the independent hardware iSCSI adapter from the list of adapters.
- 4 Under Adapter Details, click the **Advanced Options** tab and click **Edit**.
- 5 Change the value of the MTU parameter.

ESXi supports the MTU size up to 9000 Bytes.

Configuring Discovery Addresses for iSCSI Adapters

You need to set up target discovery addresses, so that the iSCSI adapter can determine which storage resource on the network is available for access.

The ESXi system supports these discovery methods:

Dynamic Discovery

Also known as SendTargets discovery. Each time the initiator contacts a specified iSCSI server, the initiator sends the SendTargets request to the server. The server responds by supplying a list of available targets to the initiator. The names and IP addresses of these targets appear on the **Static Discovery** tab. If you remove a static target added by dynamic discovery, the target might be returned to the list the next time a rescan happens, the iSCSI adapter is reset, or the host is rebooted.

NOTE With software and dependent hardware iSCSI, ESXi filters target addresses based on the IP family of the iSCSI server address specified. If the address is IPv4, IPv6 addresses that might come in the SendTargets response from the iSCSI server are filtered out. When DNS names are used to specify an iSCSI server, or when the SendTargets response from the iSCSI server has DNS names, ESXi relies on the IP family of the first resolved entry from DNS lookup.

Static Discovery

In addition to the dynamic discovery method, you can use static discovery and manually enter information for the targets. The iSCSI adapter uses a list of targets that you provide to contact and communicate with the iSCSI servers.

Set Up Dynamic or Static Discovery for iSCSI

With dynamic discovery, each time the initiator contacts a specified iSCSI storage system, it sends the SendTargets request to the system. The iSCSI system responds by supplying a list of available targets to the initiator. In addition to the dynamic discovery method, you can use static discovery and manually enter information for the targets.

When you set up static or dynamic discovery, you can only add new iSCSI targets. You cannot change any parameters of an existing target. To make changes, remove the existing target and add a new one.

Prerequisites

Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters** and select the iSCSI adapter to configure from the list.
- 4 Under Adapter Details, click the **Targets** tab.

- 5 Configure the discovery method.

Option	Description
Dynamic Discovery	<ol style="list-style-type: none"> a Click Dynamic Discovery and click Add. b Type the IP address or DNS name of the storage system and click OK. c Rescan the iSCSI adapter. <p>After establishing the SendTargets session with the iSCSI system, you host populates the Static Discovery list with all newly discovered targets.</p>
Static Discovery	<ol style="list-style-type: none"> a Click Static Discovery and click Add. b Enter the target's information and click OK c Rescan the iSCSI adapter.

Remove Dynamic or Static iSCSI Targets

Remove iSCSI servers that appear on the list of targets.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters** and select the iSCSI adapter to modify from the list.
- 4 Under Adapter Details, click the **Targets** tab.
- 5 Switch between **Dynamic Discovery** and **Static Discovery**.
- 6 Select an iSCSI server to remove and click **Remove**.
- 7 Rescan the iSCSI adapter.

If you are removing the static target that was dynamically discovered, you need to remove it from the storage system before performing the rescan. Otherwise, your host will automatically discover and add the target to the list of static targets when you rescan the adapter.

Configuring CHAP Parameters for iSCSI Adapters

Because the IP networks that the iSCSI technology uses to connect to remote targets do not protect the data they transport, you must ensure security of the connection. One of the protocols that iSCSI implements is the Challenge Handshake Authentication Protocol (CHAP), which verifies the legitimacy of initiators that access targets on the network.

CHAP uses a three-way handshake algorithm to verify the identity of your host and, if applicable, of the iSCSI target when the host and target establish a connection. The verification is based on a predefined private value, or CHAP secret, that the initiator and target share.

ESXi supports CHAP authentication at the adapter level. In this case, all targets receive the same CHAP name and secret from the iSCSI initiator. For software and dependent hardware iSCSI adapters, ESXi also supports per-target CHAP authentication, which allows you to configure different credentials for each target to achieve greater level of security.

Choosing CHAP Authentication Method

ESXi supports unidirectional CHAP for all types of iSCSI initiators, and bidirectional CHAP for software and dependent hardware iSCSI.

Before configuring CHAP, check whether CHAP is enabled at the iSCSI storage system and check the CHAP authentication method the system supports. If CHAP is enabled, enable it for your initiators, making sure that the CHAP authentication credentials match the credentials on the iSCSI storage.

ESXi supports the following CHAP authentication methods:

- Unidirectional CHAP** In unidirectional CHAP authentication, the target authenticates the initiator, but the initiator does not authenticate the target.
- Bidirectional CHAP** In bidirectional CHAP authentication, an additional level of security enables the initiator to authenticate the target. VMware supports this method for software and dependent hardware iSCSI adapters only.

For software and dependent hardware iSCSI adapters, you can set unidirectional CHAP and bidirectional CHAP for each adapter or at the target level. Independent hardware iSCSI supports CHAP only at the adapter level.

When you set the CHAP parameters, specify a security level for CHAP.

NOTE When you specify the CHAP security level, how the storage array responds depends on the array's CHAP implementation and is vendor specific. For information on CHAP authentication behavior in different initiator and target configurations, consult the array documentation.

Table 10-4. CHAP Security Level

CHAP Security Level	Description	Supported
None	The host does not use CHAP authentication. Select this option to disable authentication if it is currently enabled.	Software iSCSI Dependent hardware iSCSI Independent hardware iSCSI
Use unidirectional CHAP if required by target	The host prefers a non-CHAP connection, but can use a CHAP connection if required by the target.	Software iSCSI Dependent hardware iSCSI
Use unidirectional CHAP unless prohibited by target	The host prefers CHAP, but can use non-CHAP connections if the target does not support CHAP.	Software iSCSI Dependent hardware iSCSI Independent hardware iSCSI
Use unidirectional CHAP	The host requires successful CHAP authentication. The connection fails if CHAP negotiation fails.	Software iSCSI Dependent hardware iSCSI Independent hardware iSCSI
Use bidirectional CHAP	The host and the target support bidirectional CHAP.	Software iSCSI Dependent hardware iSCSI

Set Up CHAP for iSCSI Adapter

When you set up CHAP name and secret at the iSCSI adapter level, all targets receive the same parameters from the adapter. By default, all discovery addresses or static targets inherit CHAP parameters that you set up at the adapter level.

The CHAP name should not exceed 511 alphanumeric characters and the CHAP secret should not exceed 255 alphanumeric characters. Some adapters, for example the QLogic adapter, might have lower limits, 255 for the CHAP name and 100 for the CHAP secret.

Prerequisites

- Before setting up CHAP parameters for software or dependent hardware iSCSI, determine whether to configure unidirectional or bidirectional CHAP. Independent hardware iSCSI adapters do not support bidirectional CHAP.
- Verify CHAP parameters configured on the storage side. Parameters that you configure must match the ones on the storage side.

- Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

- 1 Display storage adapters and select the iSCSI adapter to configure.
- 2 Under Adapter Details, click the **Properties** tab and click **Edit** in the Authentication panel.
- 3 Specify authentication method.
 - **None**
 - **Use unidirectional CHAP if required by target**
 - **Use unidirectional CHAP unless prohibited by target**
 - **Use unidirectional CHAP**
 - **Use bidirectional CHAP.** To configure bidirectional CHAP, you must select this option.
- 4 Specify the outgoing CHAP name.

Make sure that the name you specify matches the name configured on the storage side.

 - To set the CHAP name to the iSCSI adapter name, select **Use initiator name**.
 - To set the CHAP name to anything other than the iSCSI initiator name, deselect **Use initiator name** and type a name in the **Name** text box.
- 5 Enter an outgoing CHAP secret to be used as part of authentication. Use the same secret that you enter on the storage side.
- 6 If configuring bidirectional CHAP, specify incoming CHAP credentials.

Make sure to use different secrets for the outgoing and incoming CHAP.
- 7 Click **OK**.
- 8 Rescan the iSCSI adapter.

If you change the CHAP parameters, they are used for new iSCSI sessions. For existing sessions, new settings are not used until you log out and log in again.

Set Up CHAP for Target

If you use software and dependent hardware iSCSI adapters, you can configure different CHAP credentials for each discovery address or static target.

The CHAP name should not exceed 511 and the CHAP secret 255 alphanumeric characters.

Prerequisites

- Before setting up CHAP parameters for software or dependent hardware iSCSI, determine whether to configure unidirectional or bidirectional CHAP.
- Verify CHAP parameters configured on the storage side. Parameters that you configure must match the ones on the storage side.
- Access storage adapters.
- Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

- 1 Select the iSCSI adapter to configure, and click the **Targets** tab under Adapter Details.
- 2 Click either **Dynamic Discovery** or **Static Discovery**.

- 3 From the list of available targets, select a target to configure and click **Authentication**.
- 4 Deselect **Inherit settings from parent** and specify authentication method.
 - **None**
 - **Use unidirectional CHAP if required by target**
 - **Use unidirectional CHAP unless prohibited by target**
 - **Use unidirectional CHAP**
 - **Use bidirectional CHAP**. To configure bidirectional CHAP, you must select this option.
- 5 Specify the outgoing CHAP name.

Make sure that the name you specify matches the name configured on the storage side.

 - To set the CHAP name to the iSCSI adapter name, select **Use initiator name**.
 - To set the CHAP name to anything other than the iSCSI initiator name, deselect **Use initiator name** and type a name in the **Name** text box.
- 6 Enter an outgoing CHAP secret to be used as part of authentication. Use the same secret that you enter on the storage side.
- 7 If configuring bi-directional CHAP, specify incoming CHAP credentials.

Make sure to use different secrets for the outgoing and incoming CHAP.
- 8 Click **OK**.
- 9 Rescan the iSCSI adapter.

If you change the CHAP parameters, they are used for new iSCSI sessions. For existing sessions, new settings are not used until you log out and login again.

Disable CHAP

You can disable CHAP if your storage system does not require it.

If you disable CHAP on a system that requires CHAP authentication, existing iSCSI sessions remain active until you reboot your host, end the session through the command line, or the storage system forces a logout. After the session ends, you can no longer connect to targets that require CHAP.

Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

- 1 Open the CHAP Credentials dialog box.
- 2 For software and dependent hardware iSCSI adapters, to disable just the mutual CHAP and leave the one-way CHAP, select **Do not use CHAP** in the Mutual CHAP area.
- 3 To disable one-way CHAP, select **Do not use CHAP** in the CHAP area.

The mutual CHAP, if set up, automatically turns to **Do not use CHAP** when you disable the one-way CHAP.
- 4 Click **OK**.

Configuring Advanced Parameters for iSCSI

You might need to configure additional parameters for your iSCSI initiators. For example, some iSCSI storage systems require ARP (Address Resolution Protocol) redirection to move iSCSI traffic dynamically from one port to another. In this case, you must activate ARP redirection on your host.

The following table lists advanced iSCSI parameters that you can configure using the vSphere Web Client. In addition, you can use the vSphere CLI commands to configure some of the advanced parameters. For information, see the *Getting Started with vSphere Command-Line Interfaces* documentation.

IMPORTANT Do not make any changes to the advanced iSCSI settings unless you are directed by VMware support or Storage Vendors.

Table 10-5. Additional Parameters for iSCSI Initiators

Advanced Parameter	Description	Configurable On
Header Digest	Increases data integrity. When header digest is enabled, the system performs a checksum over each iSCSI Protocol Data Unit's (PDU's) header part and verifies using the CRC32C algorithm.	Software iSCSI Dependent Hardware iSCSI
Data Digest	Increases data integrity. When data digest is enabled, the system performs a checksum over each PDU's data part and verifies using the CRC32C algorithm. NOTE Systems that use Intel Nehalem processors offload the iSCSI digest calculations for software iSCSI, thus reducing the impact on performance.	Software iSCSI Dependent Hardware iSCSI
Maximum Outstanding R2T	Defines the R2T (Ready to Transfer) PDUs that can be in transition before an acknowledge PDU is received.	Software iSCSI Dependent Hardware iSCSI
First Burst Length	Specifies the maximum amount of unsolicited data an iSCSI initiator can send to the target during the execution of a single SCSI command, in bytes.	Software iSCSI Dependent Hardware iSCSI
Maximum Burst Length	Maximum SCSI data payload in a Data-In or a solicited Data-Out iSCSI sequence, in bytes.	Software iSCSI Dependent Hardware iSCSI
Maximum Receive Data Segment Length	Maximum data segment length, in bytes, that can be received in an iSCSI PDU.	Software iSCSI Dependent Hardware iSCSI
Session Recovery Timeout	Specifies the amount of time, in seconds, that can lapse while a session recovery is performed. If the timeout exceeds its limit, the iSCSI initiator terminates the session.	Software iSCSI Dependent Hardware iSCSI
No-Op Interval	Specifies the time interval, in seconds, between NOP-Out requests sent from your iSCSI initiator to an iSCSI target. The NOP-Out requests serve as the ping mechanism to verify that a connection between the iSCSI initiator and the iSCSI target is active.	Software iSCSI Dependent Hardware iSCSI
No-Op Timeout	Specifies the amount of time, in seconds, that can lapse before your host receives a NOP-In message. The message is sent by the iSCSI target in response to the NOP-Out request. When the no-op timeout limit is exceeded, the initiator terminates the current session and starts a new one.	Software iSCSI Dependent Hardware iSCSI

Table 10-5. Additional Parameters for iSCSI Initiators (Continued)

Advanced Parameter	Description	Configurable On
ARP Redirect	Allows storage systems to move iSCSI traffic dynamically from one port to another. ARP is required by storage systems that do array-based failover.	Software iSCSI Dependent Hardware iSCSI Independent Hardware iSCSI
Delayed ACK	Allows systems to delay acknowledgment of received data packets.	Software iSCSI Dependent Hardware iSCSI

Configure Advanced Parameters for iSCSI

The advanced iSCSI settings control such parameters as header and data digest, ARP redirection, delayed ACK, and so on.



CAUTION Do not make any changes to the advanced iSCSI settings unless you are working with the VMware support team or otherwise have thorough information about the values to provide for the settings.

Prerequisites

Required privilege: **Host.Configuration.Storage Partition Configuration**

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters**, and select the adapter (vmhba#) to configure.
- 4 Configure advanced parameters.
 - To configure advanced parameters at the adapter level, under Adapter Details, click the **Advanced Options** tab and click **Edit**.
 - Configure advanced parameters at the target level.
 - a Click the **Targets** tab and click either **Dynamic Discovery** or **Static Discovery**.
 - b From the list of available targets, select a target to configure and click **Advanced Options**.
- 5 Enter any required values for the advanced parameters you want to modify.

iSCSI Session Management

To communicate with each other, iSCSI initiators and targets establish iSCSI sessions. You can review and manage iSCSI sessions using vSphere CLI.

By default, software iSCSI and dependent hardware iSCSI initiators start one iSCSI session between each initiator port and each target port. If your iSCSI initiator or target have more than one port, your host can have multiple sessions established. The default number of sessions for each target equals the number of ports on the iSCSI adapter times the number of target ports.

Using vSphere CLI, you can display all current sessions to analyze and debug them. To create more paths to storage systems, you can increase the default number of sessions by duplicating existing sessions between the iSCSI adapter and target ports.

You can also establish a session to a specific target port. This can be useful if your host connects to a single-port storage system that, by default, presents only one target port to your initiator, but can redirect additional sessions to a different target port. Establishing a new session between your iSCSI initiator and another target port creates an additional path to the storage system.

The following considerations apply to iSCSI session management:

- Some storage systems do not support multiple sessions from the same initiator name or endpoint. Attempts to create multiple sessions to such targets can result in unpredictable behavior of your iSCSI environment.
- Storage vendors can provide automatic session managers. Using the automatic session managers to add or delete sessions, does not guarantee lasting results and can interfere with the storage performance.

Review iSCSI Sessions

Use the vCLI command to display iSCSI sessions between an iSCSI adapter and a storage system.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ To list iSCSI sessions, run the following command:

```
esxcli --server=server_name iscsi session list
```

The command takes these options:

Option	Description
<code>-A --adapter=str</code>	The iSCSI adapter name, for example, vmhba34.
<code>-s --isid=str</code>	The iSCSI session identifier.
<code>-n --name=str</code>	The iSCSI target name, for example, iqn.X.

Add iSCSI Sessions

Use the vCLI to add an iSCSI session for a target you specify or to duplicate an existing session. By duplicating sessions, you increase the default number of sessions and create additional paths to storage systems.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ To add or duplicate an iSCSI session, run the following command:

```
esxcli --server=server_name iscsi session add
```

The command takes these options:

Option	Description
-A --adapter=<i>str</i>	The iSCSI adapter name, for example, vmhba34. This option is required.
-s --isid=<i>str</i>	The ISID of a session to duplicate. You can find it by listing all session.
-n --name=<i>str</i>	The iSCSI target name, for example, iqn.X.

What to do next

Rescan the iSCSI adapter.

Remove iSCSI Sessions

Use the vCLI command to remove an iSCSI session between an iSCSI adapter and a target.

In the procedure, **--server=*server_name*** specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ To remove a session, run the following command:

```
esxcli --server=server_name iscsi session remove
```

The command takes these options:

Option	Description
-A --adapter=<i>str</i>	The iSCSI adapter name, for example, vmhba34. This option is required.
-s --isid=<i>str</i>	The ISID of a session to remove. You can find it by listing all session.
-n --name=<i>str</i>	The iSCSI target name, for example, iqn.X.

What to do next

Rescan the iSCSI adapter.

Booting from iSCSI SAN

When you set up your host to boot from a SAN, your host's boot image is stored on one or more LUNs in the SAN storage system. When the host starts, it boots from the LUN on the SAN rather than from its local disk.

You can use boot from the SAN if you do not want to handle maintenance of local storage or have diskless hardware configurations, such as blade systems.

ESXi supports different methods of booting from the iSCSI SAN.

Table 11-1. Boot from iSCSI SAN support

Independent Hardware iSCSI	Software iSCSI and Dependent Hardware iSCSI
Configure the iSCSI HBA to boot from the SAN. For information on configuring the HBA, see “Configure Independent Hardware iSCSI Adapter for SAN Boot,” on page 108	Use the network adapter that supports the iBFT. For information, see “iBFT iSCSI Boot Overview,” on page 109.

This chapter includes the following topics:

- [“General Boot from iSCSI SAN Recommendations,”](#) on page 107
- [“Prepare the iSCSI SAN,”](#) on page 108
- [“Configure Independent Hardware iSCSI Adapter for SAN Boot,”](#) on page 108
- [“iBFT iSCSI Boot Overview,”](#) on page 109

General Boot from iSCSI SAN Recommendations

If you plan to set up and use an iSCSI LUN as the boot device for your host, you need to follow certain general guidelines.

The following guidelines apply to booting from independent hardware iSCSI and iBFT.

- Review any vendor recommendations for the hardware you use in your boot configuration.
- For installation prerequisites and requirements, review *vSphere Installation and Setup*.
- Use static IP addresses to reduce the chances of DHCP conflicts.
- Use different LUNs for VMFS datastores and boot partitions.
- Configure proper ACLs on your storage system.
 - The boot LUN should be visible only to the host that uses the LUN. No other host on the SAN should be permitted to see that boot LUN.
 - If a LUN is used for a VMFS datastore, it can be shared by multiple hosts. ACLs on the storage systems can allow you to do this.

- Configure a diagnostic partition.
 - With independent hardware iSCSI only, you can place the diagnostic partition on the boot LUN. If you configure the diagnostic partition in the boot LUN, this LUN cannot be shared across multiple hosts. If a separate LUN is used for the diagnostic partition, it can be shared by multiple hosts.
 - If you boot from SAN using iBFT, you cannot set up a diagnostic partition on a SAN LUN. To collect your host's diagnostic information, use the vSphere ESXi Dump Collector on a remote server. For information about the ESXi Dump Collector, see *vSphere Installation and Setup* and *vSphere Networking*.

Prepare the iSCSI SAN

Before you configure your host to boot from an iSCSI LUN, prepare and configure your storage area network.



CAUTION If you use scripted installation to install ESXi when booting from a SAN, you must take special steps to avoid unintended data loss.

Procedure

- 1 Connect network cables, referring to any cabling guide that applies to your setup.
- 2 Ensure IP connectivity between your storage system and server.

This includes proper configuration of any routers or switches on your storage network. Storage systems must be able to ping the iSCSI adapters in your hosts.
- 3 Configure the storage system.
 - a Create a volume (or LUN) on the storage system for your host to boot from.
 - b Configure the storage system so that your host has access to the assigned LUN.

This could involve updating ACLs with the IP addresses, iSCSI names, and the CHAP authentication parameter you use on your host. On some storage systems, in addition to providing access information for the ESXi host, you must also explicitly associate the assigned LUN with the host.
 - c Ensure that the LUN is presented to the host correctly.
 - d Ensure that no other system has access to the configured LUN.
 - e Record the iSCSI name and IP addresses of the targets assigned to the host.

You must have this information to configure your iSCSI adapters.

Configure Independent Hardware iSCSI Adapter for SAN Boot

If your ESXi host uses an independent hardware iSCSI adapter, such as QLogic HBA, you need to configure the adapter to boot from the SAN.

This procedure discusses how to enable the QLogic iSCSI HBA to boot from the SAN. For more information and more up-to-date details about QLogic adapter configuration settings, see the QLogic web site.

Prerequisites

Because you first need to boot from the VMware installation media, set up your host to boot from CD/DVD-ROM. To achieve this, change the system boot sequence in your system BIOS setup.

Procedure

- 1 Insert the installation CD/DVD in the CD/DVD-ROM drive and reboot the host.

- 2 Use the BIOS to set the host to boot from the CD/DVD-ROM drive first.
- 3 During server POST, press Ctrl+q to enter the QLogic iSCSI HBA configuration menu.
- 4 Select the I/O port to configure.
By default, the Adapter Boot mode is set to Disable.
- 5 Configure the HBA.
 - a From the **Fast!UTIL Options** menu, select **Configuration Settings > Host Adapter Settings**.
 - b Configure the following settings for your host adapter: initiator IP address, subnet mask, gateway, initiator iSCSI name, and CHAP (if required).
- 6 Configure iSCSI settings.
See [“Configure iSCSI Boot Settings,”](#) on page 109.
- 7 Save your changes and restart the system.

Configure iSCSI Boot Settings

When setting up your ESXi host to boot from iSCSI, you need to configure iSCSI boot settings.

Procedure

- 1 From the **Fast!UTIL Options** menu, select **Configuration Settings > iSCSI Boot Settings**.
- 2 Before you can set SendTargets, set Adapter Boot mode to **Manual**.
- 3 Select **Primary Boot Device Settings**.
 - a Enter the discovery **Target IP** and **Target Port**.
 - b You can leave the **Boot LUN** and **iSCSI Name** fields blank if only one iSCSI target and one LUN are at the specified address to boot from. Otherwise, you must specify these fields to ensure that you do not boot from a volume for some other system. After the target storage system is reached, these fields will be populated after a rescan.
 - c Save changes.
- 4 From the **iSCSI Boot Settings** menu, select the primary boot device. An auto rescan of the HBA is made to find new target LUNS.
- 5 Select the iSCSI target.

NOTE If more than one LUN exists within the target, you can choose a specific LUN ID by pressing **Enter** after you locate the iSCSI device.

- 6 Return to the **Primary Boot Device Setting** menu. After the rescan, the **Boot LUN** and **iSCSI Name** fields are populated. Change the value of **Boot LUN** to the desired LUN ID.

iBFT iSCSI Boot Overview

ESXi hosts can boot from an iSCSI SAN using the software or dependent hardware iSCSI adapters and network adapters.

To deploy ESXi and boot from the iSCSI SAN, the host must have an iSCSI boot capable network adapter that supports the iSCSI Boot Firmware Table (iBFT) format. The iBFT is a method of communicating parameters about the iSCSI boot device to an operating system.

Before installing ESXi and booting from the iSCSI SAN, configure the networking and iSCSI boot parameters on the network adapter and enable the adapter for the iSCSI boot. Because configuring the network adapter is vendor specific, review your vendor documentation for instructions.

When you first boot from iSCSI, the iSCSI boot firmware on your system connects to an iSCSI target. If login is successful, the firmware saves the networking and iSCSI boot parameters in the iBFT and stores the table in the system's memory. The system uses this table to configure its own iSCSI connection and networking and to start up.

The following list describes the iBFT iSCSI boot sequence.

- 1 When restarted, the system BIOS detects the iSCSI boot firmware on the network adapter.
- 2 The iSCSI boot firmware uses the preconfigured boot parameters to connect with the specified iSCSI target.
- 3 If the connection to the iSCSI target is successful, the iSCSI boot firmware writes the networking and iSCSI boot parameters in to the iBFT and stores the table in the system memory.

NOTE The system uses this table to configure its own iSCSI connection and networking and to start up.

- 4 The BIOS boots the boot device.
- 5 The VMkernel starts loading and takes over the boot operation.
- 6 Using the boot parameters from the iBFT, the VMkernel connects to the iSCSI target.
- 7 After the iSCSI connection is established, the system boots.

iBFT iSCSI Boot Considerations

When you boot the ESXi host from iSCSI using iBFT-enabled network adapters, certain considerations apply.

- Update your NIC's boot code and iBFT firmware using vendor supplied tools before trying to install and boot VMware ESXi. Consult vendor documentation and VMware HCL for supported boot code and iBFT firmware versions for VMware ESXi iBFT boot.
- The iBFT iSCSI boot does not support failover for the iBFT-enabled network adapters.
- After you set up your host to boot from iBFT iSCSI, the following restrictions apply:
 - You cannot disable the software iSCSI adapter. If the iBFT configuration is present in the BIOS, the host re-enables the software iSCSI adapter during each reboot.

NOTE If you do not use the iBFT-enabled network adapter for the iSCSI boot and do not want the software iSCSI adapter to be always enabled, remove the iBFT configuration from the network adapter.

- You cannot remove the iBFT iSCSI boot target using the vSphere Web Client. The target appears on the list of adapter static targets.

Configuring iBFT Boot from SAN

You can boot from the iSCSI SAN using the software iSCSI adapter or a dependent hardware iSCSI adapter and a network adapter. The network adapter must support iBFT.

When you set up your host to boot with iBFT, you perform a number of tasks.

- 1 [Configure iSCSI Boot Parameters](#) on page 111
To begin an iSCSI boot process, a network adapter on your host must have a specially configured iSCSI boot firmware. When you configure the firmware, you specify the networking and iSCSI parameters and enable the adapter for the iSCSI boot.
- 2 [Change Boot Sequence in BIOS](#) on page 111
When setting up your host to boot from iBFT iSCSI, change the boot sequence to force your host to boot in an appropriate order.

3 [Install ESXi to iSCSI Target](#) on page 111

When setting up your host to boot from iBFT iSCSI, install the ESXi image to the target LUN.

4 [Boot ESXi from iSCSI Target](#) on page 112

After preparing the host for an iBFT iSCSI boot and copying the ESXi image to the iSCSI target, perform the actual boot.

Configure iSCSI Boot Parameters

To begin an iSCSI boot process, a network adapter on your host must have a specially configured iSCSI boot firmware. When you configure the firmware, you specify the networking and iSCSI parameters and enable the adapter for the iSCSI boot.

Configuration on the network adapter can be dynamic or static. If you use the dynamic configuration, you indicate that all target and initiator boot parameters are acquired using DHCP. For the static configuration, you manually enter data that includes your host's IP address and initiator IQN, and the target parameters.

Procedure

- ◆ On the network adapter that you use for the boot from iSCSI, specify networking and iSCSI parameters. Because configuring the network adapter is vendor specific, review your vendor documentation for instructions.

Change Boot Sequence in BIOS

When setting up your host to boot from iBFT iSCSI, change the boot sequence to force your host to boot in an appropriate order.

Change the BIOS boot sequence to the following sequence:

- iSCSI
- DVD-ROM

Because changing the boot sequence in the BIOS is vendor specific, refer to vendor documentation for instructions. The following sample procedure explains how to change the boot sequence on a Dell host with a Broadcom network adapter.

Procedure

- 1 Turn on the host.
- 2 During Power-On Self-Test (POST), press F2 to enter the BIOS Setup.
- 3 In the BIOS Setup, select **Boot Sequence** and press Enter.
- 4 In the Boot Sequence menu, arrange the bootable items so that iSCSI precedes the DVD-ROM.
- 5 Press Esc to exit the Boot Sequence menu.
- 6 Press Esc to exit the BIOS Setup.
- 7 Select **Save Changes** and click **Exit** to exit the BIOS Setup menu.

Install ESXi to iSCSI Target

When setting up your host to boot from iBFT iSCSI, install the ESXi image to the target LUN.

Prerequisites

- Configure iSCSI boot firmware on your boot NIC to point to the target LUN that you want to use as the boot LUN.
- Change the boot sequence in the BIOS so that iSCSI precedes the DVD-ROM.

- If you use Broadcom adapters, set **Boot to iSCSI target** to **Disabled**.

Procedure

- 1 Insert the installation media in the CD/DVD-ROM drive and restart the host.
- 2 When the installer starts, follow the typical installation procedure.
- 3 When prompted, select the iSCSI LUN as the installation target.
The installer copies the ESXi boot image to the iSCSI LUN.
- 4 After the system restarts, remove the installation DVD.

Boot ESXi from iSCSI Target

After preparing the host for an iBFT iSCSI boot and copying the ESXi image to the iSCSI target, perform the actual boot.

Prerequisites

- Configure the iSCSI boot firmware on your boot NIC to point to the boot LUN.
- Change the boot sequence in the BIOS so that iSCSI precedes the boot device.
- If you use Broadcom adapters, set **Boot to iSCSI target** to **Enabled**

Procedure

- 1 Restart the host.
The host boots from the iSCSI LUN using iBFT data. During the first boot, the iSCSI initialization script sets up default networking. The network setup is persistent after subsequent reboots.
- 2 (Optional) Adjust networking configuration using the vSphere Web Client.

Networking Best Practices

To boot the ESXi host from iSCSI using iBFT, you must properly configure networking.

To achieve greater security and better performance, have redundant network adapters on the host.

How you set up all the network adapters depends on whether your environment uses shared or isolated networks for the iSCSI traffic and host management traffic.

Shared iSCSI and Management Networks

Configure the networking and iSCSI parameters on the first network adapter on the host. After the host boots, you can add secondary network adapters to the default port group.

Isolated iSCSI and Management Networks

When you configure isolated iSCSI and management networks, follow these guidelines to avoid bandwidth problems.

- Your isolated networks must be on different subnets.
- If you use VLANs to isolate the networks, they must have different subnets to ensure that routing tables are properly set up.
- VMware recommends that you configure the iSCSI adapter and target to be on the same subnet. If you set up the iSCSI adapter and target on different subnets, the following restrictions apply:
 - The default VMkernel gateway must be able to route both the management and iSCSI traffic.
 - After you boot your host, you can use the iBFT-enabled network adapter only for iBFT. You cannot use the adapter for other iSCSI traffic.

- Use the first physical network adapter for the management network.
- Use the second physical network adapter for the iSCSI network. Make sure to configure the iBFT.
- After the host boots, you can add secondary network adapters to both the management and iSCSI networks.

Change iBFT iSCSI Boot Settings

If settings, such as the IQN name, IP address, and so on, change on the iSCSI storage or your host, update the iBFT. This task assumes that the boot LUN and the data stored on the LUN remain intact.

Procedure

- 1 Shut down the ESXi host.
- 2 Change iSCSI storage settings.
- 3 Update the iBFT on the host with the new settings.
- 4 Restart the host.

The host boots using the new information stored in the iBFT.

Troubleshooting iBFT iSCSI Boot

The topics in this section help you to identify and solve problems you might encounter when using iBFT iSCSI boot.

Loss of System's Gateway Causes Loss of Network Connectivity

You lose network connectivity when you delete a port group associated with the iBFT network adapter.

Problem

A loss of network connectivity occurs after you delete a port group.

Cause

When you specify a gateway in the iBFT-enabled network adapter during ESXi installation, this gateway becomes the system's default gateway. If you delete the port group associated with the network adapter, the system's default gateway is lost. This action causes the loss of network connectivity.

Solution

Do not set an iBFT gateway unless it is required. If the gateway is required, after installation, manually set the system's default gateway to the one that the management network uses.

Changing iSCSI Boot Parameters Causes ESXi to Boot in Stateless Mode

Changing iSCSI boot parameters on the network adapter after the first boot does not update the iSCSI and networking configuration on the ESXi host.

Problem

If you change the iSCSI boot parameters on the network adapter after the first ESXi boot from iSCSI, the host will boot in a stateless mode.

Cause

The firmware uses the updated boot configuration and is able to connect to the iSCSI target and load the ESXi image. However, when loaded, the system does not pick up the new parameters, but continues to use persistent networking and iSCSI parameters from the previous boot. As a result, the host cannot connect to the target and boots in the stateless mode.

Solution

- 1 Use the vSphere Web Client to connect to the ESXi host.
- 2 Re-configure the iSCSI and networking on the host to match the iBFT parameters.
- 3 Perform a rescan.

Best Practices for iSCSI Storage

When using ESXi with the iSCSI SAN, follow best practices that VMware offers to avoid problems.

Check with your storage representative if your storage system supports Storage API - Array Integration hardware acceleration features. If it does, refer to your vendor documentation for information on how to enable hardware acceleration support on the storage system side. For more information, see [Chapter 22, “Storage Hardware Acceleration,”](#) on page 243.

This chapter includes the following topics:

- [“Preventing iSCSI SAN Problems,”](#) on page 115
- [“Optimizing iSCSI SAN Storage Performance,”](#) on page 116
- [“Checking Ethernet Switch Statistics,”](#) on page 119

Preventing iSCSI SAN Problems

When using ESXi in conjunction with a SAN, you must follow specific guidelines to avoid SAN problems.

You should observe these tips for avoiding problems with your SAN configuration:

- Place only one VMFS datastore on each LUN. Multiple VMFS datastores on one LUN is not recommended.
- Do not change the path policy the system sets for you unless you understand the implications of making such a change.
- Document everything. Include information about configuration, access control, storage, switch, server and iSCSI HBA configuration, software and firmware versions, and storage cable plan.
- Plan for failure:
 - Make several copies of your topology maps. For each element, consider what happens to your SAN if the element fails.
 - Cross off different links, switches, HBAs and other elements to ensure you did not miss a critical failure point in your design.
- Ensure that the iSCSI HBAs are installed in the correct slots in the ESXi host, based on slot and bus speed. Balance PCI bus load among the available busses in the server.
- Become familiar with the various monitor points in your storage network, at all visibility points, including ESXi performance charts, Ethernet switch statistics, and storage performance statistics.
- Be cautious when changing IDs of the LUNs that have VMFS datastores being used by your host. If you change the ID, virtual machines running on the VMFS datastore will fail.

If there are no running virtual machines on the VMFS datastore, after you change the ID of the LUN, you must use `rescan` to reset the ID on your host. For information on using `rescan`, see [“Storage Refresh and Rescan Operations,”](#) on page 124.

- If you need to change the default iSCSI name of your iSCSI adapter, make sure the name you enter is worldwide unique and properly formatted. To avoid storage access problems, never assign the same iSCSI name to different adapters, even on different hosts.

Optimizing iSCSI SAN Storage Performance

Several factors contribute to optimizing a typical SAN environment.

If the network environment is properly configured, the iSCSI components provide adequate throughput and low enough latency for iSCSI initiators and targets. If the network is congested and links, switches or routers are saturated, iSCSI performance suffers and might not be adequate for ESXi environments.

Storage System Performance

Storage system performance is one of the major factors contributing to the performance of the entire iSCSI environment.

If issues occur with storage system performance, consult your storage system vendor’s documentation for any relevant information.

When you assign LUNs, remember that you can access each shared LUN through a number of hosts, and that a number of virtual machines can run on each host. One LUN used by the ESXi host can service I/O from many different applications running on different operating systems. Because of this diverse workload, the RAID group that contains the ESXi LUNs should not include LUNs that other hosts use that are not running ESXi for I/O intensive applications.

Enable read caching and write caching.

Load balancing is the process of spreading server I/O requests across all available SPs and their associated host server paths. The goal is to optimize performance in terms of throughput (I/O per second, megabytes per second, or response times).

SAN storage systems require continual redesign and tuning to ensure that I/O is load balanced across all storage system paths. To meet this requirement, distribute the paths to the LUNs among all the SPs to provide optimal load balancing. Close monitoring indicates when it is necessary to manually rebalance the LUN distribution.

Tuning statically balanced storage systems is a matter of monitoring the specific performance statistics (such as I/O operations per second, blocks per second, and response time) and distributing the LUN workload to spread the workload across all the SPs.

Server Performance with iSCSI

You must consider several factors to ensure optimal server performance.

Each server application must have access to its designated storage with the following conditions:

- High I/O rate (number of I/O operations per second)
- High throughput (megabytes per second)
- Minimal latency (response times)

Because each application has different requirements, you can meet these goals by choosing an appropriate RAID group on the storage system. To achieve performance goals, perform the following tasks:

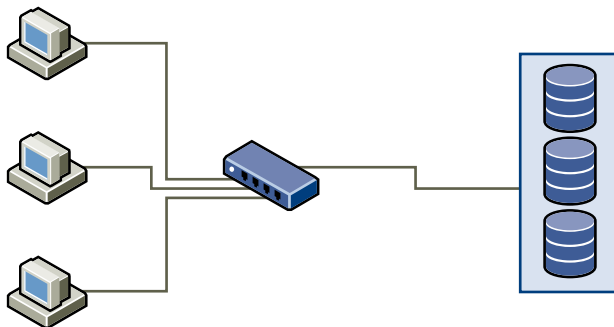
- Place each LUN on a RAID group that provides the necessary performance levels. Pay attention to the activities and resource utilization of other LUNs in the assigned RAID group. A high-performance RAID group that has too many applications doing I/O to it might not meet performance goals required by an application running on the ESXi host.
- Provide each server with a sufficient number of network adapters or iSCSI hardware adapters to allow maximum throughput for all the applications hosted on the server for the peak period. I/O spread across multiple ports provides higher throughput and less latency for each application.
- To provide redundancy for software iSCSI, make sure the initiator is connected to all network adapters used for iSCSI connectivity.
- When allocating LUNs or RAID groups for ESXi systems, multiple operating systems use and share that resource. As a result, the performance required from each LUN in the storage subsystem can be much higher if you are working with ESXi systems than if you are using physical machines. For example, if you expect to run four I/O intensive applications, allocate four times the performance capacity for the ESXi LUNs.
- When using multiple ESXi systems in conjunction with vCenter Server, the performance needed from the storage subsystem increases correspondingly.
- The number of outstanding I/Os needed by applications running on an ESXi system should match the number of I/Os the SAN can handle.

Network Performance

A typical SAN consists of a collection of computers connected to a collection of storage systems through a network of switches. Several computers often access the same storage.

Single Ethernet Link Connection to Storage shows several computer systems connected to a storage system through an Ethernet switch. In this configuration, each system is connected through a single Ethernet link to the switch, which is also connected to the storage system through a single Ethernet link. In most configurations, with modern switches and typical traffic, this is not a problem.

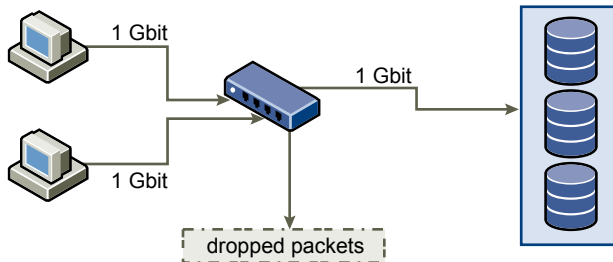
Figure 12-1. Single Ethernet Link Connection to Storage



When systems read data from storage, the maximum response from the storage is to send enough data to fill the link between the storage systems and the Ethernet switch. It is unlikely that any single system or virtual machine gets full use of the network speed, but this situation can be expected when many systems share one storage device.

When writing data to storage, multiple systems or virtual machines might attempt to fill their links. As Dropped Packets shows, when this happens, the switch between the systems and the storage system has to drop data. This happens because, while it has a single connection to the storage device, it has more traffic to send to the storage system than a single link can carry. In this case, the switch drops network packets because the amount of data it can transmit is limited by the speed of the link between it and the storage system.

Figure 12-2. Dropped Packets



Recovering from dropped network packets results in large performance degradation. In addition to time spent determining that data was dropped, the retransmission uses network bandwidth that could otherwise be used for current transactions.

iSCSI traffic is carried on the network by the Transmission Control Protocol (TCP). TCP is a reliable transmission protocol that ensures that dropped packets are retried and eventually reach their destination. TCP is designed to recover from dropped packets and retransmits them quickly and seamlessly. However, when the switch discards packets with any regularity, network throughput suffers significantly. The network becomes congested with requests to resend data and with the resent packets, and less data is actually transferred than in a network without congestion.

Most Ethernet switches can buffer, or store, data and give every device attempting to send data an equal chance to get to the destination. This ability to buffer some transmissions, combined with many systems limiting the number of outstanding commands, allows small bursts from several systems to be sent to a storage system in turn.

If the transactions are large and multiple servers are trying to send data through a single switch port, a switch's ability to buffer one request while another is transmitted can be exceeded. In this case, the switch drops the data it cannot send, and the storage system must request retransmission of the dropped packet. For example, if an Ethernet switch can buffer 32KB on an input port, but the server connected to it thinks it can send 256KB to the storage device, some of the data is dropped.

Most managed switches provide information on dropped packets, similar to the following:

```
*: interface is up
IHQ: pkts in input hold queue      IQD: pkts dropped from input queue
OHQ: pkts in output hold queue    OQD: pkts dropped from output queue
RXBS: rx rate (bits/sec)          RXPS: rx rate (pkts/sec)
TXBS: tx rate (bits/sec)          TXPS: tx rate (pkts/sec)
TRTL: throttle count
```

Table 12-1. Sample Switch Information

Interface	IHQ	IQD	OHQ	OQD	RXBS	RXPS	TXBS	TXPS	TRTL
* GigabitEt hernet0/1	3	9922	0	0	47630300 0	62273	47784000 0	63677	0

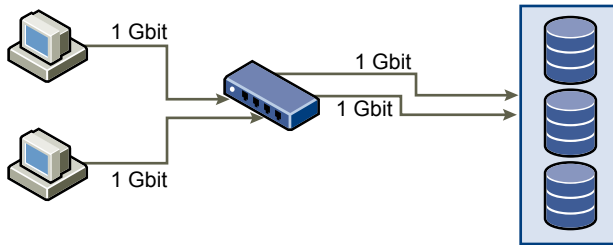
In this example from a Cisco switch, the bandwidth used is 476303000 bits/second, which is less than half of wire speed. In spite of this, the port is buffering incoming packets and has dropped quite a few packets. The final line of this interface summary indicates that this port has already dropped almost 10,000 inbound packets in the IQD column.

Configuration changes to avoid this problem involve making sure several input Ethernet links are not funneled into one output link, resulting in an oversubscribed link. When a number of links transmitting near capacity are switched to a smaller number of links, oversubscription is a possibility.

Generally, applications or systems that write a lot of data to storage, such as data acquisition or transaction logging systems, should not share Ethernet links to a storage device. These types of applications perform best with multiple connections to storage devices.

Multiple Connections from Switch to Storage shows multiple connections from the switch to the storage.

Figure 12-3. Multiple Connections from Switch to Storage



Using VLANs or VPNs does not provide a suitable solution to the problem of link oversubscription in shared configurations. VLANs and other virtual partitioning of a network provide a way of logically designing a network, but do not change the physical capabilities of links and trunks between switches. When storage traffic and other network traffic end up sharing physical connections, as they would with a VPN, the possibility for oversubscription and lost packets exists. The same is true of VLANs that share interswitch trunks. Performance design for a SANs must take into account the physical limitations of the network, not logical allocations.

Checking Ethernet Switch Statistics

Many Ethernet switches provide different methods for monitoring switch health.

Switches that have ports operating near maximum throughput much of the time do not provide optimum performance. If you have ports in your iSCSI SAN running near the maximum, reduce the load. If the port is connected to an ESXi system or iSCSI storage, you can reduce the load by using manual load balancing.

If the port is connected between multiple switches or routers, consider installing additional links between these components to handle more load. Ethernet switches also commonly provide information about transmission errors, queued packets, and dropped Ethernet packets. If the switch regularly reports any of these conditions on ports being used for iSCSI traffic, performance of the iSCSI SAN will be poor.

Managing Storage Devices

Manage local and networked storage device that your ESXi host has access to.

This chapter includes the following topics:

- [“Storage Device Characteristics,”](#) on page 121
- [“Understanding Storage Device Naming,”](#) on page 123
- [“Storage Refresh and Rescan Operations,”](#) on page 124
- [“Identifying Device Connectivity Problems,”](#) on page 126
- [“Edit Configuration File Parameters,”](#) on page 131
- [“Enable or Disable the Locator LED on Storage Devices,”](#) on page 132

Storage Device Characteristics

You can display all storage devices or LUNs available to the host, including all local and networked devices. If you use third-party multipathing plug-ins, the storage devices available through the plug-ins also appear on the list.

For each storage adapter, you can display a separate list of storage devices available for this adapter.

Generally, when you review storage devices, you see the following information.

Table 13-1. Storage Device Information

Storage Device Information	Description
Name	Also called Display Name. It is a name that the ESXi host assigns to the device based on the storage type and manufacturer. You can change this name to a name of your choice.
Identifier	A universally unique identifier that is intrinsic to the device.
Operational State	Indicates whether the device is mounted or unmounted. For details, see “Detach Storage Devices,” on page 128.
LUN	Logical Unit Number (LUN) within the SCSI target. The LUN number is provided by the storage system. If a target has only one LUN, the LUN number is always zero (0).
Type	Type of device, for example, disk or CD-ROM.
Drive Type	Information about whether the device is a flash drive or a regular HDD drive. For information about flash drives, see Chapter 14, “Working with Flash Devices,” on page 133.
Transport	Transportation protocol your host uses to access the device. The protocol depends on the type of storage being used. See “Types of Physical Storage,” on page 14.
Capacity	Total capacity of the storage device.

Table 13-1. Storage Device Information (Continued)

Storage Device Information	Description
Owner	The plug-in, such as the NMP or a third-party plug-in, that the host uses to manage paths to the storage device. For details, see “Managing Multiple Paths,” on page 186.
Hardware Acceleration	Information about whether the storage device assists the host with virtual machine management operations. The status can be Supported, Not Supported, or Unknown. For details, see Chapter 22, “Storage Hardware Acceleration,” on page 243.
Location	A path to the storage device in the /vmfs/devices/ directory.
Partition Format	A partition scheme used by the storage device. It could be of a master boot record (MBR) or GUID partition table (GPT) format. The GPT devices can support datastores greater than 2TB. For more information, see “VMFS Datastores and Storage Disk Formats,” on page 147.
Partitions	Primary and logical partitions, including a VMFS datastore, if configured.
Multipathing Policies (VMFS datastores)	Path Selection Policy and Storage Array Type Policy the host uses to manage paths to storage. For more information, see Chapter 17, “Understanding Multipathing and Failover,” on page 181.
Paths (VMFS datastores)	Paths used to access storage and their status.

Display Storage Devices for a Host

Display all storage devices available to a host. If you use any third-party multipathing plug-ins, the storage devices available through the plug-ins also appear on the list.

The Storage Devices view allows you to list the hosts' storage devices, analyze their information, and modify properties.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Devices**.

All storage devices available to the host are listed under Storage Devices.

- 4 To view details for a specific device, select the device from the list.
- 5 Use tabs under Device Details to access additional information and modify properties for the selected device.

Tab	Description
Properties	View device properties and characteristics. View and modify multipathing policies for the device.
Paths	Display paths available for the device. Disable or enable a selected path.

Display Storage Devices for an Adapter

Display a list of storage devices accessible through a specific storage adapter on the host.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.

- 3 Click Storage Adapters.

All storage adapters installed on the host are listed under Storage Adapters.

- 4 Select the adapter from the list and click the **Devices** tab.

Storage devices that the host can access through the adapter are displayed.

Understanding Storage Device Naming

Each storage device, or LUN, is identified by several names.

Device Identifiers

Depending on the type of storage, the ESXi host uses different algorithms and conventions to generate an identifier for each storage device.

SCSI INQUIRY identifiers.

The host uses the SCSI INQUIRY command to query a storage device and uses the resulting data, in particular the Page 83 information, to generate a unique identifier. Device identifiers that are based on Page 83 are unique across all hosts, persistent, and have one of the following formats:

- *naa.number*
- *t10.number*
- *eui.number*

These formats follow the T10 committee standards. See the SCSI-3 documentation on the T10 committee Web site.

Path-based identifier.

When the device does not provide the Page 83 information, the host generates an *mpx.path* name, where *path* represents the first path to the device, for example, *mpx.vmhba1:C0:T1:L3*. This identifier can be used in the same way as the SCSI INQUIRY identifies.

The *mpx.* identifier is created for local devices on the assumption that their path names are unique. However, this identifier is neither unique nor persistent and could change after every boot.

Typically, the path to the device has the following format:

vmhbaAdapter:CChannel:TTarget:LLUN

- *vmhbaAdapter* is the name of the storage adapter. The name refers to the physical adapter on the host, not to the SCSI controller used by the virtual machines.
- *CChannel* is the storage channel number.
Software iSCSI adapters and dependent hardware adapters use the channel number to show multiple paths to the same target.
- *TTarget* is the target number. Target numbering is determined by the host and might change if the mappings of targets visible to the host change. Targets that are shared by different hosts might not have the same target number.
- *LLUN* is the LUN number that shows the position of the LUN within the target. The LUN number is provided by the storage system. If a target has only one LUN, the LUN number is always zero (0).

For example, *vmhba1:C0:T3:L1* represents LUN1 on target 3 accessed through the storage adapter *vmhba1* and channel 0.

Legacy Identifier

In addition to the SCSI INQUIRY or mp.x. identifiers, for each device, ESXi generates an alternative legacy name. The identifier has the following format:

vml.number

The legacy identifier includes a series of digits that are unique to the device and can be derived in part from the Page 83 information, if it is available. For nonlocal devices that do not support Page 83 information, the vml. name is used as the only available unique identifier.

Example: Displaying Device Names in the vSphere CLI

You can use the `esxcli --server=server_name storage core device list` command to display all device names in the vSphere CLI. The output is similar to the following example:

```
# esxcli --server=server_name storage core device list
naa.number
  Display Name: DGC Fibre Channel Disk(naa.number)
  ...
  Other UUIDs:vml.number
```

Rename Storage Devices

You can change the display name of a storage device. The display name is assigned by the ESXi host based on the storage type and manufacturer.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Devices**.
- 4 Select the device to rename and click **Rename**.
- 5 Change the device name to a friendly name.

Storage Refresh and Rescan Operations

The refresh operation for datastores, storage devices, and storage adapters updates the lists and storage information displayed in the vSphere Web Client. For example, it updates such information as the datastore capacity. When you perform storage management tasks or make changes in the SAN configuration, you might need to rescan your storage.

When you perform VMFS datastore management operations, such as creating a VMFS datastore or RDM, adding an extent, and increasing or deleting a VMFS datastore, your host or the vCenter Server automatically rescans and updates your storage. You can disable the automatic rescan feature by turning off the Host Rescan Filter. See [“Turn off Storage Filters,”](#) on page 172.

In certain cases, you need to perform a manual rescan. You can rescan all storage available to your host, or, if you are using the vCenter Server, to all hosts in a folder, cluster, and data center.

If the changes you make are isolated to storage connected through a specific adapter, perform a rescan for this adapter.

Perform the manual rescan each time you make one of the following changes.

- Zone a new disk array on a SAN.
- Create new LUNs on a SAN.

- Change the path masking on a host.
- Reconnect a cable.
- Change CHAP settings (iSCSI only).
- Add or remove discovery or static addresses (iSCSI only).
- Add a single host to the vCenter Server after you have edited or removed from the vCenter Server a datastore shared by the vCenter Server hosts and the single host.

IMPORTANT If you rescan when a path is unavailable, the host removes the path from the list of paths to the device. The path reappears on the list as soon as it becomes available and starts working again.

Perform Storage Rescan

When you make changes in your SAN configuration, you might need to rescan your storage. You can rescan all storage available to your host, cluster, or data center. If the changes you make are isolated to storage accessed through a specific host, perform rescan for only this host.

Procedure

- 1 In the vSphere Web Client object navigator, browse to a host, a cluster, a data center, or a folder that contains hosts.
- 2 Select **Storage > Rescan Storage** from the right-click menu.
- 3 Specify extent of rescan.

Option	Description
Scan for New Storage Devices	Rescan all adapters to discover new storage devices. If new devices are discovered, they appear in the device list.
Scan for New VMFS Volumes	Rescan all storage devices to discover new datastores that have been added since the last scan. Any new datastores appear in the datastore list.

Perform Adapter Rescan

When you make changes in your SAN configuration and these changes are isolated to storage accessed through a specific adapter, perform rescan for only this adapter.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Adapters**, and select the adapter to rescan from the list.
- 4 Click the **Rescan Adapter** icon.

Change the Number of Scanned Storage Devices

While an ESXi host is limited to accessing 256 SCSI storage devices, the range of LUN IDs can be from 0 to 1023. ESXi ignores LUN IDs 1024 or greater. This limit is controlled by `Disk.MaxLUN`, which has a default value of 1024.

The value of `Disk.MaxLUN` also determines how many LUNs the SCSI scan code attempts to discover using individual INQUIRY commands if the SCSI target does not support direct discovery using `REPORT_LUNS`.

You can modify the `Disk.MaxLUN` parameter depending on your needs. For example, if your environment has a smaller number of storage devices with LUN IDs from 0 through 100, you can set the value to 101 to improve device discovery speed on targets that do not support `REPORT_LUNS`. Lowering the value can shorten the rescan time and boot time. However, the time to rescan storage devices might depend on other factors, including the type of storage system and the load on the storage system.

In other cases, you might need to increase the value if your environment uses LUN IDs that are greater than 1023.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Settings**.
- 3 Under System, click **Advanced System Settings**.
- 4 Under Advanced System Settings, select **Disk.MaxLUN** and click the **Edit** icon.
- 5 Change the existing value to the value of your choice, and click **OK**.

The value you enter specifies the LUN ID that is after the last one you want to discover.

For example, to discover LUN IDs from 0 through 100, set **Disk.MaxLUN** to 101.

Identifying Device Connectivity Problems

When your ESXi host experiences a problem while connecting to a storage device, the host treats the problem as permanent or temporary depending on certain factors.

Storage connectivity problems are caused by a variety of reasons. Although ESXi cannot always determine the reason for a storage device or its paths being unavailable, the host differentiates between a permanent device loss (PDL) state of the device and a transient all paths down (APD) state of storage.

Permanent Device Loss (PDL)

A condition that occurs when a storage device permanently fails or is administratively removed or excluded. It is not expected to become available. When the device becomes permanently unavailable, ESXi receives appropriate sense codes or a login rejection from storage arrays, and is able to recognize that the device is permanently lost.

All Paths Down (APD)

A condition that occurs when a storage device becomes inaccessible to the host and no paths to the device are available. ESXi treats this as a transient condition because typically the problems with the device are temporary and the device is expected to become available again.

Detecting PDL Conditions

A storage device is considered to be in the permanent device loss (PDL) state when it becomes permanently unavailable to your ESXi host.

Typically, the PDL condition occurs when a device is unintentionally removed, or its unique ID changes, or when the device experiences an unrecoverable hardware error.

When the storage array determines that the device is permanently unavailable, it sends SCSI sense codes to the ESXi host. The sense codes allow your host to recognize that the device has failed and register the state of the device as PDL. The sense codes must be received on all paths to the device for the device to be considered permanently lost.

After registering the PDL state of the device, the host stops attempts to reestablish connectivity or to issue commands to the device to avoid becoming blocked or unresponsive.

The vSphere Web Client displays the following information for the device:

- The operational state of the device changes to *Lost Communication*.
- All paths are shown as *Dead*.
- Datastores on the device are grayed out.

The host automatically removes the PDL device and all paths to the device if no open connections to the device exist, or after the last connection closes. You can disable the automatic removal of paths by setting the advanced host parameter `Disk.AutoremoveOnPDL` to 0. See [“Set Advanced Host Attributes,”](#) on page 179.

If the device returns from the PDL condition, the host can discover it, but treats it as a new device. Data consistency for virtual machines on the recovered device is not guaranteed.

NOTE The host cannot detect PDL conditions and continues to treat the device connectivity problems as APD when a storage device permanently fails in a way that does not return appropriate SCSI sense codes or iSCSI login rejection.

Permanent Device Loss and SCSI Sense Codes

The following VMkernel log example of a SCSI sense code indicates that the device is in the PDL state.

```
H:0x0 D:0x2 P:0x0 Valid sense data: 0x5 0x25 0x0 or Logical Unit Not Supported
```

For information about SCSI sense codes, see *Troubleshooting Storage in vSphere Troubleshooting*.

Permanent Device Loss and iSCSI

In the case of iSCSI arrays with a single LUN per target, PDL is detected through iSCSI login failure. An iSCSI storage array rejects your host's attempts to start an iSCSI session with a reason `Target Unavailable`. As with the sense codes, this response must be received on all paths for the device to be considered permanently lost.

Permanent Device Loss and Virtual Machines

After registering the PDL state of the device, the host terminates all I/O from virtual machines. vSphere HA can detect PDL and restart failed virtual machines. For more information, see [“Device Connectivity Problems and High Availability,”](#) on page 131.

Performing Planned Storage Device Removal

When a storage device is malfunctioning, you can avoid permanent device loss (PDL) conditions or all paths down (APD) conditions and perform a planned removal and reconnection of a storage device.

Planned device removal is an intentional disconnection of a storage device. You might also plan to remove a device for such reasons as upgrading your hardware or reconfiguring your storage devices. When you perform an orderly removal and reconnection of a storage device, you complete a number of tasks.

- 1 Migrate virtual machines from the device you plan to detach.
See the *vCenter Server and Host Management* documentation.
- 2 Unmount the datastore deployed on the device.
See [“Unmount Datastores,”](#) on page 168.
- 3 Detach the storage device.
See [“Detach Storage Devices,”](#) on page 128.
- 4 For an iSCSI device with a single LUN per target, delete the static target entry from each iSCSI HBA that has a path to the storage device.
See [“Remove Dynamic or Static iSCSI Targets,”](#) on page 98.

- 5 Perform any necessary reconfiguration of the storage device by using the array console.
- 6 Reattach the storage device.
See [“Attach Storage Devices,”](#) on page 128.
- 7 Mount the datastore and restart the virtual machines. See [“Mount Datastores,”](#) on page 168.

Detach Storage Devices

Safely detach a storage device from your host.

You might need to detach the device to make it inaccessible to your host, when, for example, you perform a hardware upgrade on the storage side.

Prerequisites

- The device does not contain any datastores.
- No virtual machines use the device as an RDM disk.
- The device does not contain a diagnostic partition or a scratch partition.

Procedure

- 1 In the vSphere Web Client, display storage devices.
- 2 Select the device to detach and click the **Detach** icon.

The device becomes inaccessible. The operational state of the device changes to Unmounted.

What to do next

If multiple hosts share the device, detach the device from each host.

Attach Storage Devices

Reattach a storage device that you previously detached.

Procedure

- 1 In the vSphere Web Client, display storage devices.
- 2 Select the detached storage device and click the **Attach** icon.

The device becomes accessible.

Recovering From PDL Conditions

An unplanned permanent device loss (PDL) condition occurs when a storage device becomes permanently unavailable without being properly detached from the ESXi host.

The following items in the vSphere Web Client indicate that the device is in the PDL state:

- The datastore deployed on the device is unavailable.
- Operational state of the device changes to Lost Communication.
- All paths are shown as Dead.
- A warning about the device being permanently inaccessible appears in the VMkernel log file.

To recover from the unplanned PDL condition and remove the unavailable device from the host, you need to perform a number of tasks.

- 1 Power off and unregister all virtual machines that are running on the datastore affected by the PDL condition.

- 2 Unmount the datastore.
See [“Unmount Datastores,”](#) on page 168.
- 3 Perform a rescan on all ESXi hosts that had access to the device.
See [“Perform Storage Rescan,”](#) on page 125.

NOTE If the rescan is not successful and the host continues to list the device, some pending I/O or active references to the device might still exist. Check for virtual machines, templates, ISO images, raw device mappings, and so on that might still have an active reference to the device or datastore.

Handling Transient APD Conditions

A storage device is considered to be in the all paths down (APD) state when it becomes unavailable to your ESXi host for an unspecified period of time.

The reasons for an APD state can be, for example, a failed switch or a disconnected storage cable.

In contrast with the permanent device loss (PDL) state, the host treats the APD state as transient and expects the device to be available again.

The host indefinitely continues to retry issued commands in an attempt to reestablish connectivity with the device. If the host's commands fail the retries for a prolonged period of time, the host and its virtual machines might be at risk of having performance problems and potentially becoming unresponsive.

To avoid these problems, your host uses a default APD handling feature. When a device enters the APD state, the system immediately turns on a timer and allows your host to continue retrying nonvirtual machine commands for a limited time period.

By default, the APD timeout is set to 140 seconds, which is typically longer than most devices need to recover from a connection loss. If the device becomes available within this time, the host and its virtual machine continue to run without experiencing any problems.

If the device does not recover and the timeout ends, the host stops its attempts at retries and terminates any nonvirtual machine I/O. Virtual machine I/O will continue retrying. The vSphere Web Client displays the following information for the device with the expired APD timeout:

- The operational state of the device changes to `Dead` or `Error`.
- All paths are shown as `Dead`.
- Datastores on the device are dimmed.

Even though the device and datastores are unavailable, virtual machines remain responsive. You can power off the virtual machines or migrate them to a different datastore or host.

If later one or more device paths becomes operational, subsequent I/O to the device is issued normally and all special APD treatment ends.

Disable Storage APD Handling

The storage all paths down (APD) handling on your ESXi host is enabled by default. When it is enabled, the host continues to retry nonvirtual machine I/O commands to a storage device in the APD state for a limited time period. When the time period expires, the host stops its retry attempts and terminates any nonvirtual machine I/O. You can disable the APD handling feature on your host.

If you disable the APD handling, the host will indefinitely continue to retry issued commands in an attempt to reconnect to the APD device. Continuing to retry is the same behavior as in ESXi version 5.0. This behavior might cause virtual machines on the host to exceed their internal I/O timeout and become unresponsive or fail. The host might become disconnected from vCenter Server.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Settings**.
- 3 Under System, click **Advanced System Settings**.
- 4 Under Advanced System Settings, select the **Misc.APDHandlingEnable** parameter and click the Edit icon.
- 5 Change the value to 0.

If you disabled the APD handling, you can reenable it when a device enters the APD state. The internal APD handling feature turns on immediately and the timer starts with the current timeout value for each device in APD.

Change Timeout Limits for Storage APD

The timeout parameter controls how many seconds the ESXi host will retry nonvirtual machine I/O commands to a storage device in an all paths down (APD) state. If needed, you can change the default timeout value.

The timer starts immediately after the device enters the APD state. When the timeout expires, the host marks the APD device as unreachable and fails any pending or new nonvirtual machine I/O. Virtual machine I/O will continue to be retried.

The default timeout parameter on your host is 140 seconds. You can increase the value of the timeout if, for example, storage devices connected to your ESXi host take longer than 140 seconds to recover from a connection loss.

NOTE If you change the timeout value while an APD is in progress, it will not effect the timeout for that APD.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Settings**.
- 3 Under System, click **Advanced System Settings**.
- 4 Under Advanced System Settings, select the **Misc.APDTimeout** parameter and click the Edit icon.
- 5 Change the default value.

You can enter a value between 20 and 99999 seconds.

Check the Connection Status of a Storage Device

Use the `esxcli` command to verify the connection status of a particular storage device.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 Run the `esxcli --server=server_name storage core device list -d=device_ID` command.

- 2 Check the connection status in the `Status:` field.
 - `on` - Device is connected.
 - `dead` - Device has entered the APD state. The APD timer starts.
 - `dead timeout` - The APD timeout has expired.
 - `not connected` - Device is in the PDL state.

Device Connectivity Problems and High Availability

When a device enters a Permanent Device Loss (PDL) or an All Paths Down (APD) state, vSphere High Availability (HA) can detect connectivity problems and provide automated recovery for affected virtual machines.

vSphere HA uses VM Component Protection (VMCP) to protect virtual machines running on a host in a vSphere HA cluster against accessibility failures. For more information about VMCP and how to configure responses for datastores and virtual machines when an APD or PDL condition occurs, see the *vSphere Availability* documentation.

Edit Configuration File Parameters

You can change or add virtual machine configuration parameters when instructed by a VMware technical support representative, or if you see VMware documentation that instructs you to add or change a parameter to fix a problem with your system.

IMPORTANT Changing or adding parameters when a system does not have problems might lead to decreased system performance and instability.

The following conditions apply:

- To change a parameter, you change the existing value for the keyword/value pair. For example, if you start with the keyword/value pair, `keyword/value`, and change it to `keyword/value2`, the result is `keyword=value2`.
- You cannot delete a configuration parameter entry.



CAUTION You must assign a value to configuration parameter keywords. If you do not assign a value, the keyword can return a value of 0, false, or disable, which can result in a virtual machine that cannot power on.

Procedure

- 1 Right-click a virtual machine in the inventory and select **Edit Settings**.
- 2 Click the **VM Options** tab and expand **Advanced**.
- 3 Click **Edit Configuration**.
- 4 (Optional) To add a parameter, click **Add Row** and type a name and value for the parameter.
- 5 (Optional) To change a parameter, type a new value in the **Value** text box for that parameter.
- 6 Click **OK**.

Enable or Disable the Locator LED on Storage Devices

Use the locator LED to identify specific storage devices, so that you can locate them among other devices. You can turn the locator LED on or off.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Devices**.
- 4 From the list of storage devices, select one or more disks and enable or disable the locator LED indicator.

Option	Description
Enable	Click the Turns on the locator LED icon.
Disable	Click the Turns off the locator LED icon.

Working with Flash Devices

In addition to regular storage hard disk drives (HDDs), vSphere supports flash storage devices.

Unlike the regular HDDs that are electromechanical devices containing moving parts, flash devices use semiconductors as their storage medium and have no moving parts. Typically, flash devices are very resilient and provide faster access to data.

To detect flash devices, ESXi uses an inquiry mechanism based on T10 standards. The ESXi host can detect flash devices on a number of storage arrays. Check with your vendor whether your storage array supports the ESXi mechanism of flash device detection.

After the host detects the flash devices, you can use them for a variety of tasks and functionalities.

This chapter includes the following topics:

- [“Using Flash Devices,”](#) on page 134
- [“Marking Storage Devices,”](#) on page 134
- [“Monitor Flash Devices,”](#) on page 136
- [“Best Practices for Flash Devices,”](#) on page 136
- [“About Virtual Flash Resource,”](#) on page 137
- [“Configuring Host Swap Cache,”](#) on page 139

Using Flash Devices

You can use flash devices for a variety of functionalities.

Table 14-1. Using Flash Devices with vSphere

Functionality	Description
Virtual SAN	Virtual SAN requires flash devices. For more information, see the <i>Administering VMware Virtual SAN</i> documentation.
VMFS Datastores	<p>You can create VMFS datastores on flash devices. Use the datastores for the following purposes:</p> <ul style="list-style-type: none"> ■ Store virtual machines. Certain guest operating systems can identify virtual disks stored on these datastores as flash virtual disks. See “Identifying Flash Virtual Disks,” on page 134. ■ Allocate datastore space for ESXi host swap cache. See “Configuring Host Swap Cache,” on page 139
Virtual Flash Resource (VFFS)	<p>Set up a virtual flash resource and use it for the following functionalities:</p> <ul style="list-style-type: none"> ■ Use as Virtual Flash Read Cache for your virtual machines. See Chapter 15, “About VMware vSphere Flash Read Cache,” on page 141. ■ Allocate the virtual flash resource for ESXi host swap cache. This is an alternative method of host cache configuration that uses VFFS volumes instead of VMFS datastores. See “Configure Host Swap Cache with Virtual Flash Resource,” on page 140.

Identifying Flash Virtual Disks

Guest operating systems can identify virtual disks that reside on flash-based datastores as flash virtual disks.

To verify if this feature is enabled, guest operating systems can use standard inquiry commands such as SCSI VPD Page (B1h) for SCSI devices and ATA IDENTIFY DEVICE (Word 217) for IDE devices.

For linked clones, native snapshots, and delta-disks, the inquiry commands report the virtual flash status of the base disk.

Operating systems can detect that a virtual disk is a flash disk under the following conditions:

- Detection of flash virtual disks is supported on ESXi 5.x and later hosts and virtual hardware version 8 or later.
- Detection of flash virtual disks is supported only on VMFS5 or later.
- If virtual disks are located on shared VMFS datastores with flash device extents, the device must be marked as flash on all hosts.
- For a virtual disk to be identified as virtual flash, all underlying physical extents should be flash-backed.

Marking Storage Devices

You can use the vSphere Web Client to mark storage devices that are not automatically recognized as local flash devices.

When you configure Virtual SAN or set up a virtual flash resource, your storage environment must include local flash devices.

However, ESXi might not recognize certain storage devices as flash devices when their vendors do not support automatic flash device detection. In other cases, certain non-SATA SAS flash devices might not be detected as local. When devices are not recognized as local flash, they are excluded from the list of devices offered for Virtual SAN or virtual flash resource. Marking these devices as local flash makes them available for virtual SAN and virtual flash resource.

Mark Storage Devices as Flash

If ESXi does not automatically recognize its devices as flash, mark them as flash devices.

ESXi does not recognize certain devices as flash when their vendors do not support automatic flash disk detection. The Drive Type column for the devices shows HDD as their type.



CAUTION Marking HDD disks as flash disks could deteriorate the performance of datastores and services that use them. Mark disks as flash disks only if you are certain that those disks are flash disks.

Prerequisites

Verify that the device is not in use.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Devices**.
- 4 From the list of storage devices, select one or several HDD devices that need to be recognized as flash devices and click the **Mark as Flash Disks** icon.
- 5 Click **Yes** to save your changes.

The type of the devices changes to flash.

What to do next

If the flash device that you mark is shared among multiple hosts, make sure that you mark the device from all hosts that share the device.

Mark Storage Devices as Local

ESXi enables you to mark devices as local. This is useful in cases when ESXi is unable to determine whether certain devices are local.

Prerequisites

- Make sure that the device is not shared.
- Power off virtual machines that reside on the device and unmount an associated datastore.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Devices**.
- 4 From the list of storage devices, select one or several remote devices that need to be marked as local and click the **Mark as Local for the Host** icon.
- 5 Click **Yes** to save your changes.

Monitor Flash Devices

You can monitor certain critical flash device parameters, including Media Wearout Indicator, Temperature, and Reallocated Sector Count, from an ESXi host.

Use the `esxcli` command to monitor flash devices.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ Run the following command to display the flash device statistics:

```
esxcli server=server_name storage core device smart get -d=flash device_ID
```

Best Practices for Flash Devices

Follow these best practices when you use flash devices in vSphere environment.

- Make sure to use the latest firmware with flash devices. Frequently check with your storage vendors for any updates.
- Carefully monitor how intensively you use the flash device and calculate its estimated lifetime. The lifetime expectancy depends on how actively you continue to use the flash device.

Estimate Lifetime of Flash Devices

When working with flash devices, monitor how actively you use them and calculate their estimated lifetime.

Typically, storage vendors provide reliable lifetime estimates for a flash device under ideal conditions. For example, a vendor might guarantee a lifetime of 5 years under the condition of 20GB writes per day. However, the more realistic life expectancy of the device depends on how many writes per day your host actually generates. Follow these steps to calculate the lifetime of the flash device.

Procedure

- 1 Obtain the number of writes on the flash device by running the `esxcli storage core device stats get -d=device_ID` command.

The Write Operations item in the output shows the number. You can average this number over a period of time.

- 2 Estimate lifetime of your device by using the following formula:

vendor provided number of writes per day times vendor provided life span divided by actual average number of writes per day

For example, if your vendor guarantees a lifetime of 5 years under the condition of 20 GB writes per day, and the actual number of writes per day is 30 GB, the life span of your flash device will be approximately 3.3 years.

About Virtual Flash Resource

You can aggregate local flash devices on an ESXi host into a single virtualized caching layer called virtual flash resource.

When you set up the virtual flash resource, you create a new file system, Virtual Flash File System (VFFS). VFFS is a derivative of VMFS, which is optimized for flash devices and is used to group the physical flash devices into a single caching resource pool. As a non-persistent resource, it cannot be used to store virtual machines.

The following vSphere functionalities require the virtual flash resource:

- Virtual machine read cache. See [Chapter 15, “About VMware vSphere Flash Read Cache,”](#) on page 141.
- Host swap cache. See [“Configure Host Swap Cache with Virtual Flash Resource,”](#) on page 140.

Watch the video for information about virtual flash resource.



Configuring Virtual Flash Resource

(http://link.brightcove.com/services/player/bcpid2296383276001?bctid=ref:video_configure_flash_resource)

Before setting up the virtual flash resource, make sure that you use devices approved by the *VMware Compatibility Guide*.

Considerations for Virtual Flash Resource

When you configure a virtual flash resource that is consumed by ESXi hosts and virtual machines, several considerations apply.

- You can have only one virtual flash resource, also called a VFFS volume, on a single ESXi host. The virtual flash resource is managed only at the host's level.
- You cannot use the virtual flash resource to store virtual machines. Virtual flash resource is a caching layer only.
- You can use only local flash devices for the virtual flash resource.
- You can create the virtual flash resource from mixed flash devices. All device types are treated the same and no distinction is made between SAS, SATA, or PCI express connectivity. When creating the resource from mixed flash devices, make sure to group similar performing devices together to maximize performance.
- You cannot use the same flash devices for the virtual flash resource and Virtual SAN. Each requires its own exclusive and dedicated flash device.
- After you set up a virtual flash resource, the total available capacity can be used and consumed by both ESXi hosts as host swap cache and virtual machines as read cache.
- You cannot choose individual flash devices to be used for either swap cache or read cache. All flash devices are combined into a single flash resource entity.

Set Up Virtual Flash Resource

You can set up a virtual flash resource or add capacity to existing virtual flash resource.

To set up a virtual flash resource, you use local flash devices connected to your host. To increase the capacity of your virtual flash resource, you can add more devices, up to the maximum number indicated in the *Configuration Maximums* documentation. An individual flash device must be exclusively allocated to the virtual flash resource and cannot be shared with any other vSphere service, such as Virtual SAN or VMFS.

Procedure

- 1 In the vSphere Web Client, navigate to the host.
- 2 Click the **Manage** tab and click **Settings**.
- 3 Under Virtual Flash, select **Virtual Flash Resource Management** and click **Add Capacity**.
- 4 From the list of available flash devices, select one or more devices to use for the virtual flash resource and click **OK**.

Under certain circumstances, you might not be able to see flash devices on the list. For more information, see the Troubleshooting Flash Devices section in the *vSphere Troubleshooting* documentation.

The virtual flash resource is created. The Device Backing area lists all devices that you use for the virtual flash resource.

What to do next

You can use the virtual flash resource for cache configuration on the host and Flash Read Cache configuration on virtual disks.

You can increase the capacity by adding more flash devices to the virtual flash resource.

Remove Virtual Flash Resource

You might need to remove a virtual flash resource deployed on local flash devices to free the devices for other services.

You cannot remove a virtual flash resource if it is configured with host swap cache or if the host has virtual machines configured with Flash Read Cache that are powered on.

Procedure

- 1 In the vSphere Web Client, navigate to the host that has virtual flash configured.
- 2 Click the **Manage** tab and click **Settings**.
- 3 Under Virtual Flash, select **Virtual Flash Resource Management** and click **Remove All**.

After you remove the virtual flash resource and erase the flash device, the device is available for other operations.

Virtual Flash Advanced Settings

You can change advanced options for virtual flash.

Procedure

- 1 In the vSphere Web Client, navigate to the host.
- 2 Click the **Manage** tab and click **Settings**.
- 3 Under **System**, click **Advanced System Settings**.

- 4 Select the setting to change and click the **Edit** button.

Option	Description
VFLASH.VFlashResourceUsageThreshold	The system triggers the <code>Host vFlash resource usage</code> alarm when a virtual flash resource usage exceeds the threshold. The default threshold is 80%. You can change the threshold to an appropriate value. The alarm is automatically cleared when the virtual flash resource usage drops below the threshold.
VFLASH.MaxResourceGBForVmCache	An ESXi host stores Flash Read Cache metadata in RAM. The default limit of total virtual machine cache size on the host is 2TB. You can reconfigure this setting. You must restart the host for the new setting to take effect.

- 5 Click **OK**.

Configuring Host Swap Cache

Your ESXi hosts can use a portion of a flash-backed storage entity as a swap cache shared by all virtual machines.

The host-level cache is made up of files on a low-latency disk that ESXi uses as a write back cache for virtual machine swap files. The cache is shared by all virtual machines running on the host. Host-level swapping of virtual machine pages makes the best use of potentially limited flash device space.

Depending on your environment and licensing package, the following methods of configuring the host-level swap cache are available. Both methods provide similar results.

- You can create a VMFS datastore on a flash device, and then use the datastore to allocate space for host cache. The host reserves a certain amount of space for swapping to host cache.
- If you have an appropriate vSphere license that allows you to set up and manage a virtual flash resource, you can use the resource to configure swap cache on the host. The host swap cache is allocated from a portion of the virtual flash resource.

Configure Host Cache with VMFS Datastore

Turn on the host's ability to swap to host cache. You can also change the percentage of space allocated for host cache.

Use this task if you do not have an appropriate licence that allows you to set up and manage a virtual flash resource. If you have the license, use the virtual flash resource for host cache configuration.

Prerequisites

Create a flash-backed VMFS datastore. See [“Create a VMFS Datastore,”](#) on page 160.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Host Cache Configuration**.
- 4 Select the datastore in the list and click the **Allocate space for host cache** icon.
- 5 To enable the host swap cache on a per-datastore basis, select the **Allocate space for host cache** check box.
By default, maximum available space is allocated for host cache.
- 6 (Optional) To change the host cache size, select **Custom size** and make appropriate adjustments.
- 7 Click **OK**.

Configure Host Swap Cache with Virtual Flash Resource

You can reserve a certain amount of virtual flash resource for host swap cache.

Prerequisites

Set up a virtual flash resource. [“Set Up Virtual Flash Resource,”](#) on page 137.

NOTE If an ESXi host configured with virtual flash is in maintenance mode, you cannot add or modify a host swap cache. You must first exit maintenance mode on the host before you configure a host swap cache.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Settings**.
- 3 Under Virtual Flash, select **Virtual Flash Host Swap Cache Configuration** and click **Edit**.
- 4 Select the **Enable virtual flash host swap cache** check box.
- 5 Specify the amount of virtual flash resource to reserve for host swap cache.
- 6 Click **OK**.

About VMware vSphere Flash Read Cache

15

Flash Read Cache™ lets you accelerate virtual machine performance through the use of host resident flash devices as a cache.

You can reserve a Flash Read Cache for any individual virtual disk. The Flash Read Cache is created only when a virtual machine is powered on, and it is discarded when a virtual machine is suspended or powered off. When you migrate a virtual machine you have the option to migrate the cache. By default the cache is migrated if the virtual flash module on the source and destination hosts are compatible. If you do not migrate the cache, the cache is rewarmed on the destination host. You can change the size of the cache while a virtual machine is powered on. In this instance, the existing cache is discarded and a new write-through cache is created, which results in a cache warm up period. The advantage of creating a new cache is that the cache size can better match the application's active data.

Flash Read Cache supports write-through or read caching. Write-back or write caching are not supported. Data reads are satisfied from the cache, if present. Data writes are dispatched to the backing storage, such as a SAN or NAS. All data that is read from or written to the backing storage is unconditionally stored in the cache.

Flash Read Cache does not support RDMs in physical compatibility. Virtual compatibility RDMs are supported with Flash Read Cache.

Watch the video for more information about Flash Read Cache.



Configuring vSphere Flash Read Cache
(http://link.brightcove.com/services/player/bcpid2296383276001?bctid=ref:video_config_vsphere_flash_read_cache)

NOTE Not all workloads benefit with a Flash Read Cache. The performance boost depends on your workload pattern and working set size. Read-intensive workloads with working sets that fit into the cache can benefit from a Flash Read Cache configuration. By configuring Flash Read Cache for your read-intensive workloads additional I/O resources become available on your shared storage, which can result in a performance increase for other workloads even though they are not configured to use Flash Read Cache.

This chapter includes the following topics:

- “DRS Support for Flash Read Cache,” on page 142
- “vSphere High Availability Support for Flash Read Cache,” on page 142
- “Configure Flash Read Cache for a Virtual Machine,” on page 142
- “Migrating Virtual Machines with Flash Read Cache,” on page 143

DRS Support for Flash Read Cache

DRS supports virtual flash as a resource.

DRS manages virtual machines with Flash Read Cache reservations. Every time DRS runs, it displays the available virtual flash capacity reported by the ESXi host. Each host supports one virtual flash resource. DRS selects a host that has sufficient available virtual flash capacity to start a virtual machine. DRS treats powered-on virtual machines with a Flash Read Cache as soft affinity to their current host and moves them only for mandatory reasons or if necessary to correct host over-utilization.

vSphere High Availability Support for Flash Read Cache

Flash Read Cache is supported by High Availability (HA).

When vSphere HA restarts a virtual machine configured with Flash Read Cache, the virtual machine is restarted on a host in the cluster where the Flash Read Cache, CPU, Memory and overhead reservations are met. vSphere HA will not restart a virtual machine if unreserved flash is insufficient to meet the virtual flash reservation. You must manually reconfigure a virtual machine to reduce or drop the Flash Read Cache, if the target host does not have sufficient virtual flash resource available.

Configure Flash Read Cache for a Virtual Machine

You can configure Flash Read Cache for a virtual machine compatible with ESXi 5.5 or later.

Enabling Flash Read Cache lets you specify block size and cache size reservation.

Block size is the minimum number of contiguous bytes that can be stored in the cache. This block size can be larger than the nominal disk block size of 512 bytes, between 4KB and 1024KB. If a guest operating system writes a single 512 byte disk block, the surrounding cache block size bytes will be cached. Do not confuse cache block size with disk block size.

Reservation is a reservation size for cache blocks. There is a minimum number of 256 cache blocks. If the cache block size is 1MB, then the minimum cache size is 256MB. If the cache block size is 4K, then the minimum cache size is 1MB.

For more information about sizing guidelines, search for the *Performance of vSphere Flash Read Cache in VMware vSphere* white paper on the VMware web site.

Prerequisites

- Set up virtual flash resource. See [“Set Up Virtual Flash Resource,”](#) on page 137.
- Verify that the virtual machine is compatible with ESXi 5.5 or later.

Procedure

- 1 To locate a virtual machine, select a data center, folder, cluster, resource pool, host, or vApp.
- 2 Click the **Related Objects** tab and click **Virtual Machines**.
- 3 Right-click the virtual machine and select **Edit Settings**.
- 4 On the **Virtual Hardware** tab, expand **Hard disk** to view the disk options.
- 5 To enable Flash Read Cache for the virtual machine, enter a value in the **Virtual Flash Read Cache** text box.

- Click **Advanced** to specify the following parameters.

Option	Description
Reservation	Select a cache size reservation.
Block Size	Select a block size.

- Click **OK**.

Migrating Virtual Machines with Flash Read Cache

You can migrate virtual machines with Flash Read Cache.

Procedure

- Right-click the virtual machine and select **Migrate**.
- Select **Change Host**.
- Select the destination host.
- Under **Select Flash Read Cache Settings**, select **Always Migrate Flash Read Cache**.

Flash Read Cache Migration Settings	Description
Automatic	Migrate the Flash Read Cache if possible.
Migrate	Migration will proceed only if the Flash Read Cache can be migrated to the target destination. This option is useful when the cache is small or the cache size closely matches the application's active data.
Discard	Flash Read Cache is not migrated, the cache is dropped. This option is useful when the cache size is large or the cache size is larger than the application's active data.

- Click **Finish**.

Working with Datastores

Datastores are logical containers, analogous to file systems, that hide specifics of physical storage and provide a uniform model for storing virtual machine files. Datastores can also be used for storing ISO images, virtual machine templates, and floppy images.

Depending on the storage you use, datastores can be of the following types:

- VMFS datastores that are backed by the Virtual Machine File System format. See [“Understanding VMFS Datastores,”](#) on page 146.
- NFS datastores that are backed by the Network File System NFS format. See [“Understanding Network File System Datastores,”](#) on page 152.
- Virtual SAN datastores. See the *Administering VMware Virtual SAN* documentation.
- Virtual Volumes datastores. See [Chapter 19, “Working with Virtual Volumes,”](#) on page 211.

After creating datastores, you can perform several administrative operations on the datastores. Certain operations, such as renaming a datastore, are available for all types of datastores. While others apply to specific types of datastores.

You can also organize datastores in different ways. For example, you can group them into folders according to business practices. This allows you to assign the same permissions and alarms on the datastores in the group at one time.

You can also add datastores to datastore clusters. A datastore cluster is a collection of datastores with shared resources and a shared management interface. When you create a datastore cluster, you can use Storage DRS to manage storage resources. For information about datastore clusters, see the *vSphere Resource Management* documentation.

This chapter includes the following topics:

- [“Understanding VMFS Datastores,”](#) on page 146
- [“Understanding Network File System Datastores,”](#) on page 152
- [“Creating Datastores,”](#) on page 160
- [“Managing Duplicate VMFS Datastores,”](#) on page 162
- [“Upgrading VMFS Datastores,”](#) on page 164
- [“Increasing VMFS Datastore Capacity,”](#) on page 165
- [“Administrative Operations for Datastores,”](#) on page 167
- [“Set Up Dynamic Disk Mirroring,”](#) on page 173
- [“Collecting Diagnostic Information for ESXi Hosts on a Storage Device,”](#) on page 174
- [“Checking Metadata Consistency with VOMA,”](#) on page 177

- [“Configuring VMFS Pointer Block Cache,”](#) on page 179

Understanding VMFS Datastores

To store virtual disks, ESXi uses datastores, which are logical containers that hide specifics of physical storage from virtual machines and provide a uniform model for storing virtual machine files. Datastores that you deploy on block storage devices use the vSphere VMFS format, a special high-performance file system format that is optimized for storing virtual machines.

Several versions of the VMFS file system have been released since its introduction. The following table shows host-to-VMFS version relationships.

Table 16-1. Host access to VMFS version

VMFS	ESX/ESXi 3.x host	ESX/ESXi 4.x host	ESXi 5.x host	ESXi 6.0 host
VMFS2	RO	RO	N	N
VMFS3	RW	RW	RW	RW
				NOTE You can continue to use existing VMFS3 datastores, but you cannot create new ones. If you have existing VMFS3 datastores, upgrade them to VMFS5.
VMFS5	N	N	RW	RW

- RW: Complete read and write support. You can create and power on virtual machines.
- RO: Read only support. You cannot create or power on virtual machines.
- N: No access. ESXi 5.x and later hosts do not support VMFS2. If your datastore was formatted with VMFS2, first upgrade the datastore to VMFS3 using legacy hosts.

Use the vSphere Web Client to set up a VMFS datastore in advance on a block-based storage device that your ESXi host discovers. A VMFS datastore can be extended to span several physical storage extents, including SAN LUNs and local storage. This feature allows you to pool storage and gives you flexibility in creating the datastore necessary for your virtual machines.

NOTE Pooling ATS-capable hardware creates a spanned VMFS datastore that can use ATS-only locking mechanism. If any device is not ATS-capable, the datastore cannot be ATS-only, but uses ATS+SCSI locking.

You can increase the capacity of a datastore while virtual machines are running on the datastore. This ability lets you add new space to your VMFS datastores as your virtual machine requires it. VMFS is designed for concurrent access from multiple physical machines and enforces the appropriate access controls on virtual machine files.

Characteristics of VMFS5 Datastores

VMFS5 provides many improvements in scalability and performance.

VMFS5 has the following Characteristics:

- Greater than 2TB storage devices for each VMFS5 extent.
- Support of virtual machines with large capacity virtual disks, or disks greater than 2TB.
- Increased resource limits such as file descriptors.
- Standard 1MB file system block size with support of 2TB virtual disks.
- Greater than 2TB disk size for RDMs.

- Support of small files of 1KB.
- Ability to open any file located on a VMFS5 datastore in a shared mode by a maximum of 32 hosts.
- Scalability improvements on storage devices that support hardware acceleration. For information, see [Chapter 22, “Storage Hardware Acceleration,”](#) on page 243.
- Default use of ATS-only locking mechanisms on storage devices that support ATS. For information about ATS-only locking and how to upgrade to ATS-only locking, see [“VMFS Locking Mechanisms,”](#) on page 149.
- Ability to reclaim physical storage space on thin provisioned storage devices. For information, see [“Array Thin Provisioning and VMFS Datastores,”](#) on page 257.
- Online upgrade process that upgrades existing datastores without disrupting hosts or virtual machines that are currently running. For information, see [“Upgrading VMFS Datastores,”](#) on page 164.

For information about block size limitations of a VMFS datastore, see the VMware knowledge base article at <http://kb.vmware.com/kb/1003565>.

VMFS Datastores and Storage Disk Formats

Storage devices that your host supports can use either the master boot record (MBR) format or the GUID partition table (GPT) format.

If you create a new VMFS5 datastore, the device is formatted with GPT. The GPT format enables you to create datastores larger than 2TB and up to 64TB for a single extent.

VMFS3 datastores continue to use the MBR format for their storage devices. Consider the following items when you work with VMFS3 datastores:

- For VMFS3 datastores, the 2TB limit still applies, even when the storage device has a capacity of more than 2TB. To be able to use the entire storage space, upgrade a VMFS3 datastore to VMFS5. Conversion of the MBR format to GPT happens only after you expand the datastore to a size larger than 2TB.
- When you upgrade a VMFS3 datastore to VMFS5, the datastore uses the MBR format. Conversion to GPT happens only after you expand the datastore to a size larger than 2TB.
- When you upgrade a VMFS3 datastore, remove from the storage device any partitions that ESXi does not recognize, for example, partitions that use the EXT2 or EXT3 formats. Otherwise, the host cannot format the device with GPT and the upgrade fails.
- You cannot expand a VMFS3 datastore on devices that have the GPT partition format.

VMFS Datastores as Repositories

ESXi can format SCSI-based storage devices as VMFS datastores. VMFS datastores primarily serve as repositories for virtual machines.

With VMFS5, you can have up to 256 VMFS datastores per host, with the maximum size of 64TB. The required minimum size for a VMFS datastore is 1.3GB, however, the recommended minimum size is 2GB.

NOTE Always have only one VMFS datastore for each LUN.

You can store multiple virtual machines on the same VMFS datastore. Each virtual machine, encapsulated in a set of files, occupies a separate single directory. For the operating system inside the virtual machine, VMFS preserves the internal file system semantics, which ensures correct application behavior and data integrity for applications running in virtual machines.

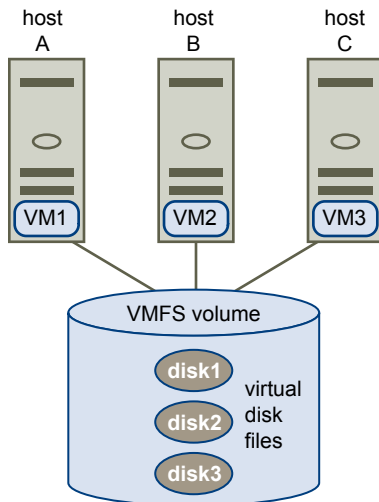
When you run multiple virtual machines, VMFS provides specific locking mechanisms for virtual machine files, so that virtual machines can operate safely in a SAN environment where multiple ESXi hosts share the same VMFS datastore.

In addition to virtual machines, the VMFS datastores can store other files, such as virtual machine templates and ISO images.

Sharing a VMFS Datastore Across Hosts

As a cluster file system, VMFS lets multiple ESXi hosts access the same VMFS datastore concurrently.

Figure 16-1. Sharing a VMFS Datastore Across Hosts



For information on the maximum number of hosts that can connect to a single VMFS datastore, see the *Configuration Maximums* document.

To ensure that the same virtual machine is not accessed by multiple servers at the same time, VMFS provides on-disk locking.

Sharing the same VMFS volume across multiple hosts offers the following advantages:

- You can use VMware Distributed Resource Scheduling (DRS) and VMware High Availability (HA).
You can distribute virtual machines across different physical servers. That means you run a mix of virtual machines on each server so that not all experience high demand in the same area at the same time. If a server fails, you can restart virtual machines on another physical server. In case of a failure, the on-disk lock for each virtual machine is released. For more information about VMware DRS, see the *vSphere Resource Management* documentation. For information about VMware HA, see the *vSphere Availability* documentation.
- You can use vMotion to migrate running virtual machines from one physical server to another. For information about migrating virtual machines, see the *vCenter Server and Host Management* documentation.

VMFS Metadata Updates

A VMFS datastore holds virtual machine files, directories, symbolic links, RDM descriptor files, and so on. The datastore also maintains a consistent view of all the mapping information for these objects. This mapping information is called metadata.

Metadata is updated each time you perform datastore or virtual machine management operations. Examples of operations requiring metadata updates include the following:

- Creating, growing, or locking a virtual machine file

- Changing a file's attributes
- Powering a virtual machine on or off
- Creating or deleting a VMFS datastore
- Expanding a VMFS datastore
- Creating a template
- Deploying a virtual machine from a template
- Migrating a virtual machine with vMotion

When metadata changes are made in a shared storage environment, VMFS uses special locking mechanisms to protect its data and prevent multiple hosts from concurrently writing to the metadata.

VMFS Locking Mechanisms

In a shared storage environment, when multiple hosts access the same VMFS datastore, specific locking mechanisms are used. These locking mechanisms prevent multiple hosts from concurrently writing to the metadata and ensure that no data corruption occurs.

Depending on its configuration and the type of underlying storage, a VMFS datastore can exclusively use the atomic test and set locking mechanism (ATS-only), or use a combination of ATS and SCSI reservations (ATS+SCSI).

ATS-Only Mechanism

For storage devices that support T10 standard-based VAAI specifications, VMFS provides ATS locking, also called hardware assisted locking. The ATS algorithm supports discrete locking per disk sector. All newly formatted VMFS5 datastores use the ATS-only mechanism if the underlying storage supports it, and never use SCSI reservations.

When you create a multi-extent datastore where ATS is used, vCenter Server filters out non-ATS devices. This filtering allows you to use only those devices that support the ATS primitive.

In certain cases, you might need to turn off the ATS-only setting for a VMFS5 datastore. For information, see [“Change Locking Mechanism to ATS+SCSI,”](#) on page 152.

ATS+SCSI Mechanism

A VMFS datastore that supports the ATS+SCSI mechanism is configured to use ATS and attempts to use it when possible. If ATS fails, the VMFS datastore reverts to SCSI reservations. In contrast with the ATS locking, the SCSI reservations lock an entire storage device while an operation that requires metadata protection is performed. After the operation completes, VMFS releases the reservation and other operations can continue.

Datastores that use the ATS+SCSI mechanism include VMFS5 datastores that were upgraded from VMFS3. In addition, new VMFS5 datastores on storage devices that do not support ATS use the ATS+SCSI mechanism.

If your VMFS datastore reverts to SCSI reservations, you might notice performance degradation caused by excessive SCSI reservations. For information about how to reduce SCSI reservations, see the *vSphere Troubleshooting* documentation.

Display VMFS Locking Information

Use the `esxcli` command to obtain information about the locking mechanism that a VMFS datastore uses.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ To display information related to VMFS locking mechanisms, run the following command:

```
esxcli --server=server_name storage vmfs lockmode list .
```

The table lists items that the output of the command might include.

Table 16-2. VMFS Locking Information

Fields	Values	Descriptions
Locking Mode		Indicates the locking configuration of the datastore.
	ATS-only	The datastore is configured to use ATS-only.
	ATS+SCSI	The datastore is configured to use ATS, but can revert to SCSI if ATS fails or is not supported.
	ATS upgrade pending	The datastore is in the process of an online upgrade to ATS-only.
	ATS downgrade pending	The datastore is in the process of an online downgrade to ATS+SCSI.
ATS Compatible		Indicates whether the datastore can be configured for ATS-only or not.
ATS Upgrade Modes		Indicates the type of upgrade that the datastore supports.
	None	The datastore is not ATS-only compatible.
	Online	The datastore can be used during its upgrade to ATS-only.
	Offline	The datastore cannot be used during its upgrade to ATS-only.
ATS Incompatibility Reason		If the datastore is not compatible with ATS-only, indicates the reason for the incompatibility.

Change VMFS Locking to ATS-Only

If your VMFS5 datastores use the ATS+SCSI locking mechanism, you can change to ATS-only locking.

Typically, VMFS5 datastores that were upgraded from VMFS3 use the ATS+SCSI locking mechanism. If the datastores are deployed on ATS-enabled hardware, they are generally eligible for an upgrade to ATS-only locking. Depending on your vSphere environment, you can use one of the following upgrade modes:

- The online upgrade to ATS-only is available for most single-extent VMFS5 datastores. While you perform the online upgrade on one of the hosts, other hosts can continue using the datastore.
- The offline upgrade to ATS-only must be used for VMFS5 datastores that span multiple physical extents. Datastores composed of multiple extents are not eligible for the online upgrade. These datastores require that no hosts actively use the datastores at the time of the upgrade request.

Procedure

- 1 [Prepare for an Upgrade to ATS-Only](#) on page 151
You must perform several steps to prepare your environment for an online or offline upgrade to ATS-only locking.
- 2 [Upgrade Locking Mechanism to ATS-Only](#) on page 151
If a VMFS datastore is ATS-only compatible, you can upgrade its locking mechanism from ATS+SCSI to ATS-only.

Prepare for an Upgrade to ATS-Only

You must perform several steps to prepare your environment for an online or offline upgrade to ATS-only locking.

Procedure

- 1 Upgrade all hosts that access the VMFS5 datastore to the newest version of vSphere.
- 2 Determine whether the datastore is eligible for an upgrade of its current locking mechanism by running the `esxcli storage vmfs lockmode list` command.

The following sample output fields indicate that the datastore is eligible for an upgrade, and show its current locking mechanism and an upgrade mode available for the datastore.

```
Locking Mode  ATS Compatible  ATS Upgrade Modes
-----
ATS+SCSI      true             Online or Offline
```

- 3 Depending on the upgrade mode available for the datastore, perform one of the following actions:

Upgrade Mode	Action
Online	Verify that all hosts have consistent storage connectivity to the VMFS datastore.
Offline	Verify that no hosts are actively using the datastore.

Upgrade Locking Mechanism to ATS-Only

If a VMFS datastore is ATS-only compatible, you can upgrade its locking mechanism from ATS+SCSI to ATS-only.

Most datastores that do not span multiple extents are eligible for an online upgrade. While you perform the online upgrade on one of the ESXi hosts, other hosts can continue using the datastore. The online upgrade completes only after all hosts have closed the datastore.

Prerequisites

If you plan to complete the upgrade of the locking mechanism by putting the datastore into maintenance mode, disable Storage DRS. This prerequisite applies only to an online upgrade.

Procedure

- 1 Perform an upgrade of the locking mechanism by running the following command:
`esxcli storage vmfs lockmode set -a|--ats -l|--volume-label= VMFS label -u|--volume-uuid= VMFS UUID.`

- 2 For an online upgrade, perform additional steps.
 - a Close the datastore on all hosts that have access to the datastore, so that the hosts can recognise the change.

You can use one of the following methods:

- Unmount and mount the datastore.
 - Put the datastore into maintenance mode and exit maintenance mode.
- b Verify that the Locking Mode status for the datastore changed to ATS-only by running:


```
esxcli storage vmfs lockmode list
```
 - c If the Locking Mode displays any other status, for example ATS UPGRADE PENDING, check which host has not yet processed the upgrade by running:

```
esxcli storage vmfs host list
```

Change Locking Mechanism to ATS+SCSI

When you create a VMFS5 datastore on a device that supports atomic test and set (ATS) locking, the datastore is set to use the ATS-only locking mechanism. In certain circumstances, you might need to downgrade the ATS-only locking to ATS+SCSI.

You might need to switch to the ATS+SCSI locking mechanism when, for example, your storage device is downgraded or firmware updates fail and the device no longer supports ATS.

The downgrade process is similar to the ATS-only upgrade. As with the upgrade, depending on your storage configuration, you can perform the downgrade in online or offline mode.

Procedure

- 1 Change the locking mechanism to ATS+SCSI by running the following command:


```
esxcli storage vmfs lockmode set -s|--scsi -l|--volume-label= VMFS label -u|--volume-uuid= VMFS UUID.
```
- 2 For an online mode, close the datastore on all hosts that have access to the datastore, so that the hosts can recognise the change.

Understanding Network File System Datastores

An NFS client built into ESXi uses the Network File System (NFS) protocol over TCP/IP to access a designated NFS volume that is located on a NAS server. The ESXi host can mount the volume and use it for its storage needs. vSphere supports versions 3 and 4.1 of the NFS protocol.

Typically, the NFS volume or directory is created by a storage administrator and is exported from the NFS server. The NFS volume is not required to be formatted with a local file system, such as VMFS. You can mount the volume directly on ESXi hosts and use it to store and boot virtual machines in the same way that you use VMFS datastores.

In addition to storing virtual disks on NFS datastores, you can use NFS as a central repository for ISO images, virtual machine templates, and so on. If you use the datastore for ISO images, you can connect the virtual machine's CD-ROM device to an ISO file on the datastore and install a guest operating system from the ISO file.

ESXi supports the following storage capabilities on most NFS volumes:

- vMotion and Storage vMotion
- High Availability (HA) and Distributed Resource Scheduler (DRS)

- Fault Tolerance (FT) and Host Profiles

NOTE NFS 4.1 does not support legacy Fault Tolerance.

- ISO images, which are presented as CD-ROMs to virtual machines
- Virtual machine snapshots
- Virtual machines with large capacity virtual disks, or disks greater than 2TB. Virtual disks created on NFS datastores are thin-provisioned by default, unless you use hardware acceleration that supports the Reserve Space operation. NFS 4.1 does not support hardware acceleration. For information, see [“Hardware Acceleration on NAS Devices,”](#) on page 249.

NFS Storage Guidelines and Requirements

When using NFS storage, you must follow specific configuration, networking, and NFS datastore guidelines.

NFS Server Configuration Guidelines

- Make sure that NFS servers you use are listed in the *VMware HCL*. Use the correct version for the server firmware.
- When configuring NFS storage, follow the recommendation of your storage vendor.
- Ensure that the NFS volume is exported using NFS over TCP.
- Make sure that the NFS server exports a particular share as either NFS 3 or NFS 4.1, but does not provide both protocol versions for the same share. This policy needs to be enforced by the server because ESXi does not prevent mounting the same share through different NFS versions.
- NFS 3 and non-Kerberos NFS 4.1 do not support the delegate user functionality that enables access to NFS volumes using nonroot credentials. If you use NFS 3 or non-Kerberos NFS 4.1, ensure that each host has root access to the volume. Different storage vendors have different methods of enabling this functionality, but typically this is done on the NAS servers by using the `no_root_squash` option. If the NAS server does not grant root access, you might still be able to mount the NFS datastore on the host. However, you will not be able to create any virtual machines on the datastore.
- If the underlying NFS volume, on which files are stored, is read-only, make sure that the volume is exported as a read-only share by the NFS server, or configure it as a read-only datastore on the ESXi host. Otherwise, the host considers the datastore to be read-write and might not be able to open the files.

NFS Networking Guidelines

- For network connectivity, the host requires a standard network adapter.
- ESXi supports Layer 2 and Layer 3 Network switches. If you use Layer 3 switches, ESXi hosts and NFS storage arrays must be on different subnets and the network switch must handle the routing information.
- A VMkernel port group is required for NFS storage. You can create a new VMkernel port group for IP storage on an already existing virtual switch (vSwitch) or on a new vSwitch when it is configured. The vSwitch can be a vSphere Standard Switch (VSS) or a vSphere Distributed Switch (VDS).
- If you use multiple ports for NFS traffic, make sure that you correctly configure your virtual switches and physical switches. For information, see the *vSphere Networking* documentation.
- NFS 3 and non-Kerberos NFS 4.1 support IPv4 and IPv6.

NFS Datastore Guidelines

- To use NFS 4.1, upgrade your vSphere environment to version 6.0. You cannot mount an NFS 4.1 datastore to hosts that do not support version 4.1.
- You cannot use different NFS versions to mount the same datastore. NFS 3 and NFS 4.1 clients do not use the same locking protocol. As a result, accessing the same virtual disks from two incompatible clients might result in incorrect behavior and cause data corruption.
- NFS 3 and NFS 4.1 datastores can coexist on the same host.
- vSphere does not support datastore upgrades from NFS version 3 to version 4.1.
- When you mount the same NFS 3 volume on different hosts, make sure that the server and folder names are identical across the hosts. If the names do not match, the hosts see the same NFS version 3 volume as two different datastores. This error might result in a failure of such features as vMotion. An example of such discrepancy is entering `filer` as the server name on one host and `filer.domain.com` on the other. This guideline does not apply to NFS version 4.1.
- If you use non-ASCII characters to name datastores and virtual machines, make sure that the underlying NFS server offers internationalization support. If the server does not support international characters, use only ASCII characters, or unpredictable failures might occur.

NFS Protocols and ESXi

ESXi supports NFS protocols version 3 and 4.1. To support both versions, ESXi uses two different NFS clients.

NFS Protocol Version 3

vSphere supports NFS version 3 in TCP. When you use this version, the following considerations apply:

- With NFS version 3, storage traffic is transmitted in an unencrypted format across the LAN. Because of this limited security, use NFS storage on trusted networks only and isolate the traffic on separate physical switches. You can also use a private VLAN.
- NFS 3 uses only one TCP connection for I/O. As a result, ESXi supports I/O on only one IP address or hostname for the NFS server, and does not support multiple paths. Depending on your network infrastructure and configuration, you can use network stack to configure multiple connections to the storage targets. In this case, you must have multiple datastores, with each datastore using separate network connections between the host and the storage.
- With NFS 3, ESXi does not support the delegate user functionality that enables access to NFS volumes by using nonroot credentials. You must ensure that each host has root access to the volume.
- NFS 3 supports hardware acceleration that allows your host to integrate with NAS devices and use several hardware operations that NAS storage provides. For more information, see [“Hardware Acceleration on NAS Devices,”](#) on page 249.
- When hardware acceleration is supported, you can create thick-provisioned virtual disk on NFS 3 datastores.
- NFS 3 locking on ESXi does not use the Network Lock Manager (NLM) protocol. Instead, VMware provides its own locking protocol. NFS 3 locks are implemented by creating lock files on the NFS server. Lock files are named `.lck-file_id..`

NFS Protocol Version 4.1

When you use NFS 4.1, the following considerations apply:

- NFS 4.1 provides multipathing for servers that support session trunking. When trunking is available, you can use multiple IP addresses to access a single NFS volume. Client ID trunking is not supported.

- NFS 4.1 does not support hardware acceleration. This limitation does not allow you to create thick virtual disks on NFS 4.1 datastores.
- NFS 4.1 supports the Kerberos authentication protocol to secure communication with the NFS server. For more information, see [“Using Kerberos Credentials for NFS 4.1,”](#) on page 158.
- NFS 4.1 uses share reservations as a locking mechanism.
- NFS 4.1 supports inbuilt file locking.
- NFS 4.1 supports nonroot users to access files when used with Kerberos.
- NFS 4.1 supports traditional non-Kerberos mounts. In this case, use security and root access guidelines recommended for NFS version 3.
- Does not support simultaneous AUTH_SYS and Kerberos mounts.
- NFS 4.1 with Kerberos does not support IPv6. NFS 4.1 with AUTH_SYS supports both IPv4 and IPv6.

NFS Protocols and vSphere Solutions

vSphere Features	NFS version 3	NFS version 4.1
vMotion and Storage vMotion	Yes	Yes
High Availability (HA)	Yes	Yes
Fault Tolerance (FT)	Yes	Yes
Distributed Resource Scheduler (DRS)	Yes	Yes
Host Profiles	Yes	Yes
Storage DRS	Yes	No
Storage I/O Control	Yes	No
Site Recovery Manager	Yes	No
Virtual Volumes	Yes	No

NFS Version Upgrades

vSphere does not support automatic datastore conversions from NFS version 3 to NFS 4.1. If you want to upgrade your NFS 3 datastore, the following options are available:

- You can create a new NFS 4.1 datastore, and then use Storage vMotion to migrate virtual machines from the old datastore to the new one.
- Use conversion methods provided by your NFS storage server. For more information, contact your storage vendor.
- Unmount from one version and then mount as the other.



CAUTION If you use this option, make sure to unmount the datastore from all hosts that have access to the datastore. The datastore can never be mounted by using both protocols at the same time.

Firewall Configurations for NFS Storage

ESXi includes a firewall between the management interface and the network. The firewall is enabled by default. At installation time, the ESXi firewall is configured to block incoming and outgoing traffic, except traffic for the default services, such as NFS.

Supported services, including NFS, are described in a rule set configuration file in the ESXi firewall directory `/etc/vmware/firewall/`. The file contains firewall rules and lists each rule's relationship with ports and protocols.

The behavior of the NFS Client rule set (`nfsClient`) is different from other rule sets. When the NFS Client rule set is enabled, all outbound TCP ports are open for the destination hosts in the list of allowed IP addresses.

The NFS 4.1 rule set opens outgoing connections to destination port 2049, which is the port named in the specification for version 4.1 protocol. The outgoing connections are open for all IP addresses at the time of the first mount. This port remains open until the ESXi host is rebooted.

For more information about firewall configurations, see the *vSphere Security* documentation.

NFS Client Firewall Behavior

The NFS Client firewall rule set behaves differently than other ESXi firewall rule sets. ESXi configures NFS Client settings when you mount or unmount an NFS datastore. The behavior differs for different versions of NFS.

When you add, mount, or unmount an NFS datastore, the resulting behavior depends on the version of NFS.

NFS v3 Firewall Behavior

When you add or mount an NFS v3 datastore, ESXi checks the state of the NFS Client (`nfsClient`) firewall rule set.

- If the `nfsClient` rule set is disabled, ESXi enables the rule set and disables the Allow All IP Addresses policy by setting the `allowedAll` flag to `FALSE`. The IP address of the NFS server is added to the allowed list of outgoing IP addresses.
- If the `nfsClient` rule set is enabled, the state of the rule set and the allowed IP address policy are not changed. The IP address of the NFS server is added to the allowed list of outgoing IP addresses.

NOTE If you manually enable the `nfsClient` rule set or manually set the Allow All IP Addresses policy, either before or after you add an NFS v3 datastore to the system, your settings are overridden when the last NFS v3 datastore is unmounted. The `nfsClient` rule set is disabled when all NFS v3 datastores are unmounted.

When you remove or unmount an NFS v3 datastore, ESXi performs one of the following actions.

- If none of the remaining NFS v3 datastores are mounted from the server of the datastore being unmounted, ESXi removes the server's IP address from the list of outgoing IP addresses.
- If no mounted NFS v3 datastores remain after the unmount operation, ESXi disables the `nfsClient` firewall rule set.

NFS v4.1 Firewall Behavior

When you mount the first NFS v4.1 datastore, ESXi enables the `nfs41client` rule set and sets its `allowedAll` flag to `TRUE`. This action opens port 2049 for all IP addresses. Unmounting an NFS v4.1 datastore does not affect the firewall state. That is, the first NFS v4.1 mount opens port 2049 and that port remains enabled unless you close it explicitly.

Verify Firewall Ports for NFS Clients

To enable access to NFS storage, ESXi automatically opens firewall ports for the NFS clients when you mount an NFS datastore. For troubleshooting reasons, you might need to verify that the ports are open.

Procedure

- 1 In the vSphere Web Client, select the ESXi host.
- 2 Click the **Manage** tab, and click **Settings**.
- 3 Select **Security Profile** in the **System** area, and click **Edit**.

- 4 Scroll down to an appropriate version of NFS to make sure that the port is opened.

Using Layer 3 Routed Connections to Access NFS Storage

When you use Layer 3 (L3) routed connections to access NFS storage, consider certain requirements and restrictions.

Ensure that your environment meets the following requirements:

- Use Cisco's Hot Standby Router Protocol (HSRP) in IP Router. If you are using non-Cisco router, be sure to use Virtual Router Redundancy Protocol (VRRP) instead.
- Use Quality of Service (QoS) to prioritize NFS L3 traffic on networks with limited bandwidths, or on networks that experience congestion. See your router documentation for details.
- Follow Routed NFS L3 best practices recommended by storage vendor. Contact your storage vendor for details.
- Disable Network I/O Resource Management (NetIORM).
- If you are planning to use systems with top-of-rack switches or switch-dependent I/O device partitioning, contact your system vendor for compatibility and support.

In an L3 environment the following restrictions apply:

- The environment does not support VMware Site Recovery Manager.
- The environment supports only the NFS protocol. Do not use other storage protocols such as FCoE over the same physical network.
- The NFS traffic in this environment does not support IPv6.
- The NFS traffic in this environment can be routed only over a LAN. Other environments such as WAN are not supported.

Set Up NFS Storage Environment

You must perform several configuration steps before you mount an NFS datastore in vSphere.

Prerequisites

- Familiarize yourself with the guidelines in [“NFS Storage Guidelines and Requirements,”](#) on page 153.
- For details on configuring NFS storage, consult your storage vendor documentation.

Procedure

- 1 On the NFS server, configure an NFS volume and export it to be mounted on the ESXi hosts.
 - a Note the IP address or the DNS name of the NFS server and the full path, or folder name, for the NFS share.

For NFS 4.1 you can collect multiple IP addresses or DNS names to take advantage of the multipathing support that an NFS 4.1 datastore provides. NFS 3 and non-Kerberos NFS 4.1 support IPv4 and IPv6 addresses.
 - b If you plan to use Kerberos authentication with NFS 4.1, specify the Kerberos credentials to be used by ESXi for authentication.

- 2 On each ESXi host, configure a VMkernel Network port for NFS traffic.

For more information, see the *vSphere Networking* documentation.

- 3 If you plan to use Kerberos authentication with the NFS 4.1 datastore, configure the ESXi hosts for Kerberos authentication.

Make sure that each host that mounts this datastore is a part of an Active Directory domain and its NFS authentication credentials are set.

What to do next

You can now create an NFS datastore on the ESXi hosts.

Using Kerberos Credentials for NFS 4.1

With NFS version 4.1, ESXi supports Kerberos authentication mechanism.

Kerberos is an authentication service that allows an NFS 4.1 client installed on ESXi to prove its identity to an NFS server before mounting an NFS share. Kerberos uses cryptography to work across an insecure network connection. The vSphere implementation of Kerberos for NFS 4.1 supports only identity verification for the client and server, but does not provide data integrity or confidentiality services.

When you use Kerberos authentication, the following considerations apply:

- ESXi uses Kerberos version 5 with Active Directory domain and Key Distribution Center (KDC).
- As a vSphere administrator, you specify Active Directory credentials to provide an access to NFS 4.1 Kerberos datastores to an NFS user. A single set of credentials is used to access all Kerberos datastores mounted on that host.
- When multiple ESXi hosts share the same NFS 4.1 datastore, you must use the same Active Directory credentials for all hosts that access the shared datastore. You can automate this by setting the user in host profiles and applying the profile to all ESXi hosts.
- NFS 4.1 does not support simultaneous AUTH_SYS and Kerberos mounts.
- NFS 4.1 with Kerberos does not support IPv6. Only IPv4 is supported.

Configure ESXi Hosts for Kerberos Authentication

If you use NFS 4.1 with Kerberos, you must perform several tasks to set up your hosts for Kerberos authentication.

When multiple ESXi hosts share the same NFS 4.1 datastore, you must use the same Active Directory credentials for all hosts that access the shared datastore. You can automate this by setting the user in host profiles and applying the profile to all ESXi hosts.

Prerequisites

- Make sure that Microsoft Active Directory (AD) and NFS servers are configured to use Kerberos.
- Enable DES-CBC-MD5 encryption mode on AD. The NFS 4.1 client supports only this encryption mode.
- Make sure that the NFS server exports are configured to grant full access to the Kerberos user.

Procedure

- 1 [Configure DNS for NFS 4.1 with Kerberos](#) on page 159

When you use NFS 4.1 with Kerberos, you must change the DNS settings on ESXi hosts to point to the DNS server that is configured to hand out DNS records for the Kerberos Key Distribution Center (KDC). For example, use the Active Directory server address, if AD is used as a DNS server.

- 2 [Configure Network Time Protocol for NFS 4.1 with Kerberos](#) on page 159

If you use NFS 4.1 with Kerberos, configure Network Time Protocol (NTP) to make sure all ESXi hosts on the vSphere network are synchronized.

3 [Enable Kerberos Authentication in Active Directory](#) on page 160

If you use NFS 4.1 storage with Kerberos, you must add each ESXi host to an Active Directory domain and enable Kerberos authentication. Kerberos integrates with Active Directory to enable single sign-on and provides an additional layer of security when used across an insecure network connection.

What to do next

After you configure your host for Kerberos, you can create an NFS 4.1 datastore with Kerberos enabled.

Configure DNS for NFS 4.1 with Kerberos

When you use NFS 4.1 with Kerberos, you must change the DNS settings on ESXi hosts to point to the DNS server that is configured to hand out DNS records for the Kerberos Key Distribution Center (KDC). For example, use the Active Directory server address, if AD is used as a DNS server.

Procedure

- 1 In the vSphere Web Client, navigate to the host.
- 2 Click the **Manage** tab, and click **Networking** and select **TCP/IP configuration**.
- 3 Select **TCP/IP configuration** and click the **Edit** icon.
- 4 Enter the DNS setting information.

Option	Description
Domain	<i>AD Domain Name</i>
Preferred DNS server	<i>AD Server IP</i>
Search domains	<i>AD Domain Name</i>

Configure Network Time Protocol for NFS 4.1 with Kerberos

If you use NFS 4.1 with Kerberos, configure Network Time Protocol (NTP) to make sure all ESXi hosts on the vSphere network are synchronized.

Procedure

- 1 Select the host in the vSphere inventory.
- 2 Click the **Manage** tab and click **Settings**.
- 3 In the System section, select **Time Configuration**.
- 4 Click **Edit** and set up the NTP server.
 - a Select **Use Network Time Protocol (Enable NTP client)**.
 - b Set the NTP Service Startup Policy.
 - c Enter the IP addresses of the NTP servers to synchronize with.
 - d Click **Start** or **Restart** in the NTP Service Status section.
- 5 Click **OK**.

The host synchronizes with the NTP server.

Enable Kerberos Authentication in Active Directory

If you use NFS 4.1 storage with Kerberos, you must add each ESXi host to an Active Directory domain and enable Kerberos authentication. Kerberos integrates with Active Directory to enable single sign-on and provides an additional layer of security when used across an insecure network connection.

Prerequisites

Set up an AD domain and a domain administrator account with the rights to add hosts to the domain.

Procedure

- 1 Add an ESXi host to an Active Directory domain.
 - a In the vSphere Web Client, select the ESXi host.
 - b Click the **Manage** tab and click **Settings**.
 - c Under System, select **Authentication Services**.
 - d Click **Join Domain**, supply the domain settings, and click **OK**.

The directory services type changes to Active Directory.

- 2 Configure or edit credentials for an NFS Kerberos user.
 - a Under NFS Kerberos Credentials, click **Edit**.
 - b Enter a user name and password.

Files stored in all Kerberos datastores will be accessed using these credentials.

The state for NFS Kerberos credentials changes to Enabled.

Creating Datastores

You use the New Datastore wizard to create your datastores. Depending on the type of storage you have in your environment and your storage needs, you can create a VMFS, NFS, or virtual datastore.

A Virtual SAN datastore is automatically created when you enable Virtual SAN. For information, see the *Administering VMware Virtual SAN* documentation.

You can also use the New Datastore wizard to manage VMFS datastore copies.

- [Create a VMFS Datastore](#) on page 160
VMFS datastores serve as repositories for virtual machines. You can set up VMFS datastores on any SCSI-based storage devices that the host discovers, including Fibre Channel, iSCSI, and local storage devices.
- [Create an NFS Datastore](#) on page 161
You can use the New Datastore wizard to mount an NFS volume.
- [Create a Virtual Datastore](#) on page 162
You use the New Datastore wizard to create a virtual datastore.

Create a VMFS Datastore

VMFS datastores serve as repositories for virtual machines. You can set up VMFS datastores on any SCSI-based storage devices that the host discovers, including Fibre Channel, iSCSI, and local storage devices.

NOTE You cannot create VMFS3 datastores in vSphere 6.0. Even though existing VMFS3 datastores continue to be available and usable, you must upgrade them to VMFS5.

Prerequisites

Install and configure any adapters that your storage requires. Rescan the adapters to discover newly added storage devices.

Procedure

- 1 In the vSphere Web Client navigator, select **vCenter Inventory Lists > Datastores**
- 2 Click the Create a New Datastore icon.
- 3 Type the datastore name and if required, select the placement location for the datastore.
The vSphere Web Client enforces a 42 character limit for the datastore name.
- 4 Select VMFS as the datastore type.
- 5 Select the device to use for your datastore.

IMPORTANT The device you select must not have any values displayed in the Snapshot Volume column. If a value is present, the device contains a copy of an existing VMFS datastore. For information on managing datastore copies, see [“Managing Duplicate VMFS Datastores,”](#) on page 162.

- 6 Specify partition configuration.

Option	Description
Use all available partitions	Dedicates the entire disk to a single VMFS datastore. If you select this option, all file systems and data currently stored on this device are destroyed.
Use free space	Deploys a VMFS datastore in the remaining free space of the disk.

- 7 (Optional) If the space allocated for the datastore is excessive for your purposes, adjust the capacity values in the Datastore Size field.

By default, the entire free space on the storage device is allocated.

- 8 In the Ready to Complete page, review the datastore configuration information and click **Finish**.

The datastore on the SCSI-based storage device is created. It is available to all hosts that have access to the device.

Create an NFS Datastore

You can use the New Datastore wizard to mount an NFS volume.

Prerequisites

- Set up NFS storage environment.
- If you plan to use Kerberos authentication with the NFS 4.1 datastore, make sure to configure the ESXi hosts for Kerberos authentication.

Procedure

- 1 In the vSphere Web Client navigator, select **vCenter Inventory Lists > Datastores**
- 2 Click the Create a New Datastore icon.
- 3 Type the datastore name and if required, select the placement location for the datastore.
The vSphere Web Client enforces a 42 character limit for the datastore name.
- 4 Select NFS as the datastore type.

- 5 Specify an NFS version.
 - NFS 3
 - NFS 4.1

IMPORTANT If multiple hosts access the same datastore, you must use the same protocol on all hosts.

- 6 Type the server name or IP address and the mount point folder name.
 With NFS 4.1, you can add multiple IP addresses or server names if the server supports trunking. The host uses these values to achieve multipathing to the NFS server mount point.
 You can use IPv4 or IPv6 addresses for NFS 3 and non-Kerberos NFS 4.1.
- 7 Select **Mount NFS read only** if the volume is exported as read-only by the NFS server.
- 8 If you use Kerberos authentication with NFS 4.1, enable Kerberos on the datastore.
- 9 If you are creating a datastore at the data center or cluster level, select hosts that mount the datastore.
- 10 Review the configuration options and click **Finish**.

Create a Virtual Datastore

You use the New Datastore wizard to create a virtual datastore.

Procedure

- 1 In the vSphere Web Client navigator, select **vCenter Inventory Lists > Datastores**
- 2 Click the Create a New Datastore icon.
- 3 Type the datastore name and if required, select the placement location for the datastore.
 Make sure to use the name that does not duplicate another datastore name in your data center environment.
 If you mount the same virtual datastore to several hosts, the name of the datastore must be consistent across all hosts.
- 4 Select **VVOL** as the datastore type.
- 5 From the list of storage containers, select a backing storage container.
- 6 Select the hosts that require access to the datastore.
- 7 Review the configuration options and click **Finish**.

What to do next

After you create the virtual datastore, you can perform such datastore operations as renaming the datastore, browsing datastore files, unmounting the datastore, and so on.

You cannot add the virtual datastore to a datastore cluster.

Managing Duplicate VMFS Datastores

When a storage device contains a VMFS datastore copy, you can mount the datastore with the existing signature or assign a new signature.

Each VMFS datastore created in a storage disk has a unique signature, also called UUID, that is stored in the file system superblock. When the storage disk is replicated or its snapshot is taken on the storage side, the resulting disk copy is identical, byte-for-byte, with the original disk. As a result, if the original storage disk contains a VMFS datastore with UUID X, the disk copy appears to contain an identical VMFS datastore, or a VMFS datastore copy, with exactly the same UUID X.

In addition to LUN snapshotting and replication, the following storage device operations might cause ESXi to mark the existing datastore on the device as a copy of the original datastore:

- LUN ID changes
- SCSI device type changes, for example, from SCSI-2 to SCSI-3
- SPC-2 compliancy enablement

ESXi can detect the VMFS datastore copy and display it in the vSphere Web Client. You have an option of mounting the datastore copy with its original UUID or changing the UUID to resignature the datastore.

Whether you chose resignaturing or mounting without resignaturing depends on how the LUNs are masked in the storage environment. If your hosts are able to see both copies of the LUN, then resignaturing is the recommended method. Otherwise, mounting is an option.

Keep Existing Datastore Signature

If you do not need to resignature a VMFS datastore copy, you can mount it without changing its signature.

You can keep the signature if, for example, you maintain synchronized copies of virtual machines at a secondary site as part of a disaster recovery plan. In the event of a disaster at the primary site, you mount the datastore copy and power on the virtual machines at the secondary site.

Prerequisites

- Perform a storage rescan on your host to update the view of storage devices presented to the host.
- Unmount the original VMFS datastore that has the same UUID as the copy you plan to mount. You can mount the VMFS datastore copy only if it does not collide with the original VMFS datastore.

Procedure

- 1 In the vSphere Web Client navigator, select **vCenter Inventory Lists > Datastores**
- 2 Click the Create a New Datastore icon.
- 3 Type the datastore name and if required, select the placement location for the datastore.
- 4 Select VMFS as the datastore type.
- 5 From the list of storage devices, select the device that has a specific value displayed in the Snapshot Volume column.

The value present in the Snapshot Volume column indicates that the device is a copy that contains a copy of an existing VMFS datastore.

- 6 Under Mount Options, select **Keep Existing Signature**.
- 7 Review the datastore configuration information and click **Finish**.

What to do next

If you later want to resignature the mounted datastore, you must unmount it first.

Resignature a VMFS Datastore Copy

Use datastore resignaturing if you want to retain the data stored on the VMFS datastore copy.

When resignaturing a VMFS copy, ESXi assigns a new signature (UUID) to the copy, and mounts the copy as a datastore distinct from the original. All references to the original signature from virtual machine configuration files are updated.

When you perform datastore resignaturing, consider the following points:

- Datastore resignaturing is irreversible.

- After resignaturing, the storage device replica that contained the VMFS copy is no longer treated as a replica.
- A spanned datastore can be resignatured only if all its extents are online.
- The resignaturing process is crash and fault tolerant. If the process is interrupted, you can resume it later.
- You can mount the new VMFS datastore without a risk of its UUID conflicting with UUIDs of any other datastore, such as an ancestor or child in a hierarchy of storage device snapshots.

Prerequisites

- Unmount the datastore copy.
- Perform a storage rescan on your host to update the view of storage devices presented to the host.

Procedure

- 1 In the vSphere Web Client navigator, select **vCenter Inventory Lists > Datastores**
- 2 Click the Create a New Datastore icon.
- 3 Type the datastore name and if required, select the placement location for the datastore.
- 4 Select VMFS as the datastore type.
- 5 From the list of storage devices, select the device that has a specific value displayed in the Snapshot Volume column.

The value present in the Snapshot Volume column indicates that the device is a copy that contains a copy of an existing VMFS datastore.
- 6 Under Mount Options, select **Assign a New Signature** and click **Next**.
- 7 Review the datastore configuration information and click **Finish**.

Upgrading VMFS Datastores

If your datastores were formatted with VMFS2 or VMFS3, you must upgrade the datastores to VMFS5.

When you perform datastore upgrades, consider the following items:

- To upgrade a VMFS2 datastore, you use a two-step process that involves upgrading VMFS2 to VMFS3 first. To access the VMFS2 datastore and perform the VMFS2 to VMFS3 conversion, use an ESX/ESXi 4.x or earlier host.

After you upgrade your VMFS2 datastore to VMFS3, the datastore becomes available on the ESXi 6.0 host, where you complete the process of upgrading to VMFS5.
- You can perform a VMFS3 to VMFS5 upgrade while the datastore is in use with virtual machines powered on.
- While performing an upgrade, your host preserves all files on the datastore.
- The datastore upgrade is a one-way process. After upgrading your datastore, you cannot revert it back to its previous VMFS format.

An upgraded VMFS5 datastore differs from a newly formatted VMFS5.

Table 16-3. Comparing Upgraded and Newly Formatted VMFS5 Datastores

Characteristics	Upgraded VMFS5	Formatted VMFS5
File block size	1, 2, 4, and 8MB	1MB
Subblock size	64KB	8KB

Table 16-3. Comparing Upgraded and Newly Formatted VMFS5 Datastores (Continued)

Characteristics	Upgraded VMFS5	Formatted VMFS5
Partition format	MBR. Conversion to GPT happens only after you expand the datastore to a size larger than 2TB.	GPT
Datastore limits	Retains limits of VMFS3 datastore.	
VMFS locking mechanism	ATS+SCSI	ATS-only (on hardware that supports ATS) ATS+SCSI (on hardware that does not support ATS)

For more information about VMFS locking mechanisms and how to upgrade to ATS-only, see [“VMFS Locking Mechanisms,”](#) on page 149.

Upgrade a Datastore to VMFS5

If you use VMFS3 datastores, you must upgrade them to VMFS5.

You can perform an upgrade while the datastore is in use with virtual machines powered on.

Prerequisites

- If you have a VMFS2 datastore, you must first upgrade it to VMFS3 using an ESX/ESXi 3.x or ESX/ESXi 4.x host. Use the vSphere Host Client to access the host.
- All hosts accessing the datastore must support VMFS5.
- Verify that the volume to be upgraded has at least 2MB of free blocks available and 1 free file descriptor.

Procedure

- 1 In the vSphere Web Client navigator, select **vCenter Inventory Lists > Datastores**
- 2 Click the datastore to upgrade.
- 3 Click the **Manage** tab, and click **Settings**.
- 4 Click **Upgrade to VMFS5**.
- 5 Verify that the hosts accessing the datastore support VMFS5.
- 6 Click **OK** to start the upgrade.
- 7 Perform a rescan on all hosts that are associated with the datastore.

The datastore is upgraded to VMFS5 and is available to all hosts that are associated with the datastore.

Increasing VMFS Datastore Capacity

If your VMFS datastore requires more space, increase the datastore capacity. You can dynamically increase the capacity by growing a datastore extent or by adding a new extent.

Use one of the following methods to increase a datastore capacity:

- Dynamically grow any expandable datastore extent, so that it fills the available adjacent capacity. The extent is considered expandable when the underlying storage device has free space immediately after the extent.

- Dynamically add a new extent. The datastore can span over up to 32 extents with the size of each extent of more than 2TB, yet appear as a single volume. The spanned VMFS datastore can use any or all of its extents at any time. It does not need to fill up a particular extent before using the next one.

NOTE Datastores that only support hardware assisted locking, also called atomic test and set (ATS) mechanism cannot span over non-ATS devices. For more information, see [“VMFS Locking Mechanisms,”](#) on page 149.

Increase VMFS Datastore Capacity

When you need to add virtual machines to a datastore, or when the virtual machines running on a datastore require more space, you can dynamically increase the capacity of a VMFS datastore.

If a shared datastore has powered on virtual machines and becomes 100% full, you can increase the datastore's capacity only from the host with which the powered on virtual machines are registered.

Procedure

- 1 In the vSphere Web Client navigator, select **vCenter Inventory Lists > Datastores**
- 2 Select the datastore to grow and click the Increase Datastore Capacity icon.
- 3 Select a device from the list of storage devices.

Your selection depends on whether an expandable storage device is available.

Option	Description
To expand an existing extent	Select the device for which the Expandable column reads YES. A storage device is reported as expandable when it has free space immediately after the extent.
To add a new extent	Select the device for which the Expandable column reads NO.

- 4 Review the **Current Disk Layout** to see the available configurations and click **Next**.
- 5 Select a configuration option from the bottom panel.

Depending on the current layout of the disk and on your previous selections, the options you see might vary.

Option	Description
Use free space to add new extent	Adds the free space on this disk as a new extent.
Use free space to expand existing extent	Expands an existing extent to a required capacity.
Use free space	Deploys an extent in the remaining free space of the disk. This option is available only when you are adding an extent.
Use all available partitions	Dedicates the entire disk to a single extent. This option is available only when you are adding an extent and when the disk you are formatting is not blank. The disk is reformatted, and the datastores and any data that it contains are erased.

- 6 Set the capacity for the extent.
The minimum extent size is 1.3GB. By default, the entire free space on the storage device is available.
- 7 Click **Next**.
- 8 Review the proposed layout and the new configuration of your datastore, and click **Finish**.

Administrative Operations for Datastores

After creating datastores, you can perform several administrative operations on the datastores. Certain operations, such as renaming a datastore, are available for all types of datastores. Others apply to specific types of datastores.

- [Change Datastore Name](#) on page 167
You can change the name of an existing datastore.
- [Unmount Datastores](#) on page 168
When you unmount a datastore, it remains intact, but can no longer be seen from the hosts that you specify. The datastore continues to appear on other hosts, where it remains mounted.
- [Mount Datastores](#) on page 168
You can mount a datastore you previously unmounted. You can also mount a datastore on additional hosts.
- [Remove VMFS Datastores](#) on page 169
You can delete any type of VMFS datastore, including copies that you have mounted without resignaturing. When you delete a datastore, it is destroyed and disappears from all hosts that have access to the datastore.
- [Use Datastore Browser](#) on page 169
Use the datastore file browser to manage contents of your datastores.
- [Turn off Storage Filters](#) on page 172
When you perform VMFS datastore management operations, vCenter Server uses default storage protection filters. The filters help you to avoid storage corruption by retrieving only the storage devices that can be used for a particular operation. Unsuitable devices are not displayed for selection. You can turn off the filters to view all devices.

Change Datastore Name

You can change the name of an existing datastore.

Procedure

- 1 In the vSphere Web Client navigator, select **vCenter Inventory Lists > Datastores**
- 2 Select the datastore to rename.
- 3 Select **Rename** from the right-click menu.
- 4 Type a new datastore name.

The vSphere Web Client enforces a 42 character limit for the datastore name.

The new name appears on all hosts that have access to the datastore.

Unmount Datastores

When you unmount a datastore, it remains intact, but can no longer be seen from the hosts that you specify. The datastore continues to appear on other hosts, where it remains mounted.

Do not perform any configuration operations that might result in I/O to the datastore while the unmount is in progress.

NOTE Make sure that the datastore is not used by vSphere HA heartbeating. vSphere HA heartbeating does not prevent you from unmounting the datastore. However, if the datastore is used for heartbeating, unmounting it might cause the host to fail and restart any active virtual machine.

Prerequisites

When appropriate, before unmounting datastores, make sure that the following prerequisites are met:

- No virtual machines reside on the datastore.
- The datastore is not managed by Storage DRS.
- Storage I/O control is disabled for this datastore.

Procedure

- 1 In the vSphere Web Client navigator, select **vCenter Inventory Lists > Datastores**
- 2 Right-click the datastore to unmount and select **Unmount Datastore**.
- 3 If the datastore is shared, specify which hosts should no longer access the datastore.
- 4 Confirm that you want to unmount the datastore.

After you unmount a VMFS datastore from all hosts, the datastore is marked as inactive. If you unmount an NFS or a virtual datastore from all hosts, the datastore disappears from the inventory.

What to do next

If you unmounted the VMFS datastore as a part of an orderly storage removal procedure, you can now detach the storage device that is backing the datastore. See [“Detach Storage Devices,”](#) on page 128.

Mount Datastores

You can mount a datastore you previously unmounted. You can also mount a datastore on additional hosts.

A VMFS datastore that has been unmounted from all hosts remains in inventory, but is marked as inaccessible. You can use this task to mount the VMFS datastore to a specified host or multiple hosts.

If you have unmounted an NFS or a virtual datastore from all hosts, the datastore disappears from the inventory. To mount the NFS or virtual datastore that has been removed from the inventory, use the New Datastore wizard.

A datastore of any type that is unmounted from some hosts while being mounted on others, is shown as active in the inventory.

Procedure

- 1 In the vSphere Web Client navigator, select **vCenter Inventory Lists > Datastores**

- 2 Right-click the datastore to mount and select one of the following options:
 - **Mount Datastore**
 - **Mount Datastore on Additional Hosts**

Whether you see one or another option depends on the type of datastore you use.

- 3 Select the hosts that should access the datastore.

Remove VMFS Datastores

You can delete any type of VMFS datastore, including copies that you have mounted without resignaturing. When you delete a datastore, it is destroyed and disappears from all hosts that have access to the datastore.

NOTE The datastore delete operation permanently deletes all files associated with virtual machines on the datastore. Although you can delete the datastore without unmounting, it is preferable that you unmount the datastore first.

Prerequisites

- Remove or migrate all virtual machines from the datastore.
- Make sure that no other host is accessing the datastore.
- Disable Storage DRS for the datastore.
- Disable Storage I/O control for the datastore.
- Make sure that the datastore is not used for vSphere HA heartbeating.

Procedure

- 1 In the vSphere Web Client navigator, select **vCenter Inventory Lists > Datastores**
- 2 Right-click the datastore to remove.
- 3 Select **Delete Datastore**.
- 4 Confirm that you want to remove the datastore.

Use Datastore Browser

Use the datastore file browser to manage contents of your datastores.

You can perform a number of tasks that include the following.

- [Upload Files to Datastores](#) on page 170
Use the datastore file browser to upload files to datastores accessible to ESXi hosts.
- [Copy Datastore Folders or Files](#) on page 170
Use the datastore browser to copy folders or files to a new location, either on the same datastore or on a different datastore.
- [Move Datastore Folders or Files](#) on page 171
Use the datastore browser to move folders or files to a new location, either on the same datastore or on a different datastore.
- [Rename Datastore Folders or Files](#) on page 171
Use the datastore browser to rename folders or files.
- [Inflate Thin Virtual Disks](#) on page 171
If you created a virtual disk in the thin format, you can convert the thin disk to a virtual disk in thick provision format.

Upload Files to Datastores

Use the datastore file browser to upload files to datastores accessible to ESXi hosts.

In addition to their traditional use as a storage for virtual machines files, datastores can serve to store data or files related to virtual machines. For example, you can upload ISO images of operating systems from a local computer to a datastore on the host. You then use these images to install guest operating systems on the new virtual machines.

NOTE Virtual Volumes do not support uploading files directly to the virtual datastores. You must first create a folder on the virtual datastore, and then upload the files into the folder.

Prerequisites

Required privilege: **Datastore.Browse Datastore**

Procedure

- 1 In the vSphere Web Client navigator, select **vCenter Inventory Lists > Datastores**
- 2 Select the datastore where you want to store the file and click the **Navigate to the datastore file browser** icon.
- 3 (Optional) Create a new folder to store the file.
- 4 Select the target folder and click the **Upload the file to the datastore** icon.
- 5 Locate the item to upload on the local computer and click **Open**.
- 6 Refresh the datastore file browser to see the uploaded file on the list.

What to do next

You might experience problems when deploying an OVF template that you previously exported and then uploaded to a datastore. For details and a workaround, see the VMware Knowledge Base article [2117310](#).

Copy Datastore Folders or Files

Use the datastore browser to copy folders or files to a new location, either on the same datastore or on a different datastore.

NOTE Virtual disk files are moved or copied without format conversion. If you move a virtual disk to a datastore on a different type of host from the source host, you might need to convert the virtual disks before you can use them.

Prerequisites

Required privilege: **Datastore.Browse Datastore**

Procedure

- 1 Open the datastore browser.
 - a Display the datastore in the inventory.
 - b Right-click the datastore and select **Browse Files**.
- 2 Browse to an object you want to copy, either a folder or a file.
- 3 Select the object and click the **Copy selection to a new location** icon.
- 4 Specify the destination location.
- 5 (Optional) Select **Overwrite files and folders with matching names at the destination**.

- 6 Click **OK**.

Move Datastore Folders or Files

Use the datastore browser to move folders or files to a new location, either on the same datastore or on a different datastore.

NOTE Virtual disk files are moved or copied without format conversion. If you move a virtual disk to a datastore on a different type of host from the source host, you might need to convert the virtual disks before you can use them.

Prerequisites

Required privilege: **Datastore.Browse Datastore**

Procedure

- 1 Open the datastore browser.
 - a Display the datastore in the inventory.
 - b Right-click the datastore and select **Browse Files**.
- 2 Browse to an object you want to move, either a folder or a file.
- 3 Select the object and click the **Move selection to a new location** icon.
- 4 Specify the destination location.
- 5 (Optional) Select **Overwrite files and folders with matching names at the destination**.
- 6 Click **OK**.

Rename Datastore Folders or Files

Use the datastore browser to rename folders or files.

Prerequisites

Required privilege: **Datastore.Browse Datastore**

Procedure

- 1 Open the datastore browser.
 - a Display the datastore in the inventory.
 - b Right-click the datastore and select **Browse Files**.
- 2 Browse to an object you want to rename, either a folder or a file.
- 3 Select the object and click the **Rename selection** icon.
- 4 Specify the new name and click **OK**.

Inflate Thin Virtual Disks

If you created a virtual disk in the thin format, you can convert the thin disk to a virtual disk in thick provision format.

You use the datastore browser to inflate the virtual disk.

Procedure

- 1 Navigate to the folder of the virtual disk you want to inflate.
 - a In the vSphere Web Client, browse to the virtual machine.
 - b Double-click the virtual machine to display its information.
 - c Click the **Related Objects** tab and click **Datastores**.
The datastore that stores the virtual machine files is listed.
 - d Select the datastore and click the **Navigate to the datastore file browser** icon.
The datastore browser displays contents of the datastore.
- 2 Open the virtual machine folder and browse to the virtual disk file that you want to convert.
The file has the .vmdk extension and is marked with the virtual disk (🗄️) icon.
- 3 Right-click the virtual disk file and select **Inflate**.
The inflated virtual disk occupies the entire datastore space originally provisioned to it.

Turn off Storage Filters

When you perform VMFS datastore management operations, vCenter Server uses default storage protection filters. The filters help you to avoid storage corruption by retrieving only the storage devices that can be used for a particular operation. Unsuitable devices are not displayed for selection. You can turn off the filters to view all devices.

Prerequisites

Before making any changes to the device filters, consult with the VMware support team. You can turn off the filters only if you have other methods to prevent device corruption.

Procedure

- 1 Browse to the vCenter Server in the vSphere Web Client object navigator.
- 2 Click the **Manage** tab, and click **Settings**.
- 3 Click **Advanced Settings**, and click **Edit**.
- 4 In the Key text box, type an appropriate key.

Key	Filter Name
<code>config.vpxd.filter.vmfsFilter</code>	VMFS Filter
<code>config.vpxd.filter.rdmFilter</code>	RDM Filter
<code>config.vpxd.filter.SameHostAndTransportsFilter</code>	Same Host and Transports Filter
<code>config.vpxd.filter.hostRescanFilter</code>	Host Rescan Filter
	NOTE If you turn off the Host Rescan Filter, your hosts continue to perform a rescan each time you present a new LUN to a host or a cluster.

- 5 In the Value text box, type **False** for the specified key.
- 6 Click **Add** and click **OK**.
You are not required to restart the vCenter Server system.

Storage Filtering

vCenter Server provides storage filters to help you avoid storage device corruption or performance degradation that can be caused by an unsupported use of storage devices. These filters are available by default.

Table 16-4. Storage Filters

Filter Name	Description	Key
VMFS Filter	Filters out storage devices, or LUNs, that are already used by a VMFS datastore on any host managed by vCenter Server. The LUNs do not show up as candidates to be formatted with another VMFS datastore or to be used as an RDM.	config.vpxd.filter.vmfsFilter
RDM Filter	Filters out LUNs that are already referenced by an RDM on any host managed by vCenter Server. The LUNs do not show up as candidates to be formatted with VMFS or to be used by a different RDM. If you need virtual machines to access the same LUN, the virtual machines must share the same RDM mapping file. For information about this type of configuration, see the <i>vSphere Resource Management</i> documentation.	config.vpxd.filter.rdmFilter
Same Host and Transports Filter	Filters out LUNs ineligible for use as VMFS datastore extents because of host or storage type incompatibility. Prevents you from adding the following LUNs as extents: <ul style="list-style-type: none"> ■ LUNs not exposed to all hosts that share the original VMFS datastore. ■ LUNs that use a storage type different from the one the original VMFS datastore uses. For example, you cannot add a Fibre Channel extent to a VMFS datastore on a local storage device. 	config.vpxd.filter.SameHostAndTransportsFilter
Host Rescan Filter	Automatically rescans and updates VMFS datastores after you perform datastore management operations. The filter helps provide a consistent view of all VMFS datastores on all hosts managed by vCenter Server. NOTE If you present a new LUN to a host or a cluster, the hosts automatically perform a rescan no matter whether you have the Host Rescan Filter on or off.	config.vpxd.filter.hostRescanFilter

Set Up Dynamic Disk Mirroring

Typically, you cannot use logical-volume manager software on virtual machines to mirror virtual disks. However, if your Microsoft Windows virtual machines support dynamic disks, you can protect the virtual machines from an unplanned storage device loss by mirroring virtual disks across two SAN LUNs.

Prerequisites

- Use a Windows virtual machine that supports dynamic disks.
- Required privilege: **Advanced**

Procedure

- 1 Create a virtual machine with two virtual disks.
Make sure to place the disks on different datastores.

- 2 Log in to your virtual machine and configure the disks as dynamic mirrored disks.
See Microsoft documentation.
- 3 After the disks synchronise, power off the virtual machine.
- 4 Change virtual machine settings to allow the use of dynamic disk mirroring.
 - a Right-click the virtual machine and select **Edit Settings**.
 - b Click the **VM Options** tab and expand the **Advanced** menu.
 - c Click **Edit Configuration** next to Configuration Parameters.
 - d Click **Add Row** and add the following parameters:

Name	Value
scsi#.returnNoConnectDuringAPD	True
scsi#.returnBusyOnNoConnectStatus	False

- e Click **OK**.

Collecting Diagnostic Information for ESXi Hosts on a Storage Device

During a host failure, ESXi must be able to save diagnostic information to a preconfigured location for diagnostic and technical support purposes.

Typically, a partition to collect diagnostic information, also called VMkernel core dump, is created on a local storage device during ESXi installation. You can override this default behavior if, for example, you use shared storage devices instead of local storage. To prevent automatic formatting of local devices, detach the devices from the host before you install ESXi and power on the host for the first time. You can later set up a location for collecting diagnostic information on a local or remote storage device.

When you use storage devices, you can select between two options of setting up core dump collection. You can use a preconfigured diagnostic partition on a storage device or use a file on a VMFS datastore.

- [Set Up a Device Partition as Core Dump Location](#) on page 174
Create a diagnostic partition for your ESXi host.
- [Set Up a File as Core Dump Location](#) on page 175
If the size of your available core dump partition is insufficient, you can configure ESXi to generate core dump as a file.

Set Up a Device Partition as Core Dump Location

Create a diagnostic partition for your ESXi host.

When you create a diagnostic partition, the following considerations apply:

- You cannot create a diagnostic partition on an iSCSI LUN accessed through the software iSCSI or dependent hardware iSCSI adapter. For more information about diagnostic partitions with iSCSI, see [“General Boot from iSCSI SAN Recommendations,”](#) on page 107.
- You cannot create a diagnostic partition on a software FCoE LUN.
- Unless you are using diskless servers, set up a diagnostic partition on a local storage.
- Each host must have a diagnostic partition of 2.5 GB. If multiple hosts share a diagnostic partition on a SAN LUN, the partition should be large enough to accommodate core dumps of all hosts.
- If a host that uses a shared diagnostic partition fails, reboot the host and extract log files immediately after the failure. Otherwise, the second host that fails before you collect the diagnostic data of the first host might not be able to save the core dump.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Right-click the host, and select **Add Diagnostic Partition**.
If you do not see this option, the host already has a diagnostic partition.
- 3 Specify the type of diagnostic partition.

Option	Description
Private local	Creates the diagnostic partition on a local disk. This partition stores fault information only for your host.
Private SAN storage	Creates the diagnostic partition on a non-shared SAN LUN. This partition stores fault information only for your host.
Shared SAN storage	Creates the diagnostic partition on a shared SAN LUN. This partition is accessed by multiple hosts and can store fault information for more than one host.

- 4 Click **Next**.
- 5 Select the device to use for the diagnostic partition and click **Next**.
- 6 Review the partition configuration information and click **Finish**.

Verify a Diagnostic Partition

Use the `esxcli` command to verify whether a diagnostic partition is set.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ List partitions to verify that a diagnostic partition is set.

```
esxcli --server=server_name system coredump partition list
```

If a diagnostic partition is set, the command displays information about it. Otherwise, the command shows that no partition is activated and configured.

What to do next

To manage the host's diagnostic partition, use the vCLI commands. See *vSphere Command-Line Interface Concepts and Examples*.

Set Up a File as Core Dump Location

If the size of your available core dump partition is insufficient, you can configure ESXi to generate core dump as a file.

Typically, a core dump partition of 2.5 GB is created during ESXi installation. For upgrades from ESXi 5.0 and earlier, the core dump partition is limited to 100 MB. For this type of upgrade, during the boot process the system might create a core dump file on a VMFS datastore. If it does not create a core dump file, you can manually create the file.

NOTE Software iSCSI and software FCoE are not supported for core dump file locations.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 Create a VMFS datastore core dump file by running the following command:

```
esxcli system coredump file add
```

The command takes the following options, but they are not required and can be omitted:

Option	Description
--datastore -d <i>datastore_UUID or datastore_name</i>	If not provided, the system selects a datastore of sufficient size.
--file -f <i>file_name</i>	If not provided, the system specifies a unique name for the core dump file.
--size -s <i>file_size_MB</i>	If not provided, the system creates a file of the size appropriate for the memory installed in the host.

- 2 Verify that the file has been created:

```
esxcli system coredump file list
```

You can see the output similar to the following:

Path	Active	Configured	Size
/vmfs/volumes/52b021c3-.../vmkdump/test.dumpfile	false	false	104857600

- 3 Activate the core dump file for the host:

```
esxcli system coredump file set
```

The command takes the following options:

Option	Description
--path -p	The path of the core dump file to use. This must be a pre-allocated file.
--smart -s	This flag can be used only with <code>--enable -e=true</code> . It will cause the file to be selected using the smart selection algorithm. For example, esxcli system coredump file set --smart --enable true

- 4 Verify that the core dump file is active and configured:

```
esxcli system coredump file list
```

The output similar to the following indicates that the core dump file is active and configured:

Path	Active	Configured	Size
/vmfs/volumes/52b021c3-.../vmkdump/test.dumpfile	True	True	104857600

What to do next

For information about other commands you can use to manage the core dump files, see the *vSphere Command-Line Interface Reference* documentation.

Deactivate and Delete a Core Dump File

Deactivate a configured core dump file and, if needed, remove it from the VMFS datastore.

You can temporarily deactivate the core dump file. If you do not plan to use the deactivated file, you can remove it from the VMFS datastore. To remove the file that has not been deactivated, you can use the `system coredump file remove` with the `--force | -F` option.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 Deactivate the core dump file by running the following command:

```
esxcli system coredump file set --unconfigure | -u
```

- 2 Remove the file from the VMFS datastore:

```
system coredump file remove --file | -f file_name
```

The command takes the following options:

Option	Description
<code>--file -f</code>	Specify the file name of the dump file to be removed. If you do not specify the file name, the configured core dump file will be removed.
<code>--force -F</code>	Deactivate and unconfigure the dump file being removed. This option is required if the file has not been previously deactivated and is active.

The core dump file becomes disabled and is removed from the VMFS datastore.

Checking Metadata Consistency with VOMA

Use vSphere On-disk Metadata Analyser (VOMA) to identify and fix incidents of metadata corruption that affect file systems or underlying logical volumes.

Problem

You might need to check metadata consistency of a file system or a logical volume backing the file system when you experience problems with various functionalities on a VMFS datastore or a virtual flash resource. For example you might want to perform a metadata check if one of the following occurs:

- You experience storage outages.
- After you rebuild RAID or perform a disk replacement.
- You see metadata errors in the `vmkernel.log` file.
- You are unable to access files on a VMFS.
- You see corruption being reported for a datastore in events tabs of vCenter Server.

Solution

To check metadata consistency, run VOMA from the CLI of an ESXi host. VOMA can be used to check and fix metadata inconsistency issues for a VMFS datastore or a virtual flash resource. To resolve errors reported by VOMA, consult VMware Support.

Follow these guidelines when you use the VOMA tool:

- Make sure that the VMFS datastore you analyze does not span multiple extents. You can run VOMA only against a single-extent datastore.

- Power off any virtual machines that are running or migrate them to a different datastore.

The following example demonstrates how to use VOMA to check VMFS metadata consistency.

- 1 Obtain the name and partition number of the device that backs the VMFS datastore that you need to check.

```
#esxcli storage vmfs extent list
```

The Device Name and Partition columns in the output identify the device. For example:

```
Volume Name XXXXXXXX Device Name Partition
1TB_VMFS5 XXXXXXXX naa.600508e000000000b367477b3be3d703 3
```

- 2 Run VOMA to check for VMFS errors.

Provide the absolute path to the device partition that backs the VMFS datastore, and provide a partition number with the device name. For example:

```
# voma -m vmfs -f check -d /vmfs/devices/disks/naa.600508e000000000b367477b3be3d703:3
```

The output lists possible errors. For example, the following output indicates that the heartbeat address is invalid.

```
XXXXXXXXXXXXXXXXXXXXXXXXX
Phase 2: Checking VMFS heartbeat region
ON-DISK ERROR: Invalid HB address
Phase 3: Checking all file descriptors.
Phase 4: Checking pathname and connectivity.
Phase 5: Checking resource reference counts.
```

```
Total Errors Found: 1
```

Command options that the VOMA tool takes include the following.

Table 16-5. VOMA Command Options

Command Option	Description
-m --module	The module to run: <ul style="list-style-type: none"> ■ <code>vmfs</code>. This is a default option. You can check VMFS3 and VMFS 5 datastores. If you specify this module, minimal checks are performed for LVM as well. ■ <code>vmfs1</code>. Check file systems that back virtual flash volumes. ■ <code>lvm</code>. Check logical volumes that back VMFS datastores.
-f --func	Functions to be performed: <ul style="list-style-type: none"> ■ <code>query</code>. List functions supported by module. ■ <code>check</code>. Check for errors. ■ <code>fix</code>. Check and fix errors.
-d --device	Device or disk to be inspected. Make sure to provide the absolute path to the device partition backing the VMFS datastore. For example, <code>/vmfs/devices/disks/naa.00000000000000000000000000000000:1</code> .
-s --logfile	Specify the log file to output the results.
-v --version	Display the version of VOMA.
-h --help	Display the help message for the VOMA command.

Configuring VMFS Pointer Block Cache

You can use advanced VMFS parameters to configure pointer block cache.

As the size of virtual machine files on VMFS datastores increases, the number of pointer blocks used by those files also increases. Pointer blocks are used to address file blocks in large virtual machine files and virtual disks on a VMFS datastore.

You can configure the minimum and maximum sizes of the pointer block cache on each ESXi host. When the size of the pointer block cache approaches the configured maximum size, an eviction mechanism removes some pointer block entries from the cache to allow for new pointer blocks to be read in.

Base the maximum size of the pointer block cache on the working size of all open virtual disk files that reside on VMFS datastores. All VMFS datastores on the host use a single pointer block cache.

The minimum value is based on the minimum guaranteed memory that the system can allocate to the cache. 1TB of open file space requires approximately 4MB of memory.

Use the following parameters to configure the minimum and maximum values for pointer block cache.

Table 16-6. Advanced Parameters to Regulate Pointer Block Cache

Parameter	Values	Description
VMFS3.MaxAddressableSpaceTB	Default value is 32 (in TB).	Maximum size of all open files that VMFS cache will support before eviction starts.
VMFS3.MinAddressableSpaceTB	Default value is 10 (in TB).	Minimum size of all open files that VMFS cache guarantees to support.

For information about how to configure these advanced parameters, see [“Set Advanced Host Attributes,”](#) on page 179.

Set Advanced Host Attributes

You can set advanced attributes for a host.



CAUTION Changing advanced options is considered unsupported unless VMware technical support or a KB article instruct you to do so. In all other cases, changing these options is considered unsupported. In most cases, the default settings produce the optimum result.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab and click **Settings**.
- 3 Under **System**, click **Advanced System Settings**.
- 4 In Advanced Settings, select the appropriate item.
- 5 Click the **Edit** button to edit the value.
- 6 Click **OK**.

Understanding Multipathing and Failover

17

To maintain a constant connection between a host and its storage, ESXi supports multipathing. Multipathing is a technique that lets you use more than one physical path that transfers data between the host and an external storage device.

In case of a failure of any element in the SAN network, such as an adapter, switch, or cable, ESXi can switch to another physical path, which does not use the failed component. This process of path switching to avoid failed components is known as path failover.

In addition to path failover, multipathing provides load balancing. Load balancing is the process of distributing I/O loads across multiple physical paths. Load balancing reduces or removes potential bottlenecks.

NOTE Virtual machine I/O might be delayed for up to sixty seconds while path failover takes place. These delays allow the SAN to stabilize its configuration after topology changes. In general, the I/O delays might be longer on active-passive arrays and shorter on active-active arrays.

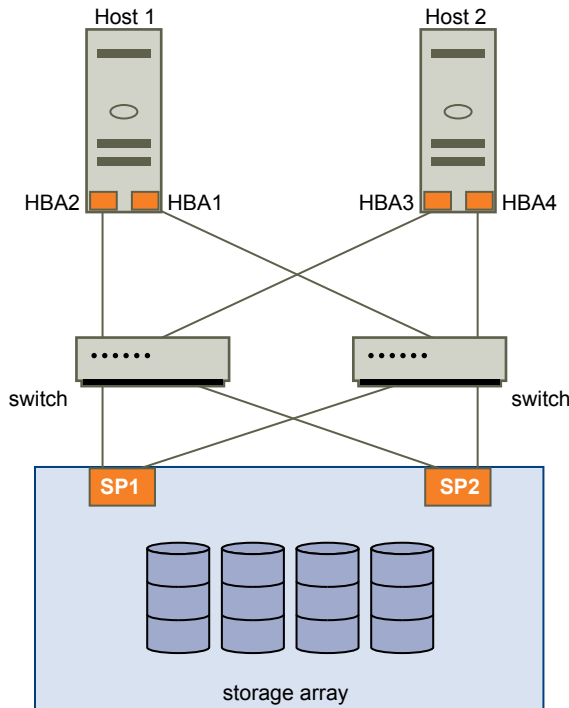
This chapter includes the following topics:

- [“Failover with Fibre Channel,”](#) on page 181
- [“Host-Based Failover with iSCSI,”](#) on page 182
- [“Array-Based Failover with iSCSI,”](#) on page 184
- [“Path Failover and Virtual Machines,”](#) on page 185
- [“Managing Multiple Paths,”](#) on page 186
- [“VMware Multipathing Module,”](#) on page 187
- [“Path Scanning and Claiming,”](#) on page 189
- [“Managing Storage Paths and Multipathing Plug-Ins,”](#) on page 192
- [“Scheduling Queues for Virtual Machine I/Os,”](#) on page 200

Failover with Fibre Channel

To support multipathing, your host typically has two or more HBAs available. This configuration supplements the SAN multipathing configuration that generally provides one or more switches in the SAN fabric and one or more storage processors on the storage array device itself.

In the following illustration, multiple physical paths connect each server with the storage device. For example, if HBA1 or the link between HBA1 and the FC switch fails, HBA2 takes over and provides the connection between the server and the switch. The process of one HBA taking over for another is called HBA failover.

Figure 17-1. Multipathing and Failover with Fibre Channel

Similarly, if SP1 fails or the links between SP1 and the switches breaks, SP2 takes over and provides the connection between the switch and the storage device. This process is called SP failover. VMware ESXi supports both HBA and SP failovers with its multipathing capability.

Host-Based Failover with iSCSI

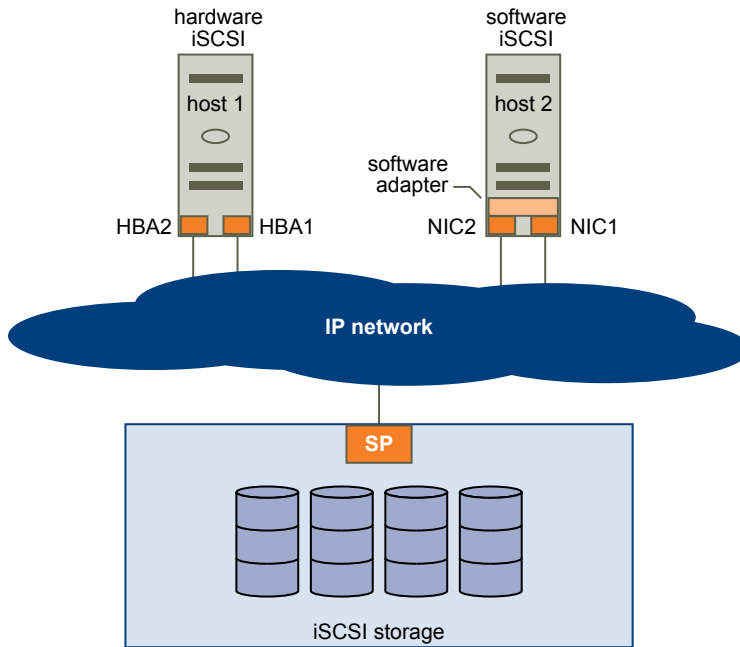
When setting up your ESXi host for multipathing and failover, you can use multiple iSCSI HBAs or multiple NICs depending on the type of iSCSI adapters on your host.

For information on different types of iSCSI adapters, see [“iSCSI Initiators,”](#) on page 65.

When you use multipathing, specific considerations apply.

- ESXi does not support multipathing when you combine an independent hardware adapter with software iSCSI or dependent iSCSI adapters in the same host.
- Multipathing between software and dependent adapters within the same host is supported.
- On different hosts, you can mix both dependent and independent adapters.

The following illustration shows multipathing setups possible with different types of iSCSI initiators.

Figure 17-2. Host-Based Path Failover

Failover with Hardware iSCSI

With hardware iSCSI, the host typically has two or more hardware iSCSI adapters available, from which the storage system can be reached using one or more switches. Alternatively, the setup might include one adapter and two storage processors so that the adapter can use a different path to reach the storage system.

On the Host-Based Path Failover illustration, Host1 has two hardware iSCSI adapters, HBA1 and HBA2, that provide two physical paths to the storage system. Multipathing plug-ins on your host, whether the VMkernel NMP or any third-party MPPs, have access to the paths by default and can monitor health of each physical path. If, for example, HBA1 or the link between HBA1 and the network fails, the multipathing plug-ins can switch the path over to HBA2.

Failover with Software iSCSI

With software iSCSI, as shown on Host 2 of the Host-Based Path Failover illustration, you can use multiple NICs that provide failover and load balancing capabilities for iSCSI connections between your host and storage systems.

For this setup, because multipathing plug-ins do not have direct access to physical NICs on your host, you first need to connect each physical NIC to a separate VMkernel port. You then associate all VMkernel ports with the software iSCSI initiator using a port binding technique. As a result, each VMkernel port connected to a separate NIC becomes a different path that the iSCSI storage stack and its storage-aware multipathing plug-ins can use.

For information on how to configure multipathing for software iSCSI, see [“Setting Up iSCSI Network,”](#) on page 88.

Array-Based Failover with iSCSI

Some iSCSI storage systems manage path use of their ports automatically and transparently to ESXi.

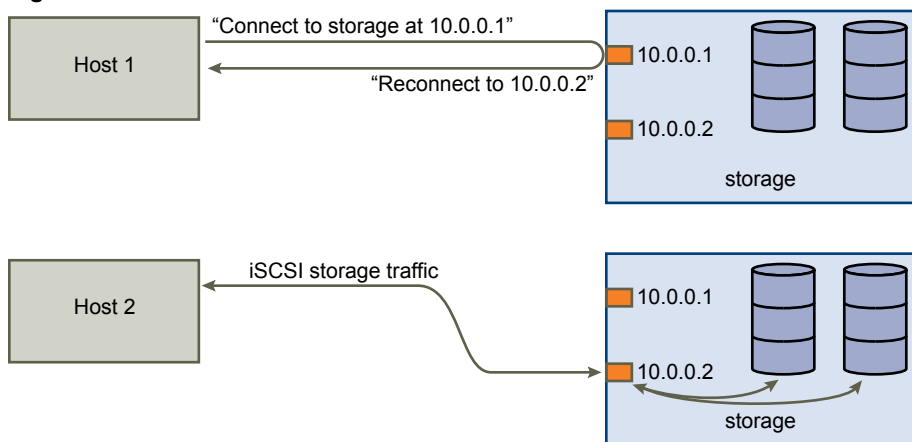
When using one of these storage systems, your host does not see multiple ports on the storage and cannot choose the storage port it connects to. These systems have a single virtual port address that your host uses to initially communicate. During this initial communication, the storage system can redirect the host to communicate with another port on the storage system. The iSCSI initiators in the host obey this reconnection request and connect with a different port on the system. The storage system uses this technique to spread the load across available ports.

If the ESXi host loses connection to one of these ports, it automatically attempts to reconnect with the virtual port of the storage system, and should be redirected to an active, usable port. This reconnection and redirection happens quickly and generally does not disrupt running virtual machines. These storage systems can also request that iSCSI initiators reconnect to the system, to change which storage port they are connected to. This allows the most effective use of the multiple ports.

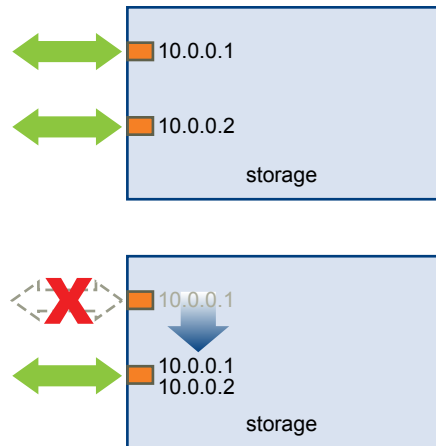
The Port Redirection illustration shows an example of port redirection. The host attempts to connect to the 10.0.0.1 virtual port. The storage system redirects this request to 10.0.0.2. The host connects with 10.0.0.2 and uses this port for I/O communication.

NOTE The storage system does not always redirect connections. The port at 10.0.0.1 could be used for traffic, also.

Figure 17-3. Port Redirection



If the port on the storage system that is acting as the virtual port becomes unavailable, the storage system reassigns the address of the virtual port to another port on the system. Port Reassignment shows an example of this type of port reassignment. In this case, the virtual port 10.0.0.1 becomes unavailable and the storage system reassigns the virtual port IP address to a different port. The second port responds to both addresses.

Figure 17-4. Port Reassignment

With this form of array-based failover, you can have multiple paths to the storage only if you use multiple ports on the ESXi host. These paths are active-active. For additional information, see [“iSCSI Session Management,”](#) on page 103.

Path Failover and Virtual Machines

Path failover occurs when the active path to a LUN is changed from one path to another, usually because of a SAN component failure along the current path.

When a path fails, storage I/O might pause for 30 to 60 seconds until your host determines that the link is unavailable and completes failover. If you attempt to display the host, its storage devices, or its adapters, the operation might appear to stall. Virtual machines with their disks installed on the SAN can appear unresponsive. After failover is complete, I/O resumes normally and the virtual machines continue to run.

However, when failovers take a long time to complete, a Windows virtual machine might interrupt the I/O and eventually fail. To avoid the failure, set the disk timeout value for the Windows virtual machine to at least 60 seconds.

Set Timeout on Windows Guest OS

Increase the standard disk timeout value on a Windows guest operating system to avoid disruptions during a path failover.

This procedure explains how to change the timeout value by using the Windows registry.

Prerequisites

Back up the Windows registry.

Procedure

- 1 Select **Start > Run**.
- 2 Type **regedit.exe**, and click **OK**.
- 3 In the left-panel hierarchy view, double-click **HKEY_LOCAL_MACHINE > System > CurrentControlSet > Services > Disk**.
- 4 Double-click **TimeOutValue**.
- 5 Set the value data to 0x3c (hexadecimal) or 60 (decimal) and click **OK**.

After you make this change, Windows waits at least 60 seconds for delayed disk operations to complete before it generates errors.

- 6 Reboot guest OS for the change to take effect.

Managing Multiple Paths

To manage storage multipathing, ESXi uses a collection of Storage APIs, also called the Pluggable Storage Architecture (PSA). The PSA is an open, modular framework that coordinates the simultaneous operation of multiple multipathing plug-ins (MPPs). The PSA allows 3rd party software developers to design their own load balancing techniques and failover mechanisms for particular storage array, and insert their code directly into the ESXi storage I/O path.

Topics discussing path management use the following acronyms.

Table 17-1. Multipathing Acronyms

Acronym	Definition
PSA	Pluggable Storage Architecture
NMP	Native Multipathing Plug-In. Generic VMware multipathing module.
PSP	Path Selection Plug-In, also called Path Selection Policy. Handles path selection for a given device.
SATP	Storage Array Type Plug-In, also called Storage Array Type Policy. Handles path failover for a given storage array.

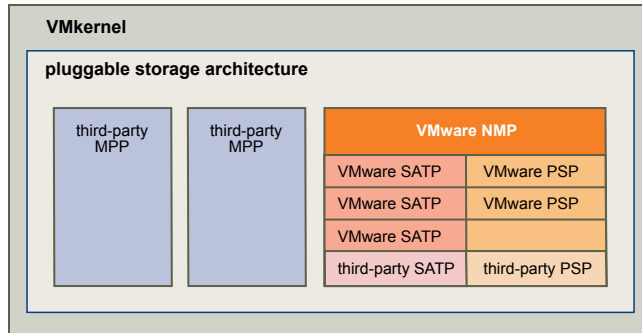
The VMkernel multipathing plug-in that ESXi provides by default is the VMware Native Multipathing Plug-In (NMP). The NMP is an extensible module that manages sub plug-ins. There are two types of NMP sub plug-ins, Storage Array Type Plug-Ins (SATPs), and Path Selection Plug-Ins (PSPs). SATPs and PSPs can be built-in and provided by VMware, or can be provided by a third party.

If more multipathing functionality is required, a third party can also provide an MPP to run in addition to, or as a replacement for, the default NMP.

When coordinating the VMware NMP and any installed third-party MPPs, the PSA performs the following tasks:

- Loads and unloads multipathing plug-ins.
- Hides virtual machine specifics from a particular plug-in.
- Routes I/O requests for a specific logical device to the MPP managing that device.
- Handles I/O queuing to the logical devices.
- Implements logical device bandwidth sharing between virtual machines.
- Handles I/O queuing to the physical storage HBAs.
- Handles physical path discovery and removal.
- Provides logical device and physical path I/O statistics.

As the Pluggable Storage Architecture illustration shows, multiple third-party MPPs can run in parallel with the VMware NMP. When installed, the third-party MPPs replace the behavior of the NMP and take complete control of the path failover and the load-balancing operations for specified storage devices.

Figure 17-5. Pluggable Storage Architecture

The multipathing modules perform the following operations:

- Manage physical path claiming and unclaiming.
- Manage creation, registration, and deregistration of logical devices.
- Associate physical paths with logical devices.
- Support path failure detection and remediation.
- Process I/O requests to logical devices:
 - Select an optimal physical path for the request.
 - Depending on a storage device, perform specific actions necessary to handle path failures and I/O command retries.
- Support management tasks, such as reset of logical devices.

VMware Multipathing Module

By default, ESXi provides an extensible multipathing module called the Native Multipathing Plug-In (NMP).

Generally, the VMware NMP supports all storage arrays listed on the VMware storage HCL and provides a default path selection algorithm based on the array type. The NMP associates a set of physical paths with a specific storage device, or LUN. The specific details of handling path failover for a given storage array are delegated to a Storage Array Type Plug-In (SATP). The specific details for determining which physical path is used to issue an I/O request to a storage device are handled by a Path Selection Plug-In (PSP). SATPs and PSPs are sub plug-ins within the NMP module.

With ESXi, the appropriate SATP for an array you use will be installed automatically. You do not need to obtain or download any SATPs.

VMware SATPs

Storage Array Type Plug-Ins (SATPs) run in conjunction with the VMware NMP and are responsible for array-specific operations.

ESXi offers a SATP for every type of array that VMware supports. It also provides default SATPs that support non-specific active-active and ALUA storage arrays, and the local SATP for direct-attached devices. Each SATP accommodates special characteristics of a certain class of storage arrays and can perform the array-specific operations required to detect path state and to activate an inactive path. As a result, the NMP module itself can work with multiple storage arrays without having to be aware of the storage device specifics.

After the NMP determines which SATP to use for a specific storage device and associates the SATP with the physical paths for that storage device, the SATP implements the tasks that include the following:

- Monitors the health of each physical path.
- Reports changes in the state of each physical path.
- Performs array-specific actions necessary for storage fail-over. For example, for active-passive devices, it can activate passive paths.

VMware PSPs

Path Selection Plug-Ins (PSPs) are sub plug-ins of the VMware NMP and are responsible for choosing a physical path for I/O requests.

The VMware NMP assigns a default PSP for each logical device based on the SATP associated with the physical paths for that device. You can override the default PSP. For information, see [“Path Scanning and Claiming,”](#) on page 189.

By default, the VMware NMP supports the following PSPs:

VMW_PSP_MRU

The host selects the path that it used most recently. When the path becomes unavailable, the host selects an alternative path. The host does not revert back to the original path when that path becomes available again. There is no preferred path setting with the MRU policy. MRU is the default policy for most active-passive storage devices.

The VMW_PSP_MRU ranking capability allows you to assign ranks to individual paths. To set ranks to individual paths, use the `esxcli storage nmp psp generic pathconfig set` command. For details, see the VMware knowledge base article at <http://kb.vmware.com/kb/2003468>.

The policy is displayed in the client as the Most Recently Used (VMware) path selection policy.

VMW_PSP_FIXED

The host uses the designated preferred path, if it has been configured. Otherwise, it selects the first working path discovered at system boot time. If you want the host to use a particular preferred path, specify it manually. Fixed is the default policy for most active-active storage devices.

NOTE If the host uses a default preferred path and the path's status turns to Dead, a new path is selected as preferred. However, if you explicitly designate the preferred path, it will remain preferred even when it becomes inaccessible.

Displayed in the client as the Fixed (VMware) path selection policy.

VMW_PSP_RR

The host uses an automatic path selection algorithm rotating through all active paths when connecting to active-passive arrays, or through all available paths when connecting to active-active arrays. RR is the default for a number of arrays and can be used with both active-active and active-passive arrays to implement load balancing across paths for different LUNs.

Displayed in the client as the Round Robin (VMware) path selection policy.

VMware NMP Flow of I/O

When a virtual machine issues an I/O request to a storage device managed by the NMP, the following process takes place.

- 1 The NMP calls the PSP assigned to this storage device.

- 2 The PSP selects an appropriate physical path on which to issue the I/O.
- 3 The NMP issues the I/O request on the path selected by the PSP.
- 4 If the I/O operation is successful, the NMP reports its completion.
- 5 If the I/O operation reports an error, the NMP calls the appropriate SATP.
- 6 The SATP interprets the I/O command errors and, when appropriate, activates the inactive paths.
- 7 The PSP is called to select a new path on which to issue the I/O.

Path Scanning and Claiming

When you start your ESXi host or rescan your storage adapter, the host discovers all physical paths to storage devices available to the host. Based on a set of claim rules, the host determines which multipathing plug-in (MPP) should claim the paths to a particular device and become responsible for managing the multipathing support for the device.

By default, the host performs a periodic path evaluation every 5 minutes causing any unclaimed paths to be claimed by the appropriate MPP.

The claim rules are numbered. For each physical path, the host runs through the claim rules starting with the lowest number first. The attributes of the physical path are compared to the path specification in the claim rule. If there is a match, the host assigns the MPP specified in the claim rule to manage the physical path. This continues until all physical paths are claimed by corresponding MPPs, either third-party multipathing plug-ins or the native multipathing plug-in (NMP).

For the paths managed by the NMP module, a second set of claim rules is applied. These rules determine which Storage Array Type Plug-In (SATP) should be used to manage the paths for a specific array type, and which Path Selection Plug-In (PSP) is to be used for each storage device.

Use the vSphere Web Client to view which SATP and PSP the host is using for a specific storage device and the status of all available paths for this storage device. If needed, you can change the default VMware PSP using the client. To change the default SATP, you need to modify claim rules using the vSphere CLI.

You can find some information about modifying claim rules in [“Managing Storage Paths and Multipathing Plug-Ins,”](#) on page 192.

For more information about the commands available to manage PSA, see *Getting Started with vSphere Command-Line Interfaces*.

For a complete list of storage arrays and corresponding SATPs and PSPs, see the SAN Array Model Reference section of the *vSphere Compatibility Guide*.

Viewing the Paths Information

You can review the storage array type policy (SATP) and path selection policy (PSP) that the ESXi host uses for a specific storage device and the status of all available paths for this storage device. You can access the path information from both the Datastores and Devices views. For datastores, you review the paths that connect to the device the datastore is deployed on.

The path information includes the SATP assigned to manage the device, the PSP, a list of paths, and the status of each path. The following path status information can appear:

Active	Paths available for issuing I/O to a LUN. A single or multiple working paths currently used for transferring data are marked as Active (I/O).
Standby	If active paths fail, the path can quickly become operational and can be used for I/O.

Disabled	The path is disabled and no data can be transferred.
Dead	The software cannot connect to the disk through this path.

If you are using the **Fixed** path policy, you can see which path is the preferred path. The preferred path is marked with an asterisk (*) in the Preferred column.

For each path you can also display the path's name. The name includes parameters that describe the path: adapter ID, target ID, and device ID. Usually, the path's name has the format similar to the following:

```
fc.adapterID-fc.targetID-naa.deviceID
```

NOTE When you use the host profiles editor to edit paths, you must specify all three parameters that describe a path, adapter ID, target ID, and device ID.

View Datastore Paths

Review the paths that connect to storage devices backing your datastores.

Procedure

- 1 In the vSphere Web Client navigator, select **vCenter Inventory Lists > Datastores**
- 2 Click the datastore to display its information.
- 3 Click the **Manage** tab, and click **Settings**.
- 4 Click **Connectivity and Multipathing**.
- 5 If the datastore is shared, select a host to view multipathing details for its devices.
- 6 Under Multipathing Details, review the multipathing policies and paths for the storage device that backs your datastore.

View Storage Device Paths

View which multipathing policies the host uses for a specific storage device and the status of all available paths for this storage device.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Devices**.
- 4 Select the storage device whose paths you want to view.
- 5 Click the **Properties** tab and review details under Multipathing Policies.
- 6 Click the **Paths** tab to review all paths available for the storage device.

Setting a Path Selection Policy

For each storage device, the ESXi host sets the path selection policy based on the claim rules.

By default, VMware supports the following path selection policies. If you have a third-party PSP installed on your host, its policy also appears on the list.

Fixed (VMware) The host uses the designated preferred path, if it has been configured. Otherwise, it selects the first working path discovered at system boot time. If you want the host to use a particular preferred path, specify it manually. Fixed is the default policy for most active-active storage devices.

NOTE If the host uses a default preferred path and the path's status turns to Dead, a new path is selected as preferred. However, if you explicitly designate the preferred path, it will remain preferred even when it becomes inaccessible.

Most Recently Used (VMware) The host selects the path that it used most recently. When the path becomes unavailable, the host selects an alternative path. The host does not revert back to the original path when that path becomes available again. There is no preferred path setting with the MRU policy. MRU is the default policy for most active-passive storage devices.

Round Robin (VMware) The host uses an automatic path selection algorithm rotating through all active paths when connecting to active-passive arrays, or through all available paths when connecting to active-active arrays. RR is the default for a number of arrays and can be used with both active-active and active-passive arrays to implement load balancing across paths for different LUNs.

Change the Path Selection Policy

Generally, you do not need to change the default multipathing settings your host uses for a specific storage device. However, if you want to make any changes, you can use the Edit Multipathing Policies dialog box to modify a path selection policy and specify the preferred path for the Fixed policy. You can also use this dialog box to change multipathing for SCSI-based protocol endpoints.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Devices** or **Protocol Endpoints**.
- 4 Select the item whose paths you want to change and click the **Properties** tab.
- 5 Under Multipathing Policies, click **Edit Multipathing**.
- 6 Select a path policy.

By default, VMware supports the following path selection policies. If you have a third-party PSP installed on your host, its policy also appears on the list.

- Fixed (VMware)
 - Most Recently Used (VMware)
 - Round Robin (VMware)
- 7 For the fixed policy, specify the preferred path.

- 8 Click **OK** to save your settings and exit the dialog box.

Disable Storage Paths

You can temporarily disable paths for maintenance or other reasons.

You disable a path using the Paths panel. You have several ways to access the Paths panel, from a datastore, a storage device, or an adapter view. This task explains how to disable a path using a storage device view.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Storage Devices**.
- 4 Select the storage device whose paths you want to disable and click the **Paths** tab.
- 5 Select the path to disable and click **Disable**.

Managing Storage Paths and Multipathing Plug-Ins

Use the `esxcli` commands to manage the PSA multipathing plug-ins and storage paths assigned to them.

You can display all multipathing plug-ins available on your host. You can list any third-party MPPs, as well as your host's NMP and SATPs and review the paths they claim. You can also define new paths and specify which multipathing plug-in should claim the paths.

For more information about commands available to manage PSA, see the *Getting Started with vSphere Command-Line Interfaces*.

Multipathing Considerations

Specific considerations apply when you manage storage multipathing plug-ins and claim rules.

The following considerations help you with multipathing:

- If no SATP is assigned to the device by the claim rules, the default SATP for iSCSI or FC devices is `VMW_SATP_DEFAULT_AA`. The default PSP is `VMW_PSP_FIXED`.
- When the system searches the SATP rules to locate a SATP for a given device, it searches the driver rules first. If there is no match, the vendor/model rules are searched, and finally the transport rules are searched. If no match occurs, NMP selects a default SATP for the device.
- If `VMW_SATP_ALUA` is assigned to a specific storage device, but the device is not ALUA-aware, no claim rule match occurs for this device. The device is claimed by the default SATP based on the device's transport type.
- The default PSP for all devices claimed by `VMW_SATP_ALUA` is `VMW_PSP_MRU`. The `VMW_PSP_MRU` selects an active/optimized path as reported by the `VMW_SATP_ALUA`, or an active/unoptimized path if there is no active/optimized path. This path is used until a better path is available (MRU). For example, if the `VMW_PSP_MRU` is currently using an active/unoptimized path and an active/optimized path becomes available, the `VMW_PSP_MRU` will switch the current path to the active/optimized one.
- While `VMW_PSP_MRU` is typically selected for ALUA arrays by default, certain ALUA storage arrays need to use `VMW_PSP_FIXED`. To check whether your storage array requires `VMW_PSP_FIXED`, see the *VMware Compatibility Guide* or contact your storage vendor. When using `VMW_PSP_FIXED` with ALUA arrays, unless you explicitly specify a preferred path, the ESXi host selects the most optimal working path and designates it as the default preferred path. If the host selected path becomes unavailable, the host selects an alternative available path. However, if you explicitly designate the preferred path, it will remain preferred no matter what its status is.

- By default, the PSA claim rule 101 masks Dell array pseudo devices. Do not delete this rule, unless you want to unmask these devices.

List Multipathing Claim Rules for the Host

Use the `esxcli` command to list available multipathing claim rules.

Claim rules indicate which multipathing plug-in, the NMP or any third-party MPP, manages a given physical path. Each claim rule identifies a set of paths based on the following parameters:

- Vendor/model strings
- Transportation, such as SATA, IDE, Fibre Channel, and so on
- Adapter, target, or LUN location
- Device driver, for example, Mega-RAID

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ Run the `esxcli --server=server_name storage core claimrule list --claimrule-class=MP` command to list the multipathing claim rules.

Example: Sample Output of the `esxcli storage core claimrule list` Command

Rule	Class	Rule	Class	Type	Plugin	Matches
MP		0	runtime	transport	NMP	transport=usb
MP		1	runtime	transport	NMP	transport=sata
MP		2	runtime	transport	NMP	transport=ide
MP		3	runtime	transport	NMP	transport=block
MP		4	runtime	transport	NMP	transport=unknown
MP		101	runtime	vendor	MASK_PATH	vendor=DELL model=Universal Xport
MP		101	file	vendor	MASK_PATH	vendor=DELL model=Universal Xport
MP		200	runtime	vendor	MPP_1	vendor=NewVend model=*
MP		200	file	vendor	MPP_1	vendor=NewVend model=*
MP		201	runtime	location	MPP_2	adapter=vmhba41 channel=* target=* lun=*
MP		201	file	location	MPP_2	adapter=vmhba41 channel=* target=* lun=*
MP		202	runtime	driver	MPP_3	driver=megaraid
MP		202	file	driver	MPP_3	driver=megaraid
MP		65535	runtime	vendor	NMP	vendor=* model=*

This example indicates the following:

- The NMP claims all paths connected to storage devices that use the USB, SATA, IDE, and Block SCSI transportation.
- You can use the MASK_PATH module to hide unused devices from your host. By default, the PSA claim rule 101 masks Dell array pseudo devices with a vendor string of DELL and a model string of Universal Xport.
- The MPP_1 module claims all paths connected to any model of the NewVend storage array.
- The MPP_3 module claims the paths to storage devices controlled by the Mega-RAID device driver.

- Any paths not described in the previous rules are claimed by NMP.
- The Rule Class column in the output describes the category of a claim rule. It can be MP (multipathing plug-in), Filter, or VAAI.
- The Class column shows which rules are defined and which are loaded. The file parameter in the Class column indicates that the rule is defined. The runtime parameter indicates that the rule has been loaded into your system. For a user-defined claim rule to be active, two lines with the same rule number should exist, one line for the rule with the file parameter and another line with runtime. Several low numbered rules, have only one line with the Class of runtime. These are system-defined claim rules that you cannot modify.

Display Multipathing Modules

Use the `esxcli` command to list all multipathing modules loaded into the system. Multipathing modules manage physical paths that connect your host with storage.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ To list multipathing modules, run the following command:

```
esxcli --server=server_name storage core plugin list --plugin-class=MP
```

This command typically shows the NMP and, if loaded, the MASK_PATH module. If any third-party MPPs have been loaded, they are listed as well.

Display SATPs for the Host

Use the `esxcli` command to list VMware NMP SATPs loaded into the system. Display information about the SATPs.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ To list VMware SATPs, run the following command:

```
esxcli --server=server_name storage nmp satp list
```

For each SATP, the output displays information that shows the type of storage array or system this SATP supports and the default PSP for any LUNs using this SATP. Placeholder (plugin not loaded) in the Description column indicates that the SATP is not loaded.

Display NMP Storage Devices

Use the `esxcli` command to list all storage devices controlled by the VMware NMP and display SATP and PSP information associated with each device.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ To list all storage devices, run the following command:

```
esxcli --server=server_name storage nmp device list
```

Use the `--device | -d=device_ID` option to filter the output of this command to show a single device.

Add Multipathing Claim Rules

Use the `esxcli` commands to add a new multipathing PSA claim rule to the set of claim rules on the system. For the new claim rule to be active, you first define the rule and then load it into your system.

You add a new PSA claim rule when, for example, you load a new multipathing plug-in (MPP) and need to define which paths this module should claim. You may need to create a claim rule if you add new paths and want an existing MPP to claim them.



CAUTION When creating new claim rules, be careful to avoid a situation where different physical paths to the same LUN are claimed by different MPPs. Unless one of the MPPs is the MASK_PATH MPP, this configuration will cause performance problems.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 To define a new claim rule, run the following command:

```
esxcli --server=server_name storage core claimrule add
```

The command takes the following options:

Option	Description
<code>-A --adapter=<str></code>	Indicate the adapter of the paths to use in this operation.
<code>-u --autoassign</code>	The system will auto assign a rule ID.
<code>-C --channel=<long></code>	Indicate the channel of the paths to use in this operation.
<code>-c --claimrule-class=<str></code>	Indicate the claim rule class to use in this operation. Valid values are: MP, Filter, VAAI.
<code>-d --device=<str></code>	Indicate the device Uid to use for this operation.

Option	Description
-D --driver=<str>	Indicate the driver of the paths to use in this operation.
-f --force	Force claim rules to ignore validity checks and install the rule anyway.
--if-unset=<str>	Execute this command if this advanced user variable is not set to 1.
-i --iqn=<str>	Indicate the iSCSI Qualified Name for the target to use in this operation.
-L --lun=<long>	Indicate the LUN of the paths to use in this operation.
-M --model=<str>	Indicate the model of the paths to use in this operation.
-P --plugin=<str>	Indicate which PSA plugin to use for this operation. (required)
-r --rule=<long>	Indicate the rule ID to use for this operation.
-T --target=<long>	Indicate the target of the paths to use in this operation.
-R --transport=<str>	Indicate the transport of the paths to use in this operation. Valid values are: block, fc, iscsi, iscsivendor, ide, sas, sata, usb, parallel, unknown.
-t --type=<str>	Indicate which type of matching is used for claim/unclaim or claimrule. Valid values are: vendor, location, driver, transport, device, target. (required)
-V --vendor=<str>	Indicate the vendor of the paths to user in this operation.
--wwnn=<str>	Indicate the World-Wide Node Number for the target to use in this operation.
--wwpn=<str>	Indicate the World-Wide Port Number for the target to use in this operation.

- To load the new claim rule into your system, run the following command:

```
esxcli --server=server_name storage core claimrule load
```

This command loads all newly created multipathing claim rules from your system's configuration file.

Example: Defining Multipathing Claim Rules

In the following example, you add and load rule # 500 to claim all paths with the NewMod model string and the NewVend vendor string for the NMP plug-in.

```
# esxcli --server=server_name storage core claimrule add -r 500 -t vendor -V NewVend -M NewMod -P NMP
```

```
# esxcli --server=server_name storage core claimrule load
```

After you run the **esxcli --server=server_name storage core claimrule list** command, you can see the new claim rule appearing on the list.

NOTE The two lines for the claim rule, one with the Class of runtime and another with the Class of file, indicate that the new claim rule has been loaded into the system and is active.

Rule	Class	Rule	Class	Type	Plugin	Matches
MP		0	runtime	transport	NMP	transport=usb
MP		1	runtime	transport	NMP	transport=sata
MP		2	runtime	transport	NMP	transport=ide
MP		3	runtime	transport	NMP	transport=block
MP		4	runtime	transport	NMP	transport=unknown
MP		101	runtime	vendor	MASK_PATH	vendor=DELL model=Universal Xport
MP		101	file	vendor	MASK_PATH	vendor=DELL model=Universal Xport
MP		500	runtime	vendor	NMP	vendor=NewVend model=NewMod
MP		500	file	vendor	NMP	vendor=NewVend model=NewMod

Delete Multipathing Claim Rules

Use the `esxcli` commands to remove a multipathing PSA claim rule from the set of claim rules on the system.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 Delete a claim rule from the set of claim rules.

```
esxcli --server=server_name storage core claimrule remove
```

NOTE By default, the PSA claim rule 101 masks Dell array pseudo devices. Do not delete this rule, unless you want to unmask these devices.

The command takes the following options:

Option	Description
<code>-c --claimrule-class=<str></code>	Indicate the claim rule class to use in this operation (MP, Filter, VAAI).
<code>-P --plugin=<str></code>	Indicate the plugin to use for this operation.
<code>-r --rule=<long></code>	Indicate the rule ID to use for this operation.

This step removes the claim rule from the File class.

- 2 Remove the claim rule from the system.

```
esxcli --server=server_name storage core claimrule load
```

This step removes the claim rule from the Runtime class.

Mask Paths

You can prevent the host from accessing storage devices or LUNs or from using individual paths to a LUN. Use the `esxcli` commands to mask the paths. When you mask paths, you create claim rules that assign the MASK_PATH plug-in to the specified paths.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 Check what the next available rule ID is.

```
esxcli --server=server_name storage core claimrule list
```

The claim rules that you use to mask paths should have rule IDs in the range of 101 – 200. If this command shows that rule 101 and 102 already exist, you can specify 103 for the rule to add.

- 2 Assign the MASK_PATH plug-in to a path by creating a new claim rule for the plug-in.

```
esxcli --server=server_name storage core claimrule add -P MASK_PATH
```

- 3 Load the MASK_PATH claim rule into your system.

```
esxcli --server=server_name storage core claimrule load
```

- 4 Verify that the MASK_PATH claim rule was added correctly.

```
esxcli --server=server_name storage core claimrule list
```

- 5 If a claim rule for the masked path exists, remove the rule.

```
esxcli --server=server_name storage core claiming unclaim
```

- 6 Run the path claiming rules.

```
esxcli --server=server_name storage core claimrule run
```

After you assign the MASK_PATH plug-in to a path, the path state becomes irrelevant and is no longer maintained by the host. As a result, commands that display the masked path's information might show the path state as dead.

Example: Masking a LUN

In this example, you mask the LUN 20 on targets T1 and T2 accessed through storage adapters vmhba2 and vmhba3.

```
1 #esxcli --server=server_name storage core claimrule list
2 #esxcli --server=server_name storage core claimrule add -P MASK_PATH -r 109 -t location -A
  vmhba2 -C 0 -T 1 -L 20
  #esxcli --server=server_name storage core claimrule add -P MASK_PATH -r 110 -t location -A
  vmhba3 -C 0 -T 1 -L 20
  #esxcli --server=server_name storage core claimrule add -P MASK_PATH -r 111 -t location -A
  vmhba2 -C 0 -T 2 -L 20
  #esxcli --server=server_name storage core claimrule add -P MASK_PATH -r 112 -t location -A
  vmhba3 -C 0 -T 2 -L 20
3 #esxcli --server=server_name storage core claimrule load
4 #esxcli --server=server_name storage core claimrule list
5 #esxcli --server=server_name storage core claiming unclaim -t location -A vmhba2
  #esxcli --server=server_name storage core claiming unclaim -t location -A vmhba3
6 #esxcli --server=server_name storage core claimrule run
```

Unmask Paths

When you need the host to access the masked storage device, unmask the paths to the device.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

NOTE When you run an unclaim operation using a device property, for example, device ID, vendor, or model, the paths claimed by the MASK_PATH plugin are not unclaimed. The MASK_PATH plugin does not keep track of any device property of the paths that it claims.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 Delete the MASK_PATH claim rule.

```
esxcli --server=server_name storage core claimrule remove -r rule#
```

- 2 Verify that the claim rule was deleted correctly.

```
esxcli --server=server_name storage core claimrule list
```

- 3 Reload the path claiming rules from the configuration file into the VMkernel.

```
esxcli --server=server_name storage core claimrule load
```

- 4 Run the `esxcli --server=server_name storage core claiming unclaim` command for each path to the masked storage device.

For example:

```
esxcli --server=server_name storage core claiming unclaim -t location -A vmhba0 -C 0 -T 0 -L 149
```

- 5 Run the path claiming rules.

```
esxcli --server=server_name storage core claimrule run
```

Your host can now access the previously masked storage device.

Define NMP SATP Rules

The NMP SATP claim rules specify which SATP should manage a particular storage device. Usually you do not need to modify the NMP SATP rules. If you need to do so, use the `esxcli` commands to add a rule to the list of claim rules for the specified SATP.

You might need to create a SATP rule when you install a third-party SATP for a specific storage array.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 To add a claim rule for a specific SATP, run the `esxcli --server=server_name storage nmp satp rule add` command. The command takes the following options.

Option	Description
<code>-b --boot</code>	This is a system default rule added at boot time. Do not modify <code>esx.conf</code> or add to host profile.
<code>-c --claim-option=<i>string</i></code>	Set the claim option string when adding a SATP claim rule.
<code>-e --description=<i>string</i></code>	Set the claim rule description when adding a SATP claim rule.
<code>-d --device=<i>string</i></code>	Set the device when adding SATP claim rules. Device rules are mutually exclusive with vendor/model and driver rules.
<code>-D --driver=<i>string</i></code>	Set the driver string when adding a SATP claim rule. Driver rules are mutually exclusive with vendor/model rules.
<code>-f --force</code>	Force claim rules to ignore validity checks and install the rule anyway.
<code>-h --help</code>	Show the help message.

Option	Description
-M --model=string	Set the model string when adding SATP a claim rule. Vendor/Model rules are mutually exclusive with driver rules.
-o --option=string	Set the option string when adding a SATP claim rule.
-P --psp=string	Set the default PSP for the SATP claim rule.
-O --psp-option=string	Set the PSP options for the SATP claim rule.
-s --satp=string	The SATP for which a new rule will be added.
-R --transport=string	Set the claim transport type string when adding a SATP claim rule.
-t --type=string	Set the claim type when adding a SATP claim rule.
-V --vendor=string	Set the vendor string when adding SATP claim rules. Vendor/Model rules are mutually exclusive with driver rules.

NOTE When searching the SATP rules to locate a SATP for a given device, the NMP searches the driver rules first. If there is no match, the vendor/model rules are searched, and finally the transport rules. If there is still no match, NMP selects a default SATP for the device.

- 2 Reboot your host.

Example: Defining an NMP SATP Rule

The following sample command assigns the VMW_SATP_INV plug-in to manage storage arrays with vendor string NewVend and model string NewMod.

```
# esxcli --server=server_name storage nmp satp rule add -V NewVend -M NewMod -s VMW_SATP_INV
```

If you run the `esxcli --server=server_name storage nmp satp list -s VMW_SATP_INV` command, you can see the new rule added to the list of VMW_SATP_INV rules.

Scheduling Queues for Virtual Machine I/Os

By default, vSphere provides a mechanism that creates scheduling queues for every virtual machine file. Each file, for example `.vmdk`, gets its own bandwidth controls.

This mechanism ensures that I/O for a particular virtual machine file, such as `.vmdk`, goes into its own separate queue and does not interfere with I/Os from other files.

This capability is enabled by default. If you need to turn it off, you can do this by adjusting the `VMkernel.Boot.isPerFileSchedModelActive` parameter in the advanced system settings.

Edit Per File I/O Scheduling

The advanced `VMkernel.Boot.isPerFileSchedModelActive` parameter controls the per file I/O scheduling mechanism. The mechanism is enabled by default.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Settings**.
- 3 Under System, click **Advanced System Settings**.
- 4 Under Advanced System Settings, select the **VMkernel.Boot.isPerFileSchedModelActive** parameter and click the **Edit** icon.

- 5 Select one of the following options:
 - To disable the per file scheduling mechanism, change the value to **No**.

NOTE After you turn off the per file I/O scheduling model, your host reverts to a legacy scheduling mechanism that uses a single I/O queue. The host maintains the single I/O queue for each virtual machine and storage device pair. All I/Os between the virtual machine and its virtual disks stored on the storage device are moved into this queue. As a result, I/Os from different virtual disks might interfere with each other in sharing the bandwidth and affect each others performance.

- To reenble the per file scheduling mechanism, change the value to **Yes**.
- 6 Reboot the host for the changes to take effect.

Use esxcli Commands to Enable or Disable Per File I/O Scheduling

You can use the esxcli commands to change to I/O scheduling capability. The capability is enabled by default.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ To enable or disable per file I/O scheduling, run the following commands:

Option	Description
<code>esxcli system settings kernel set -s isPerFileSchedModelActive -v FALSE</code>	Disable per file I/O scheduling
<code>esxcli system settings kernel set -s isPerFileSchedModelActive -v TRUE</code>	Enable per file I/O scheduling

Raw Device Mapping

Raw device mapping (RDM) provides a mechanism for a virtual machine to have direct access to a LUN on the physical storage subsystem.

The following topics contain information about RDMs and provide instructions on how to create and manage RDMs.

This chapter includes the following topics:

- [“About Raw Device Mapping,”](#) on page 203
- [“Raw Device Mapping Characteristics,”](#) on page 206
- [“Create Virtual Machines with RDMs,”](#) on page 208
- [“Manage Paths for a Mapped LUN,”](#) on page 209

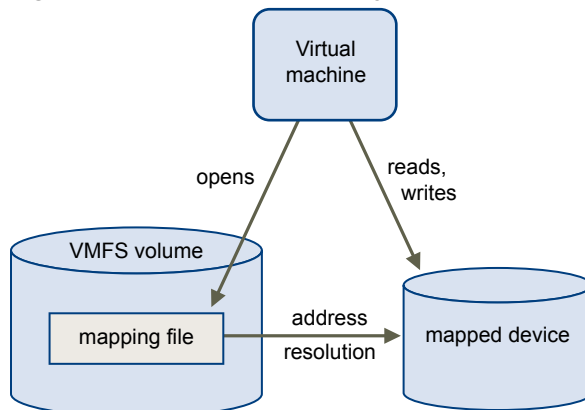
About Raw Device Mapping

An RDM is a mapping file in a separate VMFS volume that acts as a proxy for a raw physical storage device. The RDM allows a virtual machine to directly access and use the storage device. The RDM contains metadata for managing and redirecting disk access to the physical device.

The file gives you some of the advantages of direct access to a physical device while keeping some advantages of a virtual disk in VMFS. As a result, it merges VMFS manageability with raw device access.

RDMs can be described in terms such as mapping a raw device into a datastore, mapping a system LUN, or mapping a disk file to a physical disk volume. All these terms refer to RDMs.

Figure 18-1. Raw Device Mapping



Although VMware recommends that you use VMFS datastores for most virtual disk storage, on certain occasions, you might need to use raw LUNs or logical disks located in a SAN.

For example, you need to use raw LUNs with RDMs in the following situations:

- When SAN snapshot or other layered applications run in the virtual machine. The RDM better enables scalable backup offloading systems by using features inherent to the SAN.
- In any MSCS clustering scenario that spans physical hosts — virtual-to-virtual clusters as well as physical-to-virtual clusters. In this case, cluster data and quorum disks should be configured as RDMs rather than as virtual disks on a shared VMFS.

Think of an RDM as a symbolic link from a VMFS volume to a raw LUN. The mapping makes LUNs appear as files in a VMFS volume. The RDM, not the raw LUN, is referenced in the virtual machine configuration. The RDM contains a reference to the raw LUN.

Using RDMs, you can:

- Use vMotion to migrate virtual machines using raw LUNs.
- Add raw LUNs to virtual machines using the vSphere Web Client.
- Use file system features such as distributed file locking, permissions, and naming.

Two compatibility modes are available for RDMs:

- Virtual compatibility mode allows an RDM to act exactly like a virtual disk file, including the use of snapshots.
- Physical compatibility mode allows direct access of the SCSI device for those applications that need lower level control.

Benefits of Raw Device Mapping

An RDM provides a number of benefits, but it should not be used in every situation. In general, virtual disk files are preferable to RDMs for manageability. However, when you need raw devices, you must use the RDM.

RDM offers several benefits.

User-Friendly Persistent Names	Provides a user-friendly name for a mapped device. When you use an RDM, you do not need to refer to the device by its device name. You refer to it by the name of the mapping file, for example: <code>/vmfs/volumes/myVolume/myVMDirectory/myRawDisk.vmdk</code>
Dynamic Name Resolution	Stores unique identification information for each mapped device. VMFS associates each RDM with its current SCSI device, regardless of changes in the physical configuration of the server because of adapter hardware changes, path changes, device relocation, and so on.
Distributed File Locking	Makes it possible to use VMFS distributed locking for raw SCSI devices. Distributed locking on an RDM makes it safe to use a shared raw LUN without losing data when two virtual machines on different servers try to access the same LUN.
File Permissions	Makes file permissions possible. The permissions of the mapping file are enforced at file-open time to protect the mapped volume.
File System Operations	Makes it possible to use file system utilities to work with a mapped volume, using the mapping file as a proxy. Most operations that are valid for an ordinary file can be applied to the mapping file and are redirected to operate on the mapped device.

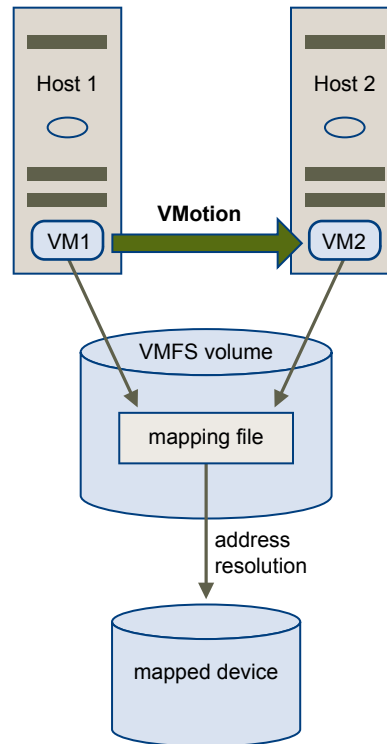
Snapshots

Makes it possible to use virtual machine snapshots on a mapped volume. Snapshots are not available when the RDM is used in physical compatibility mode.

vMotion

Lets you migrate a virtual machine with vMotion. The mapping file acts as a proxy to allow vCenter Server to migrate the virtual machine by using the same mechanism that exists for migrating virtual disk files.

Figure 18-2. vMotion of a Virtual Machine Using Raw Device Mapping

**SAN Management Agents**

Makes it possible to run some SAN management agents inside a virtual machine. Similarly, any software that needs to access a device by using hardware-specific SCSI commands can be run in a virtual machine. This kind of software is called SCSI target-based software. When you use SAN management agents, select a physical compatibility mode for the RDM.

N-Port ID Virtualization (NPIV)

Makes it possible to use the NPIV technology that allows a single Fibre Channel HBA port to register with the Fibre Channel fabric using several worldwide port names (WWPNs). This ability makes the HBA port appear as multiple virtual ports, each having its own ID and virtual port name. Virtual machines can then claim each of these virtual ports and use them for all RDM traffic.

NOTE You can use NPIV only for virtual machines with RDM disks.

VMware works with vendors of storage management software to ensure that their software functions correctly in environments that include ESXi. Some applications of this kind are:

- SAN management software
- Storage resource management (SRM) software
- Snapshot software

- Replication software

Such software uses a physical compatibility mode for RDMs so that the software can access SCSI devices directly.

Various management products are best run centrally (not on the ESXi machine), while others run well on the virtual machines. VMware does not certify these applications or provide a compatibility matrix. To find out whether a SAN management application is supported in an ESXi environment, contact the SAN management software provider.

RDM Considerations and Limitations

Certain considerations and limitations exist when you use RDMs.

- The RDM is not available for direct-attached block devices or certain RAID devices. The RDM uses a SCSI serial number to identify the mapped device. Because block devices and some direct-attach RAID devices do not export serial numbers, they cannot be used with RDMs.
- If you are using the RDM in physical compatibility mode, you cannot use a snapshot with the disk. Physical compatibility mode allows the virtual machine to manage its own, storage-based, snapshot or mirroring operations.

Virtual machine snapshots are available for RDMs with virtual compatibility mode.

- You cannot map to a disk partition. RDMs require the mapped device to be a whole LUN.
- If you use vMotion to migrate virtual machines with RDMs, make sure to maintain consistent LUN IDs for RDMs across all participating ESXi hosts.
- Flash Read Cache does not support RDMs in physical compatibility. Virtual compatibility RDMs are supported with Flash Read Cache.

Raw Device Mapping Characteristics

An RDM is a special mapping file in a VMFS volume that manages metadata for its mapped device. The mapping file is presented to the management software as an ordinary disk file, available for the usual file-system operations. To the virtual machine, the storage virtualization layer presents the mapped device as a virtual SCSI device.

Key contents of the metadata in the mapping file include the location of the mapped device (name resolution), the locking state of the mapped device, permissions, and so on.

RDM Virtual and Physical Compatibility Modes

You can use RDMs in virtual compatibility or physical compatibility modes. Virtual mode specifies full virtualization of the mapped device. Physical mode specifies minimal SCSI virtualization of the mapped device, allowing the greatest flexibility for SAN management software.

In virtual mode, the VMkernel sends only READ and WRITE to the mapped device. The mapped device appears to the guest operating system exactly the same as a virtual disk file in a VMFS volume. The real hardware characteristics are hidden. If you are using a raw disk in virtual mode, you can realize the benefits of VMFS such as advanced file locking for data protection and snapshots for streamlining development processes. Virtual mode is also more portable across storage hardware than physical mode, presenting the same behavior as a virtual disk file.

In physical mode, the VMkernel passes all SCSI commands to the device, with one exception: the REPORT LUNs command is virtualized so that the VMkernel can isolate the LUN to the owning virtual machine. Otherwise, all physical characteristics of the underlying hardware are exposed. Physical mode is useful to run SAN management agents or other SCSI target-based software in the virtual machine. Physical mode also allows virtual-to-physical clustering for cost-effective high availability.

VMFS5 supports greater than 2TB disk size for RDMs in virtual and physical modes. You cannot relocate larger than 2TB RDMs to datastores other than VMFS5.

Dynamic Name Resolution

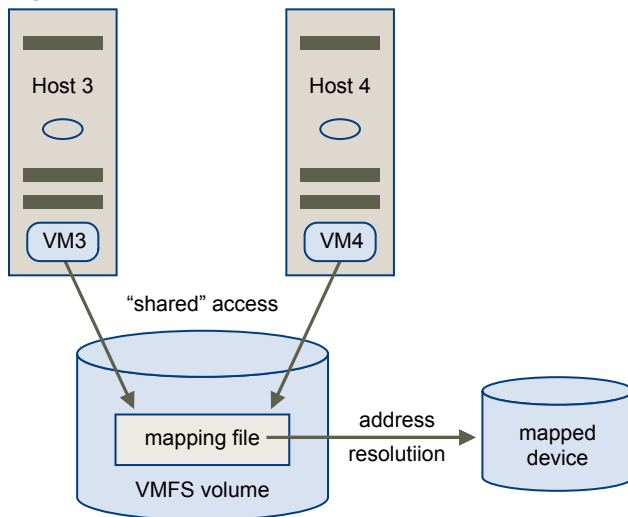
The RDM file supports dynamic name resolution when a path to a raw device changes.

VMFS uniquely identifies all mapped storage devices, and the identification is stored in its internal data structures. Any change in the path to a raw device, such as a Fibre Channel switch failure or the addition of a new HBA, can change the device name. Dynamic name resolution resolves these changes and automatically associates the original device with its new name.

Raw Device Mapping with Virtual Machine Clusters

Use an RDM with virtual machine clusters that need to access the same raw LUN for failover scenarios. The setup is similar to that of a virtual machine cluster that accesses the same virtual disk file, but an RDM replaces the virtual disk file.

Figure 18-3. Access from Clustered Virtual Machines



Comparing Available SCSI Device Access Modes

The ways of accessing a SCSI-based storage device include a virtual disk file on a VMFS datastore, virtual mode RDM, and physical mode RDM.

To help you choose among the available access modes for SCSI devices, the following table provides a quick comparison of features available with the different modes.

Table 18-1. Features Available with Virtual Disks and Raw Device Mappings

ESXi Features	Virtual Disk File	Virtual Mode RDM	Physical Mode RDM
SCSI Commands Passed Through	No	No	Yes REPORT LUNs is not passed through
vCenter Server Support	Yes	Yes	Yes

Table 18-1. Features Available with Virtual Disks and Raw Device Mappings (Continued)

ESXi Features	Virtual Disk File	Virtual Mode RDM	Physical Mode RDM
Snapshots	Yes	Yes	No
Distributed Locking	Yes	Yes	Yes
Clustering	Cluster-in-a-box only	Cluster-in-a-box cluster-across-boxes	Physical-to-virtual clustering cluster-across-boxes
SCSI Target-Based Software	No	No	Yes

VMware recommends that you use virtual disk files for the cluster-in-a-box type of clustering. If you plan to reconfigure your cluster-in-a-box clusters as cluster-across-boxes clusters, use virtual mode RDMs for the cluster-in-a-box clusters.

Create Virtual Machines with RDMs

When you give your virtual machine direct access to a raw SAN LUN, you create an RDM disk that resides on a VMFS datastore and points to the LUN. You can create the RDM as an initial disk for a new virtual machine or add it to an existing virtual machine. When creating the RDM, you specify the LUN to be mapped and the datastore on which to put the RDM.

Although the RDM disk file has the same `.vmdk` extension as a regular virtual disk file, the RDM contains only mapping information. The actual virtual disk data is stored directly on the LUN.

This procedure assumes that you are creating a new virtual machine. For information, see the *vSphere Virtual Machine Administration* documentation.

Procedure

- 1 Right-click any inventory object that is a valid parent object of a virtual machine, such as a data center, folder, cluster, resource pool, or host, and select **New Virtual Machine**.
- 2 Select **Create a new virtual machine** and click **Next**.
- 3 Follow the steps required to create a virtual machine.
- 4 On the Customize Hardware page, click the **Virtual Hardware** tab.
- 5 (Optional) To delete the default virtual hard disk that the system created for your virtual machine, move your cursor over the disk and click the **Remove** icon.
- 6 From the **New** drop-down menu at the bottom of the page, select **RDM Disk** and click **Add**.
- 7 From the list of SAN devices or LUNs, select a raw LUN for your virtual machine to access directly and click **OK**.

The system creates an RDM disk that maps your virtual machine to the target LUN. The RDM disk is shown on the list of virtual devices as a new hard disk.

- 8 Click the **New Hard Disk** triangle to expand the properties for the RDM disk.
- 9 Select a location for the RDM disk.

You can place the RDM on the same datastore where your virtual machine configuration files reside, or select a different datastore.

NOTE To use vMotion for virtual machines with enabled NPIV, make sure that the RDM files and the virtual machine files are located on the same datastore. You cannot perform Storage vMotion when NPIV is enabled.

- 10 Select a compatibility mode.

Option	Description
Physical	Allows the guest operating system to access the hardware directly. Physical compatibility is useful if you are using SAN-aware applications on the virtual machine. However, a virtual machine with a physical compatibility RDM cannot be cloned, made into a template, or migrated if the migration involves copying the disk.
Virtual	Allows the RDM to behave as if it were a virtual disk, so you can use such features as taking snapshots, cloning, and so on. When you clone the disk or make a template out of it, the contents of the LUN are copied into a .vmdk virtual disk file. When you migrate a virtual compatibility mode RDM, you can migrate the mapping file or copy the contents of the LUN into a virtual disk.

- 11 If you selected virtual compatibility mode, select a disk mode.

Disk modes are not available for RDM disks using physical compatibility mode.

Option	Description
Dependent	Dependent disks are included in snapshots.
Independent - Persistent	Disks in persistent mode behave like conventional disks on your physical computer. All data written to a disk in persistent mode are written permanently to the disk.
Independent - Nonpersistent	Changes to disks in nonpersistent mode are discarded when you power off or reset the virtual machine. With nonpersistent mode, you can restart the virtual machine with a virtual disk in the same state every time. Changes to the disk are written to and read from a redo log file that is deleted when you power off or reset.

- 12 Click **OK**.

Manage Paths for a Mapped LUN

When you use virtual machines with RDMs, you can manage paths for mapped raw LUNs.

Procedure

- 1 In the vSphere Web Client, browse to the virtual machine.
- 2 Right-click the virtual machine and select **Edit Settings**.
- 3 Click the **Virtual Hardware** tab and click **Hard Disk** to expand the disk options menu.
- 4 Click **Manage Paths**.
- 5 Use the Edit Multipathing Policies dialog box to enable or disable paths, set multipathing policy, and specify the preferred path.

For information on managing paths, see [Chapter 17, “Understanding Multipathing and Failover,”](#) on page 181.

Working with Virtual Volumes

The Virtual Volumes functionality changes the storage management paradigm from managing space inside datastores to managing abstract storage objects handled by storage arrays. With Virtual Volumes, an individual virtual machine, not the datastore, becomes a unit of storage management, while storage hardware gains complete control over virtual disk content, layout, and management.

Historically, vSphere storage management used a datastore-centric approach. With this approach, storage administrators and vSphere administrators discuss in advance the underlying storage requirements for virtual machines. The storage administrator then sets up LUNs or NFS shares and presents them to ESXi hosts. The vSphere administrator creates datastores based on LUNs or NFS, and uses these datastores as virtual machine storage. Typically, the datastore is the lowest granularity level at which data management occurs from a storage perspective. However, a single datastore contains multiple virtual machines, which might have different requirements. With the traditional approach, differentiation on a per virtual machine level is difficult.

The Virtual Volumes functionality helps to improve granularity and allows you to differentiate virtual machine services on a per application level by offering a new approach to storage management. Rather than arranging storage around features of a storage system, Virtual Volumes arranges storage around the needs of individual virtual machines, making storage virtual-machine centric.

Virtual Volumes maps virtual disks and their derivatives, clones, snapshots, and replicas, directly to objects, called virtual volumes, on a storage system. This mapping allows vSphere to offload intensive storage operations such as snapshot, cloning, and replication to the storage system.

By creating a volume for each virtual disk, you can set policies at the optimum level. You can decide in advance what the storage requirements of an application are, and communicate these requirements to the storage system, so that it creates an appropriate virtual disk based on these requirements. For example, if your virtual machine requires an active-active storage array, you no longer must select a datastore that supports the active-active model, but instead, you create an individual virtual volume that will be automatically placed to the active-active array.

This chapter includes the following topics:

- [“Virtual Volumes Concepts,”](#) on page 212
- [“Guidelines when Using Virtual Volumes,”](#) on page 215
- [“Virtual Volumes and Storage Protocols,”](#) on page 215
- [“Virtual Volumes Architecture,”](#) on page 216
- [“Before You Enable Virtual Volumes,”](#) on page 217
- [“Configure Virtual Volumes,”](#) on page 218
- [“Provision Virtual Machines on Virtual Datastores,”](#) on page 221

Virtual Volumes Concepts

With Virtual Volumes, abstract storage containers replace traditional storage volumes based on LUNs or NFS shares. In vCenter Server, the storage containers are represented by virtual datastores. Virtual datastores remove artificial boundaries of traditional datastores and are used to store virtual volumes, objects that encapsulate virtual machine files.

Watch the video to learn more about different components of the Virtual Volumes functionality.



Virtual Volumes Part 1: Concepts

(http://link.brightcove.com/services/player/bcpid2296383276001?bctid=ref:video_vvols_part1_concepts)

- [Virtual Volumes](#) on page 212
Virtual volumes are encapsulations of virtual machine files, virtual disks, and their derivatives.
- [Virtual Volumes and Storage Providers](#) on page 213
A Virtual Volumes storage provider, also called a VASA provider, is a software component that acts as a storage awareness service for vSphere. The provider mediates out-of-band communication between vCenter Server and ESXi hosts on one side and a storage system on the other.
- [Storage Containers](#) on page 213
Unlike traditional LUN and NFS based vSphere storage, the Virtual Volumes functionality does not require preconfigured volumes on a storage side. Instead, Virtual Volumes uses a storage container, which is a pool of raw storage capacity or an aggregation of storage capabilities that a storage system can provide to virtual volumes.
- [Protocol Endpoints](#) on page 214
Although storage systems manage all aspects of virtual volumes, ESXi hosts have no direct access to virtual volumes on the storage side. Instead, ESXi hosts use a logical I/O proxy, called the protocol endpoint, to communicate with virtual volumes and virtual disk files that virtual volumes encapsulate. ESXi uses protocol endpoints to establish a data path on demand from virtual machines to their respective virtual volumes.
- [Virtual Datastores](#) on page 214
A virtual datastore represents a storage container in vCenter Server and the vSphere Web Client.

Virtual Volumes

Virtual volumes are encapsulations of virtual machine files, virtual disks, and their derivatives.

Virtual volumes are stored natively inside a storage system that is connected through Ethernet or SAN. They are exported as objects by a compliant storage system and are managed entirely by hardware on the storage side. Typically, a unique GUID identifies a virtual volume. Virtual volumes are not preprovisioned, but created automatically when you perform virtual machine management operations. These operations include a VM creation, cloning, and snapshotting. ESXi and vCenter Server associate one or more virtual volumes to a virtual machine. The system creates the following types of virtual volumes for the core elements that make up the virtual machine:

- A data virtual volume that corresponds directly to each virtual disk `.vmdk` file. As virtual disk files on traditional datastores, virtual volumes are presented to virtual machines as SCSI disks.
- A configuration virtual volume, or a home directory, represents a small directory that contains metadata files for a virtual machine. The files include a `.vmx` file, descriptor files for virtual disks, log files, and so forth. The configuration virtual volume is formatted with a file system. When ESXi uses SCSI protocol to connect to storage, configuration virtual volumes are formatted with VMFS. With NFS protocol, configuration virtual volumes are presented as an NFS directory.

Additional virtual volumes can be created for other virtual machine components and virtual disk derivatives, such as clones, snapshots, and replicas. These virtual volumes include a swap virtual volume to hold virtual machine swap files and a virtual memory volume to hold the contents of virtual machine memory for a snapshot.

By using different virtual volumes for different VM components, you can apply and manipulate storage policies at the finest granularity level. For example, a virtual volume that contains a virtual disk can have a richer set of data services and performance levels than the virtual volume for the VM boot disk. Similarly, a snapshot virtual volume can use a different storage tier compared to a current virtual volume.

Virtual Volumes and Storage Providers

A Virtual Volumes storage provider, also called a VASA provider, is a software component that acts as a storage awareness service for vSphere. The provider mediates out-of-band communication between vCenter Server and ESXi hosts on one side and a storage system on the other.

The storage provider is implemented through VMware APIs for Storage Awareness (VASA) and is used to manage all aspects of Virtual Volumes storage. The storage provider integrates with the Storage Monitoring Service (SMS), shipped with vSphere, to communicate with vCenter Server and ESXi hosts.

The storage provider delivers information from the underlying storage, or storage container in the case of Virtual Volumes, so that storage container capabilities can appear in vCenter Server and the vSphere Web Client. Then, in turn, the storage provider communicates virtual machine storage requirements, which you can define in the form of a storage policy, to the storage layer. This integration process ensures that a virtual volume created in the storage layer meets the requirements outlined in the policy.

Typically, vendors are responsible for supplying storage providers that can integrate with vSphere and provide support to Virtual Volumes. Every storage provider must be certified by VMware and properly deployed. For information about deploying the Virtual Volumes storage provider, contact your storage vendor.

After you deploy the storage provider, you must register it in vCenter Server, so that it can communicate with vSphere through the SMS.

Storage Containers

Unlike traditional LUN and NFS based vSphere storage, the Virtual Volumes functionality does not require preconfigured volumes on a storage side. Instead, Virtual Volumes uses a storage container, which is a pool of raw storage capacity or an aggregation of storage capabilities that a storage system can provide to virtual volumes.

A storage container is a part of the logical storage fabric and is a logical unit of the underlying hardware. The storage container logically groups virtual volumes based on management and administrative needs. For example, the storage container can contain all virtual volumes created for a tenant in a multitenant deployment, or a department in an enterprise deployment. Each storage container serves as a virtual volume store and virtual volumes are allocated out of the storage container capacity.

Typically, a storage administrator on the storage side defines storage containers. The number of storage containers, their capacity and size depend on a vendor-specific implementation, but at least one container for each storage system is required.

NOTE A single storage container cannot span different physical arrays.

After you register a storage provider associated with the storage system, vCenter Server discovers all configured storage containers along with their storage capability profiles, protocol endpoints, and other attributes. A single storage container can export multiple capability profiles. As a result, virtual machines with diverse needs and different storage policy settings can be a part of the same storage container.

Initially, all discovered storage containers are not connected to any specific host, and you cannot see them in the vSphere Web Client. To mount a storage container, you must map it to a virtual datastore.

Protocol Endpoints

Although storage systems manage all aspects of virtual volumes, ESXi hosts have no direct access to virtual volumes on the storage side. Instead, ESXi hosts use a logical I/O proxy, called the protocol endpoint, to communicate with virtual volumes and virtual disk files that virtual volumes encapsulate. ESXi uses protocol endpoints to establish a data path on demand from virtual machines to their respective virtual volumes.

Each virtual volume is bound to a specific protocol endpoint. When a virtual machine on the host performs an I/O operation, the protocol endpoint directs the I/O to the appropriate virtual volume. Typically, a storage system requires a very small number of protocol endpoints. A single protocol endpoint can connect to hundreds or thousands of virtual volumes.

On the storage side, a storage administrator configures protocol endpoints, one or several per storage container. Protocol endpoints are a part of the physical storage fabric and are exported, along with associated storage containers, by the storage system through a storage provider. After you map a storage container to a virtual datastore, protocol endpoints are discovered by ESXi and become visible in the vSphere Web Client. Protocol endpoints can also be discovered during a storage rescan.

In the vSphere Web Client, the list of available protocol endpoints looks similar to the host storage devices list. Different storage transports can be used to expose protocol endpoints to ESXi. When the SCSI-based transport is used, the protocol endpoint represents a proxy LUN defined by a T10-based LUN WWN. For the NFS protocol, the protocol endpoint is a mount-point, such as IP address (or DNS name) and a share name. You can configure multipathing on a SCSI based protocol endpoint, but not on an NFS based protocol endpoint. However, no matter which protocol you use, a storage array can provide multiple protocol endpoints for availability purposes.

Virtual Datastores

A virtual datastore represents a storage container in vCenter Server and the vSphere Web Client.

After vCenter Server discovers storage containers exported by storage systems, you must mount them to be able to use them. You use the datastore creation wizard in the vSphere Web Client to map a storage container to a virtual datastore. The virtual datastore that you create corresponds directly to the specific storage container and becomes the container's representation in vCenter Server and the vSphere Web Client.

From a vSphere administrator prospective, the virtual datastore is similar to any other datastore and is used to hold virtual machines. Like other datastores, the virtual datastore can be browsed and lists virtual volumes by virtual machine name. Like traditional datastores, the virtual datastore supports unmounting and mounting. However, such operations as upgrade and resize are not applicable to the virtual datastore. The virtual datastore capacity is configurable by the storage administrator outside of vSphere.

You can use virtual datastores with traditional VMFS and NFS datastores and with Virtual SAN.

NOTE The size of a virtual volume must be a multiple of 1 MB, with a minimum size of 1 MB. As a result, all virtual disks that you provision on a virtual datastore or migrate from any datastore other than the virtual datastore should be an even multiple of 1 MB in size. If the virtual disk you migrate to the virtual datastore is not an even multiple of 1 MB, extend the disk manually to the nearest even multiple of 1 MB.

Guidelines when Using Virtual Volumes

The Virtual Volumes functionality offers several benefits and advantages. When you work with Virtual Volumes, you must follow specific guidelines.

Virtual Volumes has the following characteristics:

- Virtual Volumes supports offloading a number of operations to storage hardware. These operations include snapshotting, cloning, and Storage DRS.
- With Virtual Volumes, you can use advanced storage services that include replication, encryption, deduplication, and compression on individual virtual disks.
- Virtual Volumes supports such vSphere features as vMotion, Storage vMotion, snapshots, linked clones, Flash Read Cache, and DRS.
- With Virtual Volumes, storage vendors can use native snapshot facilities to improve performance of vSphere snapshots.
- You can use Virtual Volumes with storage arrays that support vSphere APIs for Array Integration (VAAI).
- Virtual Volumes supports backup software that uses vSphere APIs for Data Protection (VADP).

Virtual Volumes Guidelines and Limitations

Follow these guidelines when you use Virtual Volumes.

- Because the Virtual Volumes environment requires vCenter Server, you cannot use Virtual Volumes with a standalone host.
- Virtual Volumes does not support RDMs.
- A Virtual Volumes storage container cannot span across different physical arrays.
- Host profiles that contain virtual datastores are vCenter Server specific. After you extract this type of host profile, you can attach it only to hosts and clusters managed by the same vCenter Server as the reference host.

Virtual Volumes and Storage Protocols

The Virtual Volumes functionality supports Fibre Channel, FCoE, iSCSI, and NFS. Storage transports expose protocol endpoints to ESXi hosts.

When the SCSI-based protocol is used, the protocol endpoint represents a LUN defined by a T10-based LUN WWN. For the NFS protocol, the protocol endpoint is a mount point, such as IP address or DNS name and a share name.

No matter what storage protocol is used, a virtual volume, like a files on other traditional datastores, is presented to a virtual machine as a SCSI disk. Virtual Volumes on disk arrays supports the same set of SCSI commands as VMFS and uses ATS as a locking mechanism.

Virtual Volumes on NAS devices supports the same NFS Remote Procedure Calls (RPCs) that are required by ESXi hosts to connect to NFS mount points.

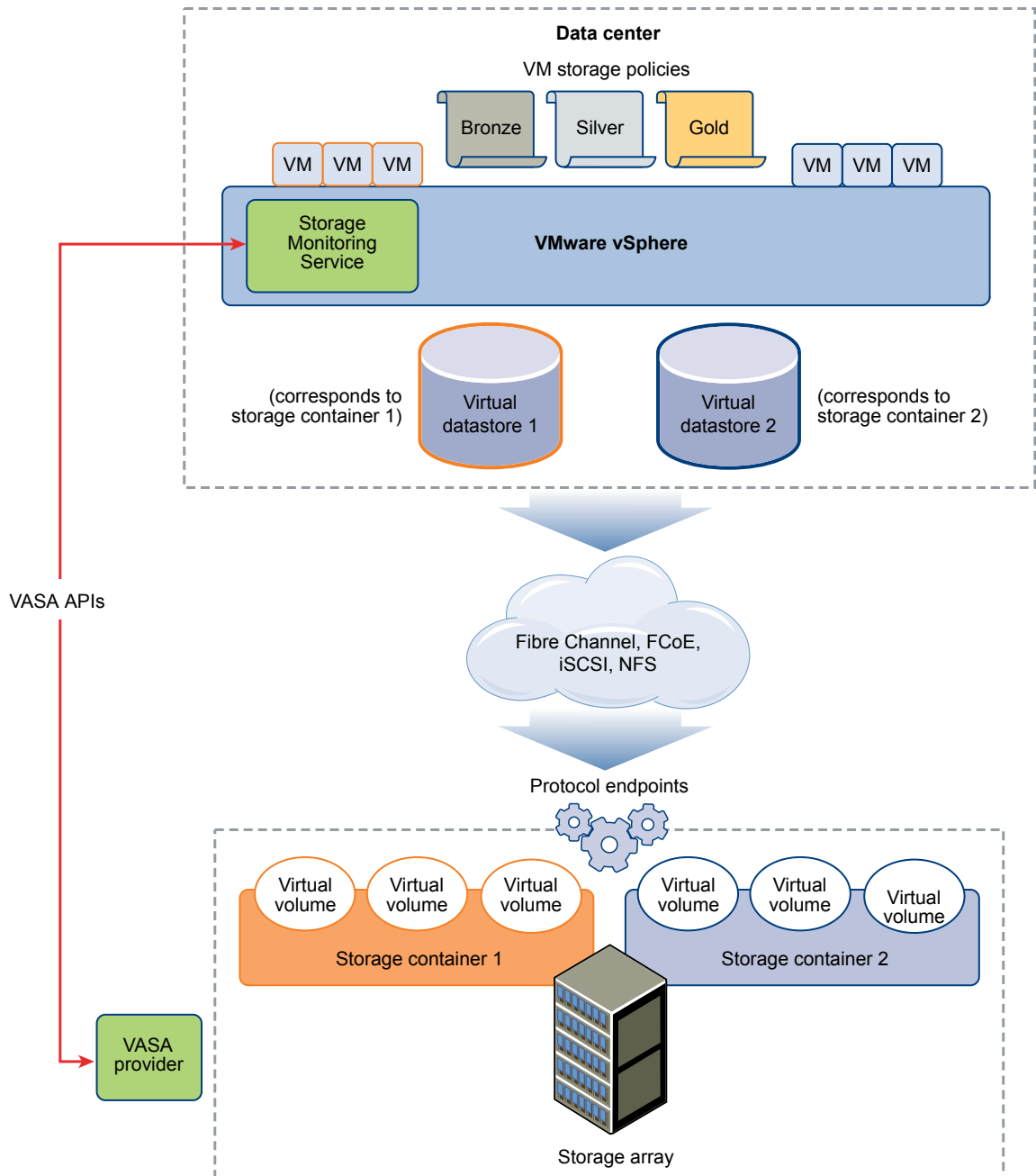
These considerations and guidelines apply when you use different storage protocols:

- With NFS, you can use version 3. Virtual Volumes does not support NFS 4.1.
- IPv6 format is not supported.
- For iSCSI, you must activate the software iSCSI adapter. Configure Dynamic Discovery and enter the IP address of the Virtual Volumes Storage Provider. See [“Configure the Software iSCSI Adapter,”](#) on page 81.

- You can configure multipathing on a SCSI-based protocol endpoint, but not on an NFS-based protocol endpoint. No matter which protocol you use, a storage array can provide multiple protocol endpoints for availability purposes.

Virtual Volumes Architecture

An architectural diagram provides an overview of how all components of the Virtual Volumes functionality interact with each other.



Virtual volumes are objects exported by a compliant storage system and typically correspond one-to-one with a virtual machine disk and other VM-related files. A virtual volume is created and manipulated out-of-band, not in the data path, by a VASA provider.

A VASA provider, or a storage provider, is developed through vSphere APIs for Storage Awareness. The storage provider enables communication between the vSphere stack — ESXi hosts, vCenter server, and the vSphere Web Client — on one side, and the storage system on the other. The VASA provider runs on the storage side and integrates with vSphere Storage Monitoring Service (SMS) to manage all aspects of Virtual Volumes storage. The VASA provider maps virtual disk objects and their derivatives, such as clones, snapshots, and replicas, directly to virtual volumes on the storage system.

ESXi hosts have no direct access to the virtual volumes storage. Instead, the hosts access virtual volumes through an intermediate point in the data path, called the protocol endpoint. Protocol endpoints establish a data path on demand from virtual machines to their respective virtual volumes and serve as a gateway for direct in-band I/O between ESXi hosts and the storage system. ESXi can use Fibre Channel, FCoE, iSCSI, and NFS protocols for in-band communication.

Virtual volumes reside inside storage containers that logically represent a pool of physical disks on the storage system. In the vSphere stack, storage containers are presented as virtual datastores. A single storage container can export multiple storage capability sets. As a result, when you create a virtual machine on the virtual datastore, you can use different storage policies to place virtual volumes inside the same storage container in such a way that diverse storage needs of a virtual machine are met.

Watch the video for information about Virtual Volumes architecture.



Virtual Volumes Part 2: Architecture

(http://link.brightcove.com/services/player/bcpid2296383276001?bctid=ref:video_vvols_part2_architecture)

Before You Enable Virtual Volumes

To work with Virtual Volumes, you must make sure that your storage and vSphere environment are set up correctly.

Follow these guidelines to prepare your storage system environment for Virtual Volumes. For additional information, contact your storage vendor.

- The storage system or storage array that you use must be able to support Virtual Volumes and integrate with vSphere through vSphere APIs for Storage Awareness (VASA).
- A Virtual Volumes storage provider must be deployed.
- Protocol endpoints, storage containers, and storage profiles must be configured on the storage side.

Prepare your vSphere environment.

- Make sure to follow appropriate setup guidelines for the type of storage you use, Fibre Channel, FCoE, iSCSI, or NFS. If required, install and configure storage adapters on your ESXi hosts.

If you use iSCSI, activate the software iSCSI adapters on your ESXi hosts. Configure Dynamic Discovery and enter the IP address of your Virtual Volumes storage system.

- Synchronize all components in the storage array with vCenter Server and all ESXi hosts. Use Network Time Protocol (NTP) to do this synchronization.

Synchronize vSphere Storage Environment with a Network Time Server

Before you enable Virtual Volumes, make sure all machines on your vSphere network have their clocks synchronized.

Procedure

- 1 Select the host in the vSphere inventory.
- 2 Click the **Manage** tab and click **Settings**.
- 3 In the System section, select **Time Configuration**.

- 4 Click **Edit** and set up the NTP server.
 - a Select **Use Network Time Protocol (Enable NTP client)**.
 - b Set the NTP Service Startup Policy.
 - c Enter the IP addresses of the NTP servers to synchronize with.
 - d Click **Start** or **Restart** in the NTP Service Status section.
- 5 Click **OK**.

The host synchronizes with the NTP server.

Configure Virtual Volumes

You must complete a series of steps to configure your Virtual Volumes environment.

Prerequisites

Follow guidelines in [“Before You Enable Virtual Volumes,”](#) on page 217.

Procedure

- 1 [Register Storage Providers for Virtual Volumes](#) on page 219
Your Virtual Volumes environment must include storage providers, also called VASA providers. Typically, third-party vendors develop storage providers through the VMware APIs for Storage Awareness (VASA). Storage providers facilitate communication between vSphere and the storage side. You must register the storage provider in vCenter Server to be able to work with Virtual Volumes.
- 2 [Create a Virtual Datastore](#) on page 219
You use the New Datastore wizard to create a virtual datastore.
- 3 [Review and Manage Protocol Endpoints](#) on page 220
ESXi hosts use a logical I/O proxy, called protocol endpoint, to communicate with virtual volumes and virtual disk files that virtual volumes encapsulate. Protocol endpoints are exported, along with associated storage containers, by the storage system through a storage provider. Protocol endpoints become visible in the vSphere Web Client after you map a storage container to a virtual datastore. You can review properties of protocol endpoints and modify specific settings.
- 4 [\(Optional\) Change the Path Selection Policy for a Protocol Endpoint](#) on page 220
If your ESXi host uses SCSI-based transport to communicate with protocol endpoints representing a storage array, you can modify default multipathing policies assigned to protocol endpoints. Use the Edit Multipathing Policies dialog box to change a path selection policy.

What to do next

You can now provision virtual machines on the virtual datastore. For information on creating virtual machines, see the *vSphere Virtual Machine Administration* documentation.

Register Storage Providers for Virtual Volumes

Your Virtual Volumes environment must include storage providers, also called VASA providers. Typically, third-party vendors develop storage providers through the VMware APIs for Storage Awareness (VASA). Storage providers facilitate communication between vSphere and the storage side. You must register the storage provider in vCenter Server to be able to work with Virtual Volumes.

After registration, the Virtual Volumes provider communicates with vCenter Server and reports characteristics of underlying storage. The characteristics appear in the VM Storage Policies interface and can be used to create a VM storage policy compatible with the virtual datastore. After you apply this storage policy to a virtual machine, the policy is pushed to Virtual Volumes storage. The policy enforces optimal placement of the virtual machine within Virtual Volumes storage and guarantees that storage can satisfy virtual machine requirements.

Prerequisites

Verify that the storage provider component is installed on the storage side and obtain its credentials from your storage administrator. For information, contact your vendor.

Procedure

- 1 Browse to vCenter Server in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage Providers**.
- 3 Click the **Register a new storage provider** icon.
- 4 (Optional) To direct vCenter Server to the storage provider certificate, select the **Use storage provider certificate** option and specify the certificate's location.

If you do not select this option, a thumbprint of the certificate is displayed. You can check the thumbprint and approve it.

- 5 Click **OK** to complete the registration.

vCenter Server discovers and registers the Virtual Volumes storage provider.

Create a Virtual Datastore

You use the New Datastore wizard to create a virtual datastore.

Procedure

- 1 In the vSphere Web Client navigator, select **vCenter Inventory Lists > Datastores**
- 2 Click the Create a New Datastore icon.
- 3 Type the datastore name and if required, select the placement location for the datastore.

Make sure to use the name that does not duplicate another datastore name in your data center environment.

If you mount the same virtual datastore to several hosts, the name of the datastore must be consistent across all hosts.

- 4 Select **VVOL** as the datastore type.
- 5 From the list of storage containers, select a backing storage container.
- 6 Select the hosts that require access to the datastore.
- 7 Review the configuration options and click **Finish**.

What to do next

After you create the virtual datastore, you can perform such datastore operations as renaming the datastore, browsing datastore files, unmounting the datastore, and so on.

You cannot add the virtual datastore to a datastore cluster.

Review and Manage Protocol Endpoints

ESXi hosts use a logical I/O proxy, called protocol endpoint, to communicate with virtual volumes and virtual disk files that virtual volumes encapsulate. Protocol endpoints are exported, along with associated storage containers, by the storage system through a storage provider. Protocol endpoints become visible in the vSphere Web Client after you map a storage container to a virtual datastore. You can review properties of protocol endpoints and modify specific settings.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Protocol Endpoints**.
- 4 To view details for a specific item, select this item from the list.
- 5 Use tabs under Protocol Endpoint Details to access additional information and modify properties for the selected protocol endpoint.

Tab	Description
Properties	View the item properties and characteristics. For SCSI (block) items, view and edit multipathing policies.
Paths (SCSI protocol endpoints only)	Display paths available for the protocol endpoint. Disable or enable a selected path. Change the Path Selection Policy.
Datastores	Display a corresponding virtual datastore. Perform datastore management operations.

(Optional) Change the Path Selection Policy for a Protocol Endpoint

If your ESXi host uses SCSI-based transport to communicate with protocol endpoints representing a storage array, you can modify default multipathing policies assigned to protocol endpoints. Use the Edit Multipathing Policies dialog box to change a path selection policy.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage**.
- 3 Click **Protocol Endpoints**.
- 4 Select the protocol endpoint whose paths you want to change and click the **Properties** tab.
- 5 Under Multipathing Policies, click **Edit Multipathing**.
- 6 Select a path policy.
 - Fixed (VMware)
 - Most Recently Used (VMware)
 - Round Robin (VMware)
- 7 For the fixed policy, specify the preferred path.

- 8 Click **OK** to save your settings and exit the dialog box.

Provision Virtual Machines on Virtual Datastores

You can provision virtual machines on a virtual datastore.

NOTE All virtual disks that you provision on a virtual datastore should be an even multiple of 1 MB.

A virtual machine that runs on a virtual datastore requires an appropriate VM storage policy.

VM storage policies are sets of rules that capture the quality of service requirements for a virtual machine. During virtual machine provisioning, storage policy management matches virtual machine requirements to data services offered by Virtual Volumes storage. In addition to placement decisions, storage policy management performs the following operations:

- Admission control
- Quality of service compliance monitoring
- Dynamic resource allocation that includes CPU, memory, network, and storage

If you do not create the VM storage policy compatible with the virtual datastore, the system uses default No Requirements policy. The No Requirements policy is a generic Virtual Volumes policy that contains no rules or storage specifications. The policy allows Virtual Volumes storage arrays to determine the optimum placement for the VM objects.

If your virtual machine has specific storage requirements, you can create a Virtual Volumes storage policy to define these requirements. When you assign the new policy to the virtual machine, the policy enforces that the Virtual Volumes storage meets the requirements. You can also use the new policy as default.

Procedure

- 1 [Define a VM Storage Policy for Virtual Volumes](#) on page 221
You can create a VM storage policy compatible with a virtual datastore.
- 2 [Assign the Virtual Volumes Storage Policy to Virtual Machines](#) on page 222
To guarantee that the virtual datastore fulfills specific storage requirements when allocating a virtual machine, associate the Virtual Volumes storage policy with the virtual machine.
- 3 [Change Default Storage Policy for a Virtual Datastore](#) on page 223
For virtual machines provisioned on virtual datastores, VMware provides a default No Requirements policy. You cannot edit this policy, but you can designate a newly created policy as default.

Define a VM Storage Policy for Virtual Volumes

You can create a VM storage policy compatible with a virtual datastore.

Prerequisites

- Verify that the Virtual Volumes storage provider is available and active. See [“Register Storage Providers for Virtual Volumes,”](#) on page 219.
- Ensure that the virtual machine storage policies are enabled on all hosts where the virtual datastore is mounted. See [“Enable Virtual Machine Storage Policies,”](#) on page 228.

Procedure

- 1 From the vSphere Web Client Home, click **Policies and Profiles > VM Storage Policies**.
- 2 Click the **Create a new VM storage policy** icon.
- 3 Select the vCenter Server instance.

- 4 Type a name and a description for the storage policy and click **Next**.
- 5 On the Rule Set page, select the Virtual Volumes storage provider from the **Rules based on data services** drop-down menu.

The page expands to show data services provided by the Virtual Volumes storage resource.

- 6 Select a data service to include and specify its value.

Make sure that the values you provide are within the range of values advertised by the Virtual Volumes profile.

Based on your input, the storage consumption mechanism calculates the amount of space required for a virtual disk that will reside on the virtual datastore.

- 7 Complete the creation of the storage policy and click **Finish**.

The new VM storage policy compatible with Virtual Volumes appears on the list.

What to do next

You can now associate this policy with a virtual machine, or designate the policy as default.

Assign the Virtual Volumes Storage Policy to Virtual Machines

To guarantee that the virtual datastore fulfills specific storage requirements when allocating a virtual machine, associate the Virtual Volumes storage policy with the virtual machine.

You can assign the Virtual Volumes storage policy during an initial deployment of a virtual machine, or when performing other virtual machine operations, such as cloning or migrating. This topic describes how to assign the Virtual Volumes storage policy when you create a new virtual machine. For information about other VM provisioning methods, see the *vSphere Virtual Machine Administration* documentation.

You can apply the same storage policy to the virtual machine configuration file and all its virtual disks. If storage requirements for your virtual disks and the configuration file are different, you can associate different storage policies with the VM configuration file and the selected virtual disks.

Procedure

- 1 In the vSphere Web Client, start the virtual machine provisioning process and follow appropriate steps.
- 2 Assign the same storage policy to all virtual machine files and disks.
 - a On the Select Storage page, select the storage policy compatible with Virtual Volumes, for example VVols Silver, from the **VM Storage Policy** drop-down menu.
 - b Select the virtual datastore from the list of available datastores and click **Next**.

The datastore becomes the destination storage resource for the virtual machine configuration file and all virtual disks.

- 3 Change the storage policy for the virtual disk.

Use this option if requirements for storage placement are different for virtual disks.

- a On the Customize Hardware page, expand the New hard disk pane.
- b From the **VM storage policy** drop-down menu, select the appropriate storage policy, for example VVols Gold, that you want to assign to the virtual disk.

- 4 Complete the virtual machine provisioning process.

After you create the virtual machine, the **Summary** tab displays the assigned storage policies and their compliance status.

What to do next

If storage placement requirements for the configuration file or the virtual disks change, you can later modify the virtual policy assignment. See [“Change Storage Policy Assignment for Virtual Machine Files and Disks,”](#) on page 235.

Change Default Storage Policy for a Virtual Datastore

For virtual machines provisioned on virtual datastores, VMware provides a default No Requirements policy. You cannot edit this policy, but you can designate a newly created policy as default.

Prerequisites

Create a storage policy compatible with Virtual Volumes.

Procedure

- 1 Browse to the virtual datastore whose default storage policy you want to change.
- 2 Click the **Manage** tab, and click **Settings**.
- 3 Click **General**, and click **Edit** in the Default Storage Policy pane.
- 4 From the list of available storage policies, select a policy that you want to designate as default and click **OK**.

The selected storage policy becomes the default policy for the virtual datastore. vSphere assigns this policy to any virtual machine objects that you provision on the virtual datastore when no other policy is explicitly selected.

Virtual Machine Storage Policies

Virtual machine storage policies are essential to virtual machine provisioning. These policies help you define storage requirements for the virtual machine and control which type of storage is provided for the virtual machine, how the virtual machine is placed within the storage, and which data services are offered for the virtual machine.

When you define a storage policy, you specify storage requirements for applications that run on virtual machines. After you apply this storage policy to a virtual machine, the virtual machine is placed in a specific datastore that can satisfy the storage requirements. In software-defined storage environments, such as Virtual SAN and Virtual Volumes, the storage policy also determines how the virtual machine storage objects are provisioned and allocated within the storage resource to guarantee the required level of service.

This chapter includes the following topics:

- [“Upgrading Legacy Storage Profiles,”](#) on page 225
- [“Understanding Virtual Machine Storage Policies,”](#) on page 226
- [“Working with Virtual Machine Storage Policies,”](#) on page 228
- [“Creating and Managing VM Storage Policies,”](#) on page 228
- [“Storage Policies and Virtual Machines,”](#) on page 233

Upgrading Legacy Storage Profiles

In vSphere 5.0 and 5.1, virtual machine storage policies were called storage profiles and had a different format. When you upgrade the vSphere environment from versions 5.0 or 5.1 to vSphere 6.0, storage profiles that you created in the earlier versions are converted to storage policies.

All the components of legacy storage profiles are transformed into new formats or objects. Any system-defined storage capabilities are converted into storage-specific data services based on metadata. User-defined capabilities become datastore tags.

Table 20-1. Old and New Formats of Storage Policies

vSphere 5.0 and 5.1	vSphere 6.0
Virtual machine storage profile	Virtual machine storage policy
System-defined capabilities	Storage-specific data services
User-defined capabilities	Datastore tags

Understanding Virtual Machine Storage Policies

Virtual machine storage policies capture storage characteristics that virtual machine home files and virtual disks require to run applications within the virtual machine. You can create several storage policies to define the types and classes of storage requirements.

Each storage policy is not only a set of constraints that apply simultaneously. A single policy can include alternative sets of subpolicies, or rule sets, that are datastore-specific and represent equally acceptable storage requirements.

When you create, clone, or migrate a virtual machine, you can apply the storage policy to the virtual machine. You can place the virtual machine in one of the datastores that matches the policy requirements. To match the policy requirement, the datastore must satisfy all rules in at least one of the datastore specific rule sets.

The virtual machine home files (.vmx, .vmsd, .nvram, .log, and so on) and the virtual disks (.vmdk) can have separate storage policies.

Table 20-2. Example Storage Policy for a Virtual Machine

Example Virtual Machine Files	Example for a Storage Policy	Example for a Datastore Compliant with the Storage Policy
<i>windows_2008r2_test.vmx</i>		
<i>windows_2008r2_test.vmx</i>		
<i>windows_2008r2_test.log</i>	Storage Policy 2	datastore02, datastore05, datastore10
<i>windows_2008r2_test.nvram</i>		
<i>windows_2008r2_test.vmem</i>		
<i>windows_2008r2_test.vmsd</i>		
<i>windows_2008r2_test.vmdk</i>	Storage Policy 3	datastore05
<i>windows_2008r2_test_1.vmdk</i>	Storage Policy 5	datastore10

Storage Policies and Rules

Rules that you include in a storage policy can be based on storage-specific data services or datastore tags.

- [Rules Based on Storage-Specific Data Services](#) on page 226

These rules are based on data services that storage entities such as Virtual SAN and Virtual Volumes advertise.

- [Rules Based on Tags](#) on page 227

Rules based on tags reference datastore tags that you associate with specific datastores. You can apply more than one tag to a datastore.

Rules Based on Storage-Specific Data Services

These rules are based on data services that storage entities such as Virtual SAN and Virtual Volumes advertise.

To supply information about underlying storage to vCenter Server, Virtual SAN and Virtual Volumes use storage providers, also called VASA providers. Storage information and datastore characteristics appear in the VM Storage Policies interface of the vSphere Web Client as data services offered by the specific datastore type.

A single datastore can offer multiple services. The data services are grouped in a datastore profile that outlines the quality of service that the datastore can deliver.

When you create rules for a VM storage policy, you reference data services that a specific datastore advertises. To the virtual machine that uses this policy, the datastore guarantees that it can satisfy the storage requirements of the virtual machine. The datastore also can provide the virtual machine with a specific set of characteristics for capacity, performance, availability, redundancy, and so on.

For more information about storage providers, see [Chapter 24, “Using Storage Providers,”](#) on page 261.

Rules Based on Tags

Rules based on tags reference datastore tags that you associate with specific datastores. You can apply more than one tag to a datastore.

Typically, tags serve the following purposes:

- Attach a broad storage-level definition to datastores that are not represented by any storage providers, for example, VMFS and NFS datastores.
- Encode policy-relevant information that is not advertised through vSphere API for Storage Awareness (VASA), such as geographical location or administrative group.

Similar to storage-specific services, all tags associated with datastores appear in the VM Storage Policies interface. You can use the tags when you define rules for the storage policies.

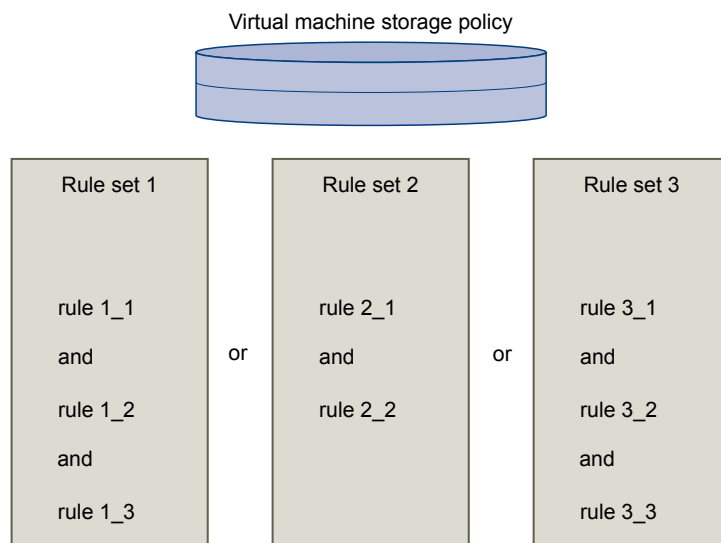
About Datastore-Specific Rule Sets

A storage policy can include one or several rule sets that describe requirements for virtual machine storage resources.

To define a policy, one rule set is required. Additional rule sets are optional. Multiple rule sets allow a single policy to define alternative selection parameters, often from several storage providers.

A single rule set contains one or several rules. Each rule can be based on a single underlying data service that a storage entity guarantees. The rule describes a specific quality or quantity that the storage resource must provide. You can also reference user-defined datastore tags in the rules. One datastore-specific rule set can include rules from only a single storage entity.

The relationship between all rule sets within a policy is defined by the Boolean operator OR, whereas the relationship between all rules within a single rule set is defined by AND. Meeting all the rules of any one rule set is sufficient to satisfy the entire policy. Each rule set represents an equally acceptable set of constraints.



Working with Virtual Machine Storage Policies

The entire process of creating and managing storage policies typically includes several steps. Whether you must perform a specific step might depend on the type of storage or data services that your environment offers.

- 1 If you use storage policies with storage providers, verify that an appropriate storage provider is registered. Entities that require storage providers include Virtual SAN and Virtual Volumes.
See [Chapter 24, “Using Storage Providers,”](#) on page 261.
- 2 Enable storage policies for a host or cluster. See [“Enable Virtual Machine Storage Policies,”](#) on page 228.
- 3 Apply storage tags to datastores. See [“Assign Tags to Datastores,”](#) on page 229.
- 4 Create storage policies by defining requirements for applications that run on a virtual machine. See [“Define a Storage Policy for a Virtual Machine,”](#) on page 230.
- 5 Apply the VM storage policy to a virtual machine. You can apply the storage policy when deploying the virtual machine or configuring its virtual disks. See [“Assign Storage Policies to Virtual Machines,”](#) on page 234.
- 6 Change the storage policy for virtual machine home files or virtual disks. See [“Change Storage Policy Assignment for Virtual Machine Files and Disks,”](#) on page 235.
- 7 Verify that virtual machines and virtual disks use datastores that are compliant with the assigned storage policy. See [“Check Compliance for a VM Storage Policy,”](#) on page 237.

Creating and Managing VM Storage Policies

To create and manage storage policies for your virtual machines, you can generally use the VM Storage Policies interface of the vSphere Web Client.

If you use storage policies with Virtual SAN or Virtual Volumes, see the following documentation for more information:

- [Administering VMware Virtual SAN](#)
- [Chapter 19, “Working with Virtual Volumes,”](#) on page 211


Enable Virtual Machine Storage Policies

To be able to use VM storage policies, you must enable them for a host or a cluster.

Prerequisites

Verify that a single host or all the hosts in a cluster where you want to enable the policies include the Storage Policies license.

Procedure

- 1 From the vSphere Web Client Home, click **Policies and Profiles > VM Storage Policies**.
- 2 Click the **Enable VM Storage Policies** icon ().
- 3 Select the vCenter Server instance.
All clusters and hosts that are available for the selected server appear. You can see the server licensing status and whether storage policies are enabled.
- 4 To enable storage policies, select a host or a cluster whose status is Disabled or Unknown and click **Enable**.

The status of the host or cluster changes to Enabled.

What to do next

Define VM storage policies.

Assign Tags to Datastores

When your datastore is not represented by a storage provider and does not display its capabilities and data services in the VM Storage Policies interface, use tags to encode information about the datastore. You can reference these tags when defining a storage policy for a virtual machine.

You can apply a new tag that contains storage information to a datastore. For information about tags, their categories, and how to manage tags, see the *vCenter Server and Host Management* documentation.

Procedure

- 1 Browse to a datastore in the vSphere Web Client navigator.
- 2 Click the **Manage** tab and click **Tags**.
- 3 Click the **New Tag** icon.
- 4 In the **vCenter Server** drop-down menu, select the vCenter Server instance on which to create the tag.
- 5 Type a name and a description for the tag.

For example, you can specify a broad storage-level definition, such as Gold Storage, or indicate a property that is not communicated through a storage provider, such as geographical location or administrative group.

Tag Property	Example
Name	Fault Tolerance
Description	Storage that has a capacity over 2 TB and is fault-tolerant

- 6 In the **Category** drop-down menu, select an existing category or create a category.
- 7 (Optional) Create a category:
 - a Select **New Category**.
 - b Specify the category options.

Category Property	Example
Category Name	Storage Category
Description	Category for tags related to storage
Cardinality	Many tags per object
Associable Object Types	Datastore and Datastore Cluster

- 8 Click **OK**.

The new tag is assigned to the datastore and appears on the datastore's **Summary** tab in the Tags pane.

What to do next

You can reference the tag when adding tag-based rules to the storage policy. See [“Add or Edit Tag-Based Rules,”](#) on page 231. The datastore appears in a list of compatible storage resources for virtual machines that use the policy.

Define a Storage Policy for a Virtual Machine

When you define storage policies for virtual machines, you specify storage requirements for applications that will run on the virtual machines.

A storage policy can be based on data services that are advertised by a storage entity or on datastore tags.

Prerequisites

- If you use VM storage policies with storage providers, verify that an appropriate storage provider is registered. See [Chapter 24, “Using Storage Providers,”](#) on page 261.
- Required privileges: **VM storage policies.Update** and **VM storage policies.View**.

Procedure

- 1 [Start VM Storage Policy Creation Process](#) on page 230
To define a virtual machine storage policy, use the Create New VM Storage Policy wizard.
- 2 [Create Storage-Specific Rules for a VM Storage Policy](#) on page 230
Datastore-specific rules are based on data services that storage entities such as Virtual SAN and Virtual Volumes advertise. To the virtual machine that uses this policy, the datastore guarantees that it can satisfy the storage requirements of the virtual machine. the datastore also guarantees that it can provide a specific set of characteristics for capacity, performance, availability, redundancy, and so on.
- 3 [Add or Edit Tag-Based Rules](#) on page 231
When you define or edit a storage policy for virtual machines, you can create or modify a rule that references tags that you used for particular datastores. The datastores become compatible with that type of storage policy.
- 4 [Finish VM Storage Policy Creation](#) on page 232
You can review the list of datastores that are compatible with the VM storage policy and change any storage policy settings.

What to do next

You can apply this storage policy to virtual machines. If you use object-based storage, such as Virtual SAN and Virtual Volumes, you can designate this storage policy as default.

Start VM Storage Policy Creation Process

To define a virtual machine storage policy, use the Create New VM Storage Policy wizard.

Procedure

- 1 From the vSphere Web Client Home, click **Policies and Profiles > VM Storage Policies**.
- 2 Click the **Create a New VM Storage Policy** icon.
- 3 Select the vCenter Server instance.
- 4 Type a name and a description for the storage policy.

Create Storage-Specific Rules for a VM Storage Policy

Datastore-specific rules are based on data services that storage entities such as Virtual SAN and Virtual Volumes advertise. To the virtual machine that uses this policy, the datastore guarantees that it can satisfy the storage requirements of the virtual machine. the datastore also guarantees that it can provide a specific set of characteristics for capacity, performance, availability, redundancy, and so on.

One datastore-specific rule set can include rules from only a single storage entity.

Prerequisites

If your environment includes storage entities such as Virtual SAN or Virtual Volumes, review these functionalities. For information, see the *Administering VMware Virtual SAN* documentation and [Chapter 19, “Working with Virtual Volumes,”](#) on page 211.

Procedure

- 1 On the Rule Set page, select a storage provider, for example, Virtual SAN or Virtual Volumes, from the **Rules based on data services** drop-down menu.

The page expands to show data services provided by the storage resource.

- 2 Select a data service to include and specify its value.

Verify that the values you provide are within the range of values that the data services profile of the storage resource advertises.

Based on your input, the storage consumption mechanism calculates the amount of space that is required for a virtual disk that will reside on this storage entity.

- 3 (Optional) Add tag-based rules.
- 4 Click **Next**.

Add or Edit Tag-Based Rules

When you define or edit a storage policy for virtual machines, you can create or modify a rule that references tags that you used for particular datastores. The datastores become compatible with that type of storage policy.

You can add tag-based rules to a rule set that includes storage-specific rules, or create a separate rule set with only tag-based rules. When you use tags in the policies, follow these guidelines:

- If the rule set contains other storage-specific rules, the datastore with the assigned tag must satisfy all of the rules in the rule set.
- If you add several tags from the same category within the same rule, the tags are treated as alternative constraints. Either of the tags can be satisfied.
- If you add the tags in separate rules within the same rule set, all tags must be satisfied.

Prerequisites

Create storage tags and apply them to datastores. See [“Assign Tags to Datastores,”](#) on page 229.

Procedure

- 1 On the Rule Set page, add or edit a tag-based rule:
 - To add a rule, click the **Add tag-based rule** button.
 - To modify an existing rule, select the rule and click the **Modify rule** icon.
- 2 Specify a category.
- 3 Make a tag selection or edit an existing selection.

Datastores that use the tags that you selected are compatible with the storage policy.

Finish VM Storage Policy Creation

You can review the list of datastores that are compatible with the VM storage policy and change any storage policy settings.

Procedure

- 1 On the Storage Compatibility page, review the list of datastores that match this policy and click **Next**.
To be eligible, the datastore must satisfy at least one rule set and all rules within this set.
- 2 Review the storage policy settings and make changes by clicking **Back** to go back to the relevant page.
- 3 Click **Finish**.

The VM storage policy appears in the list.

Delete a Virtual Machine Storage Policy

You can delete a storage policy if you are not using it for any virtual machine or virtual disk.

Procedure

- 1 From the vSphere Web Client Home, click **Policies and Profiles > VM Storage Policies**.
- 2 In the VM Storage Policies interface, select a policy to delete and click the **Delete a VM Storage Policy** icon (✖).
- 3 Click **Yes**.

The policy is removed from the inventory.

Edit or Clone a VM Storage Policy

If storage requirements for virtual machines and virtual disks change, you can modify the existing storage policy. You can also create a copy of the existing VM storage policy by cloning it. While cloning, you can optionally select to customize the original storage policy.

Prerequisites

Required privilege: **StorageProfile.View**

Procedure

- 1 From the vSphere Web Client Home, click **Policies and Profiles > VM Storage Policies**.
- 2 Select a storage policy, and click one of the following icons:
 - **Edit a VM storage policy**
 - **Clone a VM storage policy**
- 3 (Optional) Modify the policy and click **OK**.

- 4 If editing the storage policy that is used by a virtual machine, reapply the policy to the virtual machine.

Option	Description
Manually later	If you select this option, the compliance status for all virtual disks and virtual machine home objects associated with the storage policy changes to Out of Date. To update configuration and compliance, manually reapply the storage policy to all associated entities. See “Reapply Virtual Machine Storage Policy,” on page 238.
Now	Update virtual machine and compliance status immediately after editing the storage policy.

Storage Policies and Virtual Machines

After you define a VM storage policy, you can apply it to a virtual machine. You apply the storage policy when provisioning the virtual machine or configuring its virtual disks. Depending on its type and configuration, the policy might serve different purposes. It can select the most appropriate datastore for the virtual machine and enforce the required level of service, or it can enable specific data services for the virtual machine and its disks.

If you do not specify the storage policy, the system uses a default storage policy that is associated with the datastore. If your storage requirements for the applications on the virtual machine change, you can modify the storage policy that was originally applied to the virtual machine.

Default Storage Policies

When you provision a virtual machine on an object-based datastore, such as Virtual SAN or Virtual Volumes, you must assign to the virtual machine an appropriate virtual machine storage policy that is compatible with the datastore. This assignment guarantees the optimum placement for virtual machine objects within the object-based storage. If you do not explicitly assign a storage policy to the virtual machine, the system uses a default storage policy that is associated with the datastore. The default policy is also used when the policy that you assign does not include rules specific to Virtual Volumes or Virtual SAN.

The default storage policies for Virtual SAN and Virtual Volumes can be provided by VMware and user-defined. VMFS and NFS datastores do not have default policies.

Default Policies Provided by VMware

VMware provides default storage policies for Virtual SAN and virtual datastores.

Virtual SAN Default Storage Policy

The default storage policy that VMware provides is applied to all virtual machine objects that are provisioned on a Virtual SAN datastore when you do not select any other Virtual SAN policy.

The policy that VMware provides has the following characteristics:

- You cannot delete the policy.
- The policy is editable. To edit the policy, you must have the storage policy privileges that include the view and update privileges.
- When editing the policy, you cannot change the name of the policy or the Virtual SAN storage provider specification. All other parameters including rules are editable.
- You can clone the default policy and use it as a template to create a storage policy.
- The Virtual SAN default policy is compatible only with Virtual SAN datastores.
- You can create a VM storage policy for Virtual SAN and designate it as the default.

Virtual Volumes Default Storage Policy

For Virtual Volumes, VMware provides a default storage policy that contains no rules or storage requirements. As with Virtual SAN, this policy is applied to virtual machine objects when you do not specify another policy for the virtual machine that you place on the virtual datastore. With the No Requirements policy, storage arrays can determine the optimum placement for the VM objects.

The default Virtual Volumes policy that VMware provides has the following characteristics:

- You cannot delete, edit, or clone this policy.
- The Virtual Volumes default policy is compatible only with virtual datastores.
- You can create a VM storage policy for Virtual Volumes and designate it as the default.

User-Defined Default Policies for Virtual Machine Storage

You can create a VM storage policy that is compatible with Virtual SAN or Virtual Volumes, and designate this policy as the default for Virtual SAN and virtual datastores. The user-defined default policy replaces the default storage policy that VMware provided.

Each Virtual SAN and virtual datastore can have only one default policy at a time. However, you can create a single storage policy that matches multiple Virtual SAN and Virtual Volumes datastores and designate it as the default policy for all datastores.

When the VM storage policy becomes the default policy for a datastore, you cannot delete the policy unless you disassociate it from the datastore.

Change the Default Storage Policy for a Datastore

For Virtual Volumes and Virtual SAN datastores, VMware provides storage policies that are used as the default during virtual machine provisioning. You can change the default storage policy for a selected Virtual Volumes or Virtual SAN datastore.

Prerequisites

Create a storage policy that is compatible with Virtual Volumes or Virtual SAN. You can create a policy that matches both types of storage.

Procedure

- 1 In the vSphere Web Client navigator, select **vCenter Inventory Lists > Datastores**
- 2 Click the datastore.
- 3 Click the **Manage** tab and click **Settings**.
- 4 Click **General**, and click **Edit** in the Default Storage Policy pane.
- 5 From the list of available storage policies, select a policy to designate as the default and click **OK**.

The selected storage policy becomes the default policy for the datastore. vSphere assigns this policy to any virtual machine objects that you provision on the datastore when no other policy is selected.

Assign Storage Policies to Virtual Machines

You can assign a VM storage policy in an initial deployment of a virtual machine or when you perform other virtual machine operations, such as cloning or migrating.

This topic describes how to assign the VM storage policy when you create a virtual machine. For information about other deployment methods that include cloning, deployment from a template, and so on, see the *vSphere Virtual Machine Administration* documentation.

You can apply the same storage policy to the virtual machine configuration file and all its virtual disks. If storage requirements for your virtual disks and the configuration file are different, you can associate different storage policies with the VM configuration file and the selected virtual disks.

Procedure

- 1 In the vSphere Web Client, start the virtual machine provisioning process and follow the appropriate steps.
- 2 Assign the same storage policy to all virtual machine files and disks.
 - a On the Select Storage page, select a storage policy from the **VM Storage Policy** drop-down menu.

Based on its configuration, the storage policy separates all datastores into compatible and incompatible sets. If the policy references data services offered by a specific storage entity, for example, Virtual Volumes, the compatible list includes datastores that represent only that type of storage.
 - b Select an appropriate datastore from the list of compatible datastores and click **Next**.

The datastore becomes the destination storage resource for the virtual machine configuration file and all virtual disks.

- 3 Change the VM storage policy for the virtual disk.

Use this option if requirements for storage placement are different for virtual disks. You can also use this option if you must enable software data services, such as caching and replication, for your virtual disks.

 - a On the Customize Hardware page, expand the **New hard disk** pane.
 - b From the **VM storage policy** drop-down menu, select the storage policy to assign to the virtual disk.
 - c (Optional) Change the storage location of the virtual disk.

Use this option to store the virtual disk on a datastore other than the datastore where the VM configuration file resides.

- 4 Complete the virtual machine provisioning process.

After you create the virtual machine, the **Summary** tab displays the assigned storage policies and their compliance status.

What to do next

If storage placement requirements for the configuration file or the virtual disks change, you can later modify the virtual policy assignment.

Change Storage Policy Assignment for Virtual Machine Files and Disks

If your storage requirements for the applications on the virtual machine change, you can edit the storage policy that was originally applied to the virtual machine.

You can edit the storage policy for a powered-off or powered-on virtual machine.

When changing the VM storage policy assignment, you can apply the same storage policy to the virtual machine configuration file and all its virtual disks. If storage requirements for your virtual disks and the configuration file are different, you can associate different storage policies with the VM configuration file and the selected virtual disks.

Procedure

- 1 In the vSphere Web Client, browse to the virtual machine.

- 2 Click the **Manage** tab and click **Policies**.
- 3 Click **Storage**.
- 4 Click **Edit VM Storage Policies**.
- 5 Specify the VM storage policy for your virtual machine.

Option	Description
Apply the same storage policy to all virtual machine objects.	<ol style="list-style-type: none"> a Select the policy from the VM storage policy drop-down menu. b Click Apply to all.
Apply different storage policies to the VM home object and virtual disks.	<ol style="list-style-type: none"> a Select the object. b Select the policy from the VM storage policy drop-down menu for the object.

- 6 Click **OK**.

The storage policy is assigned to the virtual machine and its disks.

Monitor Storage Compliance for Virtual Machines

When you associate a policy with virtual machine objects and select the datastores on which virtual machines and virtual disks run, you can check whether virtual machines and virtual disks use datastores that are compliant with the policy.

If you check the compliance of a virtual machine whose host or cluster has storage policies disabled, the result of the check is Noncompliant because the feature is disabled.

Prerequisites

To perform a compliance check for a storage policy, associate the policy with at least one virtual machine or virtual disk.

Procedure

- 1 From the vSphere Web Client Home, click **Policies and Profiles > VM Storage Policies**.
- 2 Double-click a storage policy.
- 3 Click the **Monitor** tab, and click **VMs and Virtual Disks**.
- 4 Click **Trigger VM Storage Policy Compliance Check**.

The Compliance Status column shows the storage compliance status for virtual machines and their policies.

Compliance Status	Description
Compliant	The datastore that the virtual machine or virtual disk uses has the storage capabilities that are required by the policy.
Noncompliant	The datastore supports specified storage requirements, but cannot currently satisfy the virtual machine storage policy. For example, the status might become Not Compliant when physical resources that back up the datastore are unavailable or exhausted. You can bring the datastore into compliance by making changes in the physical configuration, for example, by adding hosts or disks to the cluster. If additional resources satisfy the virtual machine storage policy, the status changes to Compliant.
Out of Date	The status indicates that the policy has been edited, but the new requirements have not been communicated to the datastore where the virtual machine objects reside. To communicate the changes, reapply the policy to the objects that are out of date.
Not Applicable	This storage policy references datastore capabilities that are not supported by the datastore where the virtual machine resides.

What to do next

When you cannot bring the noncompliant datastore into compliance, migrate the files or virtual disks to a compatible datastore. See [“Find Compatible Storage Resource for Noncompliant Virtual Machine,”](#) on page 237.

If the status is Out of Date, reapply the policy to the objects. See [“Reapply Virtual Machine Storage Policy,”](#) on page 238.

Check Compliance for a VM Storage Policy

You can check whether a virtual machine uses a datastore that is compatible with the storage requirements specified in the VM storage policy.

Prerequisites

Verify that the virtual machine has a storage policy that is associated with it.

Procedure

- 1 In the vSphere Web Client, browse to the virtual machine.
- 2 From the right-click menu, select **VM Policies > Check VM Storage Policy Compliance**.

The system verifies the compliance.

- 3 Click the **Summary** tab for the virtual machine.
- 4 View the compliance status in the VM Storage Policies pane.

Compliance Status	Description
Compliant	The datastore that the virtual machine or virtual disk uses has the storage capabilities that are required by the policy.
Noncompliant	The datastore supports specified storage requirements, but cannot currently satisfy the virtual machine storage policy. For example, the status might become Not Compliant when physical resources that back up the datastore are unavailable or exhausted. You can bring the datastore into compliance by making changes in the physical configuration, for example, by adding hosts or disks to the cluster. If additional resources satisfy the virtual machine storage policy, the status changes to Compliant.
Out of Date	The status indicates that the policy has been edited, but the new requirements have not been communicated to the datastore where the virtual machine objects reside. To communicate the changes, reapply the policy to the objects that are out of date.
Not Applicable	This storage policy references datastore capabilities that are not supported by the datastore where the virtual machine resides.

What to do next

When you cannot bring the noncompliant datastore into compliance, migrate the files or virtual disks to a compatible datastore. See [“Find Compatible Storage Resource for Noncompliant Virtual Machine,”](#) on page 237.

If the status is Out of Date, reapply the policy to the objects. See [“Reapply Virtual Machine Storage Policy,”](#) on page 238.

Find Compatible Storage Resource for Noncompliant Virtual Machine

Determine which datastore is compatible with the storage policy that is associated with your virtual machine.

Occasionally, a storage policy that is assigned to a virtual machine can be in the noncompliant status. This status indicates that the virtual machine or its disks use datastores that are incompatible with the policy. You can migrate the virtual machine files and virtual disks to compatible datastores.

Use this task to determine which datastores satisfy the requirements of the policy.

Prerequisites

Verify that the **VM Storage Policy Compliance** field on the virtual machine **Summary** tab displays the Not Compliant status.

Procedure

- 1 In the vSphere Web Client, browse to the virtual machine.
- 2 Click the **Summary** tab.
The VM Storage Policy Compliance panel on the VM Storage Policies pane shows the Not Compliant status.
- 3 Click the policy link in the **VM Storage Policies** panel.
- 4 Click the **Monitor** tab and click **VMs and Virtual Disks** to determine which virtual machine files are noncompliant.
- 5 Click **Storage Compatibility**.
The list of datastores that match the requirements of the policy appears.

What to do next

You can migrate the virtual machine or its disks to one of the datastores in the list.

Reapply Virtual Machine Storage Policy

After you edit a storage policy that is already associated with a virtual machine object, you must reapply the policy. By reapplying the policy, you communicate new storage requirements to the datastore where the virtual machine object resides.

Prerequisites

The compliance status for a virtual machine is Out of Date. The status indicates that the policy has been edited, but the new requirements have not been communicated to the datastore.

Procedure

- 1 In the vSphere Web Client, browse to the virtual machine.
- 2 Click the **Manage** tab and click **Policies**.
- 3 Verify that the compliance status is Out of Date.
- 4 Click the **Reapply VM Storage Policy** icon.
- 5 View the compliance status in the VM Storage Policies pane.

Compliance Status	Description
Compliant	The datastore that the virtual machine or virtual disk uses has the storage capabilities that the policy requires.
Noncompliant	The datastore that the virtual machine or virtual disk uses does not have the storage capabilities that the policy requires. You can migrate the virtual machine files and virtual disks to compliant datastores. When you cannot bring the noncompliant datastore into compliance, migrate the files or virtual disks to a compatible datastore. See “Find Compatible Storage Resource for Noncompliant Virtual Machine,” on page 237.
Not Applicable	This storage service level references datastore capabilities that are not supported by the datastore where the virtual machine resides.

VMkernel and Storage

The VMkernel is a high-performance operating system that runs directly on the ESXi host. The VMkernel manages most of the physical resources on the hardware, including memory, physical processors, storage, and networking controllers.

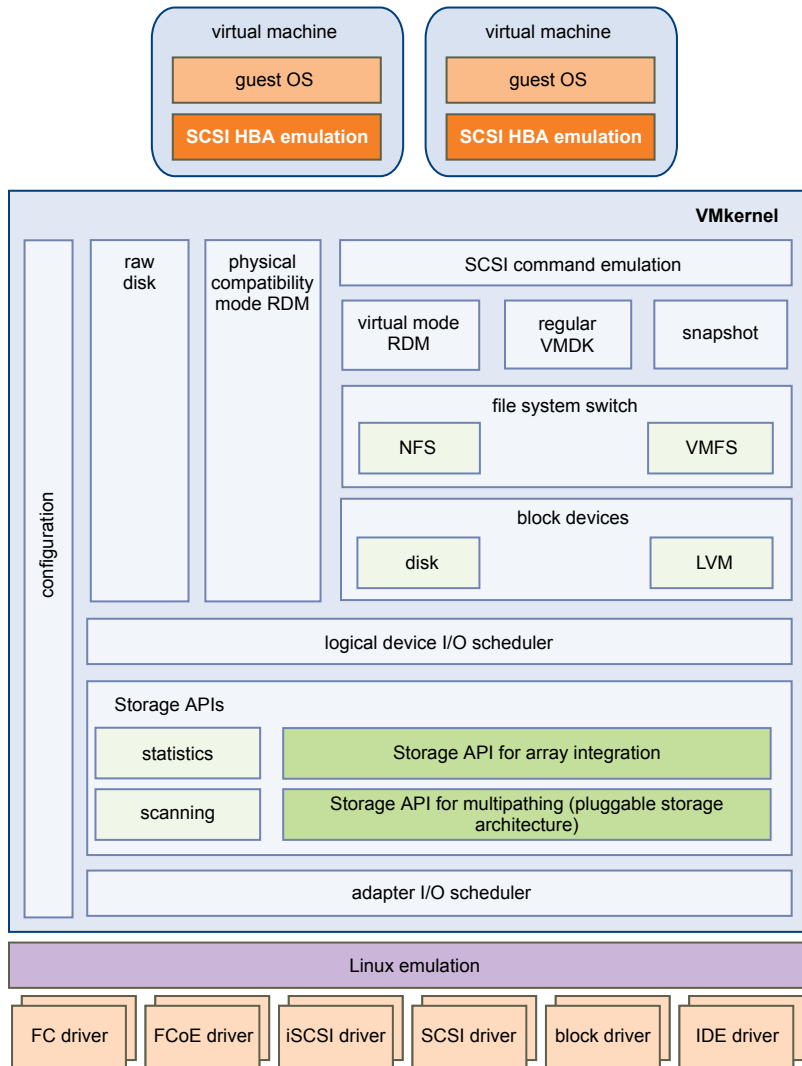
To manage storage, VMkernel has a storage subsystem that supports several Host Bus Adapters (HBAs) including parallel SCSI, SAS, Fibre Channel, FCoE, and iSCSI. These HBAs connect a wide variety of active-active, active-passive, and ALUA storage arrays that are certified for use with the VMkernel. See the *vSphere Compatibility Guide* for a list of the supported HBAs and storage arrays.

The primary file system that the VMkernel uses is the VMware Virtual Machine File System (VMFS). VMFS is a cluster file system designed and optimized to support large files such as virtual disks and swap files. The VMkernel also supports the storage of virtual disks on NFS file systems.

The storage I/O path provides virtual machines with access to storage devices through device emulation. This device emulation allows a virtual machine to access files on a VMFS or NFS file system as if they were SCSI devices. The VMkernel provides storage virtualization functions such as the scheduling of I/O requests from multiple virtual machines and multipathing.

In addition, VMkernel offers several Storage APIs that enable storage partners to integrate and optimize their products for vSphere.

The following graphic illustrates the basics of the VMkernel core, with special attention to the storage stack. Storage-related modules reside between the logical device I/O scheduler and the adapter I/O scheduler layers.

Figure 21-1. VMkernel and Storage

Storage APIs

Storage APIs is a family of APIs used by third-party hardware, software, and storage providers to develop components that enhance several vSphere features and solutions.

This publication describes the following sets of Storage APIs and explains how they contribute to your storage environment. For information about other APIs from this family, including Storage API - Data Protection and Storage API - Site Recovery Manager, see the VMware Web site.

- Storage APIs - Multipathing, also known as the Pluggable Storage Architecture (PSA). PSA is a collection of VMkernel APIs that allows storage partners to enable and certify their arrays asynchronous to ESXi release schedules, as well as deliver performance-enhancing, multipathing and load-balancing behaviors that are optimized for each array. For more information, see [“Managing Multiple Paths,”](#) on page 186.
- Storage APIs - Array Integration, formerly known as VAAI, include the following APIs:
 - Hardware Acceleration APIs. Allows arrays to integrate with vSphere to transparently offload certain storage operations to the array. This integration significantly reduces CPU overhead on the host. See [Chapter 22, “Storage Hardware Acceleration,”](#) on page 243.

- Array Thin Provisioning APIs. Help to monitor space use on thin-provisioned storage arrays to prevent out-of-space conditions, and to perform space reclamation. See [“Array Thin Provisioning and VMFS Datastores,”](#) on page 257.
- Storage APIs - Storage Awareness. These vCenter Server-based APIs enable storage arrays to inform the vCenter Server about their configurations, capabilities, and storage health and events. See [Chapter 24, “Using Storage Providers,”](#) on page 261.

Storage Hardware Acceleration

The hardware acceleration functionality enables the ESXi host to integrate with compliant storage arrays and offload specific virtual machine and storage management operations to storage hardware. With the storage hardware assistance, your host performs these operations faster and consumes less CPU, memory, and storage fabric bandwidth.

The hardware acceleration is supported by block storage devices, Fibre Channel and iSCSI, and NAS devices.

For additional details, see the VMware knowledge base article at <http://kb.vmware.com/kb/1021976>.

This chapter includes the following topics:

- [“Hardware Acceleration Benefits,”](#) on page 243
- [“Hardware Acceleration Requirements,”](#) on page 244
- [“Hardware Acceleration Support Status,”](#) on page 244
- [“Hardware Acceleration for Block Storage Devices,”](#) on page 244
- [“Hardware Acceleration on NAS Devices,”](#) on page 249
- [“Hardware Acceleration Considerations,”](#) on page 251

Hardware Acceleration Benefits

When the hardware acceleration functionality is supported, the host can get hardware assistance and perform several tasks faster and more efficiently.

The host can get assistance with the following activities:

- Migrating virtual machines with Storage vMotion
- Deploying virtual machines from templates
- Cloning virtual machines or templates
- VMFS clustered locking and metadata operations for virtual machine files
- Provisioning thick virtual disks
- Creating fault-tolerant virtual machines
- Creating and cloning thick disks on NFS datastores

Hardware Acceleration Requirements

The hardware acceleration functionality works only if you use an appropriate host and storage array combination.

Table 22-1. Hardware Acceleration Storage Requirements

ESXi	Block Storage Devices	NAS Devices
ESXi version 6.0	Support T10 SCSI standard, or block storage plug-ins for array integration (VAAI)	Support NAS plug-ins for array integration NOTE NFS 4.1 does not support hardware acceleration.

NOTE If your SAN or NAS storage fabric uses an intermediate appliance in front of a storage system that supports hardware acceleration, the intermediate appliance must also support hardware acceleration and be properly certified. The intermediate appliance might be a storage virtualization appliance, I/O acceleration appliance, encryption appliance, and so on.

Hardware Acceleration Support Status

For each storage device and datastore, the vSphere Web Client display the hardware acceleration support status.

The status values are Unknown, Supported, and Not Supported. The initial value is Unknown.

For block devices, the status changes to Supported after the host successfully performs the offload operation. If the offload operation fails, the status changes to Not Supported. The status remains Unknown if the device provides partial hardware acceleration support.

With NAS, the status becomes Supported when the storage can perform at least one hardware offload operation.

When storage devices do not support or provide partial support for the host operations, your host reverts to its native methods to perform unsupported operations.

Hardware Acceleration for Block Storage Devices

With hardware acceleration, your host can integrate with block storage devices, Fibre Channel or iSCSI, and use certain storage array operations.

ESXi hardware acceleration supports the following array operations:

- Full copy, also called clone blocks or copy offload. Enables the storage arrays to make full copies of data within the array without having the host read and write the data. This operation reduces the time and network load when cloning virtual machines, provisioning from a template, or migrating with vMotion.
- Block zeroing, also called write same. Enables storage arrays to zero out a large number of blocks to provide newly allocated storage, free of previously written data. This operation reduces the time and network load when creating virtual machines and formatting virtual disks.
- Hardware assisted locking, also called atomic test and set (ATS). Supports discrete virtual machine locking without use of SCSI reservations. This operation allows disk locking per sector, instead of the entire LUN as with SCSI reservations.

Check with your vendor for the hardware acceleration support. Certain storage arrays require that you activate the support on the storage side.

On your host, the hardware acceleration is enabled by default. If your storage does not support the hardware acceleration, you can disable it.

In addition to hardware acceleration support, ESXi includes support for array thin provisioning. For information, see [“Array Thin Provisioning and VMFS Datastores,”](#) on page 257.

Disable Hardware Acceleration for Block Storage Devices

On your host, the hardware acceleration for block storage devices is enabled by default. You can use the vSphere Web Client advanced settings to disable the hardware acceleration operations.

As with any advanced settings, before you disable the hardware acceleration, consult with the VMware support team.

Procedure

- 1 Browse to the host in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Settings**.
- 3 Under System, click **Advanced System Settings**.
- 4 Change the value for any of the options to 0 (disabled):
 - VMFS3.HardwareAcceleratedLocking
 - DataMover.HardwareAcceleratedMove
 - DataMover.HardwareAcceleratedInit

Managing Hardware Acceleration on Block Storage Devices

To integrate with the block storage arrays and to benefit from the array hardware operations, vSphere uses the ESXi extensions referred to as Storage APIs - Array Integration, formerly called VAAI.

In the vSphere 5.x and later releases, these extensions are implemented as the T10 SCSI based commands. As a result, with the devices that support the T10 SCSI standard, your ESXi host can communicate directly and does not require the VAAI plug-ins.

If the device does not support T10 SCSI or provides partial support, ESXi reverts to using the VAAI plug-ins, installed on your host, or uses a combination of the T10 SCSI commands and plug-ins. The VAAI plug-ins are vendor-specific and can be either VMware or partner developed. To manage the VAAI capable device, your host attaches the VAAI filter and vendor-specific VAAI plug-in to the device.

For information about whether your storage requires VAAI plug-ins or supports hardware acceleration through T10 SCSI commands, see the *VMware Compatibility Guide* or check with your storage vendor.

You can use several `esxcli` commands to query storage devices for the hardware acceleration support information. For the devices that require the VAAI plug-ins, the claim rule commands are also available. For information about `esxcli` commands, see *Getting Started with vSphere Command-Line Interfaces*.

Display Hardware Acceleration Plug-Ins and Filter

To communicate with the devices that do not support the T10 SCSI standard, your host uses a combination of a single VAAI filter and a vendor-specific VAAI plug-in. Use the `esxcli` command to view the hardware acceleration filter and plug-ins currently loaded into your system.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ Run the `esxcli --server=server_name storage core plugin list --plugin-class=value` command.

For *value*, enter one of the following options:

- Type VAAI to display plug-ins.

The output of this command is similar to the following example:

```
#esxcli --server=server_name storage core plugin list --plugin-class=VAAI
Plugin name      Plugin class
VMW_VAAIP_EQL   VAAI
VMW_VAAIP_NETAPP VAAI
VMW_VAAIP_CX    VAAI
```

- Type Filter to display the Filter.

The output of this command is similar to the following example:

```
esxcli --server=server_name storage core plugin list --plugin-class=Filter
Plugin name  Plugin class
VAAI_FILTER Filter
```

Verify Hardware Acceleration Support Status

Use the `esxcli` command to verify the hardware acceleration support status of a particular storage device.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ Run the `esxcli --server=server_name storage core device list -d=device_ID` command.

The output shows the hardware acceleration, or VAAI, status that can be unknown, supported, or unsupported.

```
# esxcli --server=server_name storage core device list -d naa.XXXXXXXXXXXXX4c
naa.XXXXXXXXXXXXX4c
Display Name: XXXX Fibre Channel Disk(naa.XXXXXXXXXXXXX4c)
Size: 20480
Device Type: Direct-Access
Multipath Plugin: NMP
XXXXXXXXXXXXXXXXXX
Attached Filters: VAAI_FILTER
VAAI Status: supported
XXXXXXXXXXXXXXXXXX
```

Verify Hardware Acceleration Support Details

Use the `esxcli` command to query the block storage device about the hardware acceleration support the device provides.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ Run the `esxcli --server=server_name storage core device vaaip status get -d=device_ID` command.

If the device is managed by a VAAI plug-in, the output shows the name of the plug-in attached to the device. The output also shows the support status for each T10 SCSI based primitive, if available. Output appears in the following example:

```
# esxcli --server=server_name storage core device vaaip status get -d naa.XXXXXXXXXXX4c
naa.XXXXXXXXXXX4c
VAAI Plugin Name: VMW_VAAIP_SYMM
ATS Status: supported
Clone Status: supported
Zero Status: supported
Delete Status: unsupported
```

List Hardware Acceleration Claim Rules

Each block storage device managed by a VAAI plug-in needs two claim rules, one that specifies the hardware acceleration filter and another that specifies the hardware acceleration plug-in for the device. You can use the `esxcli` commands to list the hardware acceleration filter and plug-in claim rules.

Procedure

- 1 To list the filter claim rules, run the `esxcli --server=server_name storage core claimrule list --claimrule-class=Filter` command.

In this example, the filter claim rules specify devices that should be claimed by the `VAAI_FILTER` filter.

```
# esxcli --server=server_name storage core claimrule list --claimrule-class=Filter
Rule Class Rule Class Type Plugin Matches
Filter 65430 runtime vendor VAAI_FILTER vendor=EMC model=SYMMETRIX
Filter 65430 file vendor VAAI_FILTER vendor=EMC model=SYMMETRIX
Filter 65431 runtime vendor VAAI_FILTER vendor=DGC model=*
Filter 65431 file vendor VAAI_FILTER vendor=DGC model=*
```

- 2 To list the VAAI plug-in claim rules, run the `esxcli --server=server_name storage core claimrule list --claimrule-class=VAAI` command.

In this example, the VAAI claim rules specify devices that should be claimed by a particular VAAI plug-in.

```
esxcli --server=server_name storage core claimrule list --claimrule-class=VAAI
Rule Class Rule Class Type Plugin Matches
VAAI 65430 runtime vendor VMW_VAAIP_SYMM vendor=EMC model=SYMMETRIX
VAAI 65430 file vendor VMW_VAAIP_SYMM vendor=EMC model=SYMMETRIX
VAAI 65431 runtime vendor VMW_VAAIP_CX vendor=DGC model=*
VAAI 65431 file vendor VMW_VAAIP_CX vendor=DGC model=*
```

Add Hardware Acceleration Claim Rules

To configure hardware acceleration for a new array, you need to add two claim rules, one for the VAAI filter and another for the VAAI plug-in. For the new claim rules to be active, you first define the rules and then load them into your system.

This procedure is for those block storage devices that do not support T10 SCSI commands and instead use the VAAI plug-ins.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 Define a new claim rule for the VAAI filter by running the `esxcli --server=server_name storage core claimrule add --claimrule-class=Filter --plugin=VAAI_FILTER` command.
- 2 Define a new claim rule for the VAAI plug-in by running the `esxcli --server=server_name storage core claimrule add --claimrule-class=VAAI` command.
- 3 Load both claim rules by running the following commands:


```
esxcli --server=server_name storage core claimrule load --claimrule-class=Filter
esxcli --server=server_name storage core claimrule load --claimrule-class=VAAI
```
- 4 Run the VAAI filter claim rule by running the `esxcli --server=server_name storage core claimrule run --claimrule-class=Filter` command.

NOTE Only the Filter-class rules need to be run. When the VAAI filter claims a device, it automatically finds the proper VAAI plug-in to attach.

Example: Defining Hardware Acceleration Claim Rules

This example shows how to configure hardware acceleration for IBM arrays using the `VMW_VAAIP_T10` plug-in. Use the following sequence of commands. For information about the options that the command takes, see [“Add Multipathing Claim Rules,”](#) on page 195.

```
# esxcli --server=server_name storage core claimrule add --claimrule-class=Filter --
plugin=VAAI_FILTER --type=vendor --vendor=IBM --autoassign

# esxcli --server=server_name storage core claimrule add --claimrule-class=VAAI --
plugin=VMW_VAAIP_T10 --type=vendor --vendor=IBM --autoassign

# esxcli --server=server_name storage core claimrule load --claimrule-class=Filter

# esxcli --server=server_name storage core claimrule load --claimrule-class=VAAI

# esxcli --server=server_name storage core claimrule run --claimrule-class=Filter
```


Delete Hardware Acceleration Claim Rules

Use the `esxcli` command to delete existing hardware acceleration claim rules.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ Run the following commands:

```
esxcli --server=server_name storage core claimrule remove -r claimrule_ID --claimrule-
class=Filter
```

```
esxcli --server=server_name storage core claimrule remove -r claimrule_ID --claimrule-
class=VAAI
```

Hardware Acceleration on NAS Devices

Hardware acceleration allows ESXi hosts to integrate with NAS devices and use several hardware operations that NAS storage provides. Hardware acceleration uses VMware vSphere Storage APIs for Array Integration (VAAI) to enable communication between the hosts and storage devices.

The APIs define a set of storage primitives that enable the host to offload certain storage operations to the array. The following list shows the supported NAS operations:

- Full File Clone. Enables NAS device to clone virtual disk files. This operation is similar to the VMFS block cloning, except that NAS devices clone entire files instead of file segments.
- Reserve Space. Enables storage arrays to allocate space for a virtual disk file in thick format. Typically, when you create a virtual disk on an NFS datastore, the NAS server determines the allocation policy. The default allocation policy on most NAS servers is thin and does not guarantee backing storage to the file. However, the reserve space operation can instruct the NAS device to use vendor-specific mechanisms to reserve space for a virtual disk. As a result, you can create thick virtual disks on the NFS datastore.
- Native Snapshot Support. Allows creation of virtual machine snapshots to be offloaded to the array.
- Extended Statistics. Enables visibility to space usage on NAS devices and is useful for Thin Provisioning.

With NAS storage devices, the hardware acceleration integration is implemented through vendor-specific NAS plug-ins. These plug-ins are typically created by vendors and are distributed as VIB packages through a web page. No claim rules are required for the NAS plug-ins to function.

There are several tools available for installing and upgrading VIB packages. They include the `esxcli` commands and vSphere Update Manager. For more information, see the *vSphere Upgrade and Installing and Administering VMware vSphere Update Manager* documentation.

Install NAS Plug-In

Install vendor-distributed hardware acceleration NAS plug-ins on your host.

This topic provides an example for a VIB package installation using the `esxcli` command. For more details, see the *vSphere Upgrade* documentation.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 Place your host into the maintenance mode.

- 2 Set the host acceptance level:

```
esxcli --server=server_name software acceptance set --level=value
```

The command controls which VIB package is allowed on the host. The *value* can be one of the following:

- VMwareCertified
- VMwareAccepted
- PartnerSupported
- CommunitySupported

- 3 Install the VIB package:

```
esxcli --server=server_name software vib install -v|--viburl=URL
```

The *URL* specifies the URL to the VIB package to install. `http:`, `https:`, `ftp:`, and `file:` are supported.

- 4 Verify that the plug-in is installed:

```
esxcli --server=server_name software vib list
```

- 5 Reboot your host for the installation to take effect.

Uninstall NAS Plug-Ins

To uninstall a NAS plug-in, remove the VIB package from your host.

This topic discusses how to uninstall a VIB package using the `esxcli` command. For more details, see the *vSphere Upgrade* documentation.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 Uninstall the plug-in:

```
esxcli --server=server_name software vib remove -n|--vibName=name
```

The *name* is the name of the VIB package to remove.

- 2 Verify that the plug-in is removed:

```
esxcli --server=server_name software vib list
```

- 3 Reboot your host for the change to take effect.

Update NAS Plug-Ins

Upgrade hardware acceleration NAS plug-ins on your host when a storage vendor releases a new plug-in version.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

This topic discusses how to update a VIB package using the `esxcli` command. For more details, see the *vSphere Upgrade* documentation.

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- 1 Upgrade to a new plug-in version:

```
esxcli --server=server_name software vib update -v|--viburl=URL
```

The *URL* specifies the URL to the VIB package to update. `http:`, `https:`, `ftp:`, and `file:` are supported.

- 2 Verify that the correct version is installed:

```
esxcli --server=server_name software vib list
```

- 3 Reboot the host.

Verify Hardware Acceleration Status for NAS

In addition to the client, you can use the `esxcli` command to verify the hardware acceleration status of the NAS device.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ Run the `esxcli --server=server_name storage nfs list` command.

The Hardware Acceleration column in the output shows the status.

Hardware Acceleration Considerations

When you use the hardware acceleration functionality, certain considerations apply.

Several reasons might cause a hardware-accelerated operation to fail.

For any primitive that the array does not implement, the array returns an error. The error triggers the ESXi host to attempt the operation using its native methods.

The VMFS data mover does not leverage hardware offloads and instead uses software data movement when one of the following occurs:

- The source and destination VMFS datastores have different block sizes.

- The source file type is RDM and the destination file type is non-RDM (regular file).
- The source VMDK type is eagerzeroedthick and the destination VMDK type is thin.
- The source or destination VMDK is in sparse or hosted format.
- The source virtual machine has a snapshot.
- The logical address and transfer length in the requested operation are not aligned to the minimum alignment required by the storage device. All datastores created with the vSphere Web Client are aligned automatically.
- The VMFS has multiple LUNs or extents, and they are on different arrays.

Hardware cloning between arrays, even within the same VMFS datastore, does not work.

Storage Thick and Thin Provisioning

vSphere supports two models of storage provisioning, thick provisioning and thin provisioning.

Thick provisioning	It is a traditional model of storage provisioning. With thick provisioning, large amount of storage space is provided in advance in anticipation of future storage needs. However, the space might remain unused causing underutilization of storage capacity.
Thin provisioning	This method contrast with thick provisioning and helps you eliminate storage underutilization problems by allocating storage space in a flexible on-demand manner. With ESXi, you can use two models of thin provisioning, array-level and virtual disk-level.

This chapter includes the following topics:

- [“Storage Over-Subscription,”](#) on page 253
- [“Virtual Disk Thin Provisioning,”](#) on page 253
- [“Array Thin Provisioning and VMFS Datastores,”](#) on page 257

Storage Over-Subscription

Thin provisioning allows you to report more virtual storage space than there is real physical capacity. This discrepancy can lead to storage over-subscription, also called over-provisioning.

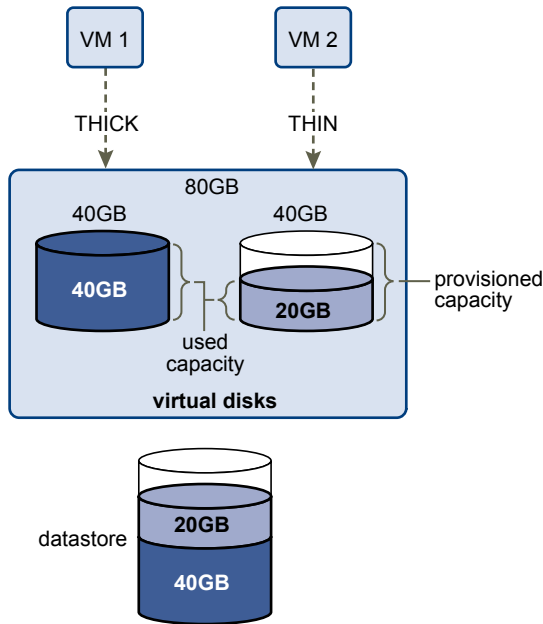
When you use thin provisioning, you should monitor actual storage usage to avoid conditions when you run out of physical storage space.

Virtual Disk Thin Provisioning

When you create a virtual machine, a certain amount of storage space on a datastore is provisioned to virtual disk files.

By default, ESXi offers a traditional storage provisioning method for virtual machines. With this method, you first estimate how much storage the virtual machine will need for its entire life cycle. You then provision a fixed amount of storage space to its virtual disk in advance, for example, 40GB, and have the entire provisioned space committed to the virtual disk. A virtual disk that immediately occupies the entire provisioned space is a thick disk.

ESXi supports thin provisioning for virtual disks. With the disk-level thin provisioning feature, you can create virtual disks in a thin format. For a thin virtual disk, ESXi provisions the entire space required for the disk’s current and future activities, for example 40GB. However, the thin disk uses only as much storage space as the disk needs for its initial operations. In this example, the thin-provisioned disk occupies only 20GB of storage. As the disk requires more space, it can grow into its entire 40GB provisioned space.

Figure 23-1. Thick and thin virtual disks

About Virtual Disk Provisioning Policies

When you perform certain virtual machine management operations, such as creating a virtual disk, cloning a virtual machine to a template, or migrating a virtual machine, you can specify a provisioning policy for the virtual disk file.

NFS datastores with Hardware Acceleration and VMFS datastores support the following disk provisioning policies. On NFS datastores that do not support Hardware Acceleration, only thin format is available.

You can use Storage vMotion or cross-host Storage vMotion to transform virtual disks from one format to another.

Thick Provision Lazy Zeroed

Creates a virtual disk in a default thick format. Space required for the virtual disk is allocated when the disk is created. Data remaining on the physical device is not erased during creation, but is zeroed out on demand at a later time on first write from the virtual machine. Virtual machines do not read stale data from the physical device.

Thick Provision Eager Zeroed

A type of thick virtual disk that supports clustering features such as Fault Tolerance. Space required for the virtual disk is allocated at creation time. In contrast to the thick provision lazy zeroed format, the data remaining on the physical device is zeroed out when the virtual disk is created. It might take longer to create virtual disks in this format than to create other types of disks.

Thin Provision

Use this format to save storage space. For the thin disk, you provision as much datastore space as the disk would require based on the value that you enter for the virtual disk size. However, the thin disk starts small and at first, uses only as much datastore space as the disk needs for its initial operations. If the thin disk needs more space later, it can grow to its maximum capacity and occupy the entire datastore space provisioned to it.

Thin provisioning is the fastest method to create a virtual disk because it creates a disk with just the header information. It does not allocate or zero out storage blocks. Storage blocks are allocated and zeroed out when they are first accessed.

NOTE If a virtual disk supports clustering solutions such as Fault Tolerance, do not make the disk thin.

You can manually inflate the thin disk, so that it occupies the entire provisioned space. If physical storage space is exhausted and the thin provisioned disk cannot grow, the virtual machine becomes unusable.

Create Thin Provisioned Virtual Disks

To save storage space, you can create a virtual disk in thin provisioned format. The thin provisioned virtual disk starts small and grows as more disk space is required. You can create thin disks only on the datastores that support disk-level thin provisioning.

This procedure assumes that you are creating a new virtual machine. For information, see the *vSphere Virtual Machine Administration* documentation.

Procedure

- 1 Right-click any inventory object that is a valid parent object of a virtual machine, such as a data center, folder, cluster, resource pool, or host, and select **New Virtual Machine**.
- 2 Select **Create a new virtual machine** and click **Next**.
- 3 Follow the steps required to create a virtual machine.
- 4 On the Customize Hardware page, click the **Virtual Hardware** tab.
- 5 Click the **New Hard Disk** triangle to expand the hard disk options.
- 6 (Optional) Adjust the default disk size.

With a thin virtual disk, the disk size value shows how much space is provisioned and guaranteed to the disk. At the beginning, the virtual disk might not use the entire provisioned space and the actual storage usage value could be less than the size of the virtual disk.

- 7 Select **Thin Provision** for Disk Provisioning.
- 8 Finish virtual machine creation.

You created a virtual machine with a disk in thin format.

What to do next

If you created a virtual disk in the thin format, you can later inflate it to its full size.

View Virtual Machine Storage Resources

You can view how datastore storage space is allocated for your virtual machines.

Storage Usage shows how much datastore space is occupied by virtual machine files, including configuration and log files, snapshots, virtual disks, and so on. When the virtual machine is running, the used storage space also includes swap files.

For virtual machines with thin disks, the actual storage usage value might be less than the size of the virtual disk.

Procedure

- 1 In the vSphere Web Client, browse to the virtual machine.
- 2 Double-click the virtual machine and click the **Summary** tab.
- 3 Review the storage usage information in the upper right area of the **Summary** tab.

Determine the Disk Format of a Virtual Machine

You can determine whether your virtual disk is in thick or thin format.

Procedure

- 1 In the vSphere Web Client, browse to the virtual machine.
- 2 Right-click the virtual machine and select **Edit Settings**.
- 3 Click the **Virtual Hardware** tab.
- 4 Click the **Hard Disk** triangle to expand the hard disk options.

The **Type** text box shows the format of your virtual disk.

What to do next


If your virtual disk is in the thin format, you can inflate it to its full size.

Inflate Thin Virtual Disks

If you created a virtual disk in the thin format, you can convert the thin disk to a virtual disk in thick provision format.

You use the datastore browser to inflate the virtual disk.

Procedure

- 1 Navigate to the folder of the virtual disk you want to inflate.
 - a In the vSphere Web Client, browse to the virtual machine.
 - b Double-click the virtual machine to display its information.
 - c Click the **Related Objects** tab and click **Datastores**.
The datastore that stores the virtual machine files is listed.
 - d Select the datastore and click the **Navigate to the datastore file browser** icon.
The datastore browser displays contents of the datastore.
- 2 Open the virtual machine folder and browse to the virtual disk file that you want to convert.
The file has the `.vmdk` extension and is marked with the virtual disk  icon.
- 3 Right-click the virtual disk file and select **Inflate**.

The inflated virtual disk occupies the entire datastore space originally provisioned to it.

Handling Datastore Over-Subscription

Because the provisioned space for thin disks can be greater than the committed space, a datastore over-subscription can occur, which results in the total provisioned space for the virtual machine disks on the datastore being greater than the actual capacity.

Over-subscription can be possible because usually not all virtual machines with thin disks need the entire provisioned datastore space simultaneously. However, if you want to avoid over-subscribing the datastore, you can set up an alarm that notifies you when the provisioned space reaches a certain threshold.

For information on setting alarms, see the *vCenter Server and Host Management* documentation.

If your virtual machines require more space, the datastore space is allocated on a first come first served basis. When the datastore runs out of space, you can add more physical storage and increase the datastore.

See [“Increase VMFS Datastore Capacity,”](#) on page 166.

Array Thin Provisioning and VMFS Datastores

You can use thin provisioned storage arrays with ESXi.

Traditional LUNs that arrays present to the ESXi host, are thick-provisioned. The entire physical space needed to back each LUN is allocated in advance.

ESXi also supports thin-provisioned LUNs. When a LUN is thin-provisioned, the storage array reports the LUN's logical size, which might be larger than the real physical capacity backing that LUN.

A VMFS datastore that you deploy on the thin-provisioned LUN can detect only the logical size of the LUN. For example, if the array reports 2TB of storage while in reality the array provides only 1TB, the datastore considers 2TB to be the LUN's size. As the datastore grows, it cannot determine whether the actual amount of physical space is still sufficient for its needs.

However, when you use the Storage APIs - Array Integration, the host can integrate with physical storage and become aware of underlying thin-provisioned LUNs and their space usage.

Using thin provision integration, your host can perform these tasks:

- Monitor the use of space on thin-provisioned LUNs to avoid running out of physical space. As your datastore grows or if you use Storage vMotion to migrate virtual machines to a thin-provisioned LUN, the host communicates with the LUN and warns you about breaches in physical space and about out-of-space conditions.
- Inform the array about the datastore space that is freed when files are deleted or removed from the datastore by Storage vMotion. The array can then reclaim the freed blocks of space.

NOTE ESXi does not support enabling and disabling of thin provisioning on a storage device.

Requirements

To use the thin provision reporting feature, your host and storage array must meet the following requirements:

- ESXi version 5.0 or later.
- Storage array has appropriate firmware that supports T10-based Storage APIs - Array Integration (Thin Provisioning). For information, contact your storage provider and check the HCL.

Space Usage Monitoring

The thin provision integration functionality helps you to monitor the space usage on thin-provisioned LUNs and to avoid running out of space.

The following sample flow demonstrates how the ESXi host and the storage array interact to generate breach of space and out-of-space warnings for a datastore with underlying thin-provisioned LUN. The same mechanism applies when you use Storage vMotion to migrate virtual machines to the thin-provisioned LUN.

- 1 Using storage-specific tools, your storage administrator provisions a thin LUN and sets a soft threshold limit that, when reached, triggers an alert. This step is vendor-specific.
- 2 Using the vSphere Web Client, you create a VMFS datastore on the thin-provisioned LUN. The datastore spans the entire logical size that the LUN reports.
- 3 As the space used by the datastore increases and reaches the specified soft threshold, the following actions take place:
 - a The storage array reports the breach to your host.
 - b Your host triggers a warning alarm for the datastore.

You can contact the storage administrator to request more physical space or use Storage vMotion to evacuate your virtual machines before the LUN runs out of capacity.

- 4 If no space is left to allocate to the thin-provisioned LUN, the following actions take place:
 - a The storage array reports out-of-space condition to your host.



CAUTION In certain cases, when a LUN becomes full, it might go offline or get unmapped from the host.

- b The host pauses virtual machines and generates an out-of-space alarm.

You can resolve the permanent out-of-space condition by requesting more physical space from the storage administrator.

Identify Thin-Provisioned Storage Devices

Use the `esxcli` command to verify whether a particular storage device is thin-provisioned.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ Run the `esxcli --server=server_name storage core device list -d=device_ID` command.

The following thin provisioning status indicates that the storage device is thin-provisioned.

```
# esxcli --server=server_name storage core device list -d naa.XXXXXXXXXXX4c
naa.XXXXXXXXXXX4c
Display Name: XXXX Fibre Channel Disk(naa.XXXXXXXXXXX4c)
Size: 20480
Device Type: Direct-Access
Multipath Plugin: NMP
-----
```

```
Thin Provisioning Status: yes
Attached Filters: VAAI_FILTER
VAAI Status: supported
-----
```

An unknown status indicates that a storage device is thick.

NOTE Some storage systems present all devices as thin-provisioned no matter whether the devices are thin or thick. Their thin provisioning status is always yes. For details, check with your storage vendor.

Reclaim Accumulated Storage Space

When VMFS datastores reside on thin-provisioned LUNs, you can use the `esxcli` command to reclaim unused storage blocks.

In the procedure, `--server=server_name` specifies the target server. The specified target server prompts you for a user name and password. Other connection options, such as a configuration file or session file, are supported. For a list of connection options, see *Getting Started with vSphere Command-Line Interfaces*.

Prerequisites

Install vCLI or deploy the vSphere Management Assistant (vMA) virtual machine. See *Getting Started with vSphere Command-Line Interfaces*. For troubleshooting, run `esxcli` commands in the ESXi Shell.

Procedure

- ◆ Reclaim unused storage blocks on the VMFS5 datastore for the thin-provisioned device by running the following command:

```
esxcli --server=server_name storage vmfs unmap --volume-label=volume_label|--volume-uid=volume_uuid --reclaim-unit=number
```

The command takes these options:

Option	Description
<code>-l --volume-label=volume_label</code>	The label of the VMFS volume to unmap. This is a mandatory argument. If you specify this argument, do not use <code>-u --volume-uid=volume_uuid</code> .
<code>-u --volume-uid=volume_uuid</code>	The UUID of the VMFS volume to unmap. This is a mandatory argument. If you specify this argument, do not use <code>-l --volume-label=volume_label</code> .
<code>-n --reclaim-unit=number</code>	Number of VMFS blocks to unmap per iteration. This is an optional argument. If it is not specified, the command uses the default value of 200.

What to do next

IMPORTANT For additional details, see the VMware knowledge base article at <http://kb.vmware.com/kb/2014849>.

Using Storage Providers

A storage provider is a software component that is either offered by vSphere or is developed by a third party through the vSphere APIs for Storage Awareness (VASA). Storage providers integrate with a variety of storage entities that include external physical storage and storage abstractions, such as Virtual SAN and Virtual Volumes.

Generally, vSphere uses storage providers to obtain information about storage topology, status, and storage data services offered in your environment. This information appears in the vSphere Web Client. The information helps you to make appropriate decision about virtual machine placement and to monitor your storage environment.

The third-party storage provider is typically installed on the storage side and acts as a storage awareness service in the vSphere environment.

To find out whether your storage supports the storage provider plug-ins, contact your storage vendor. If your storage supports third-party storage providers, use the vSphere Web Client to register and manage each storage provider component.

Built-in storage providers typically run on the ESXi hosts and do not require registration. For example, the storage provider that supports Virtual SAN becomes registered automatically when you enable Virtual SAN. For more information, see the *Administering VMware Virtual SAN* documentation.

This chapter includes the following topics:

- [“Storage Providers and Storage Data Representation,”](#) on page 262
- [“Storage Provider Requirements and Considerations,”](#) on page 262
- [“Storage Status Reporting,”](#) on page 263
- [“Register Storage Providers,”](#) on page 263
- [“Securing Communication with Storage Providers,”](#) on page 264
- [“View Storage Provider Information,”](#) on page 264
- [“Unregister Storage Providers,”](#) on page 264
- [“Update Storage Providers,”](#) on page 265

Storage Providers and Storage Data Representation

vCenter Server and ESXi communicate with the storage provider to obtain information that the storage provider collects from underlying physical and software-defined storage. vCenter Server can then display the storage data in the vSphere Web Client.

Information that the storage provider supplies can be divided into the following categories:

- Storage data services and capabilities. This type of information is essential for such functionalities as Virtual SAN and Virtual Volumes. The storage provider collects information about data services that are offered by underlying storage entities.

You reference these data services when you define storage requirements for virtual machines and virtual disks in a storage policy. Depending on your environment, the storage policy ensures appropriate storage placement for a virtual machine or enables specific data services for virtual disks. For details, see [“Understanding Virtual Machine Storage Policies,”](#) on page 226.

- Storage status. This category includes reporting about status of various storage entities. It also includes alarms and events for notifying about configuration changes.

This type of information can help you troubleshoot storage connectivity and performance problems. It can also help you to correlate array-generated events and alarms to corresponding performance and load changes on the array.

- Storage DRS information. For distributed resource scheduling (DRS) on block devices or file systems. Storage providers supply additional information about storage system, so that decisions made by Storage DRS are compatible with resource management decisions internal to the storage systems.

Storage Provider Requirements and Considerations

When you use the storage provider functionality, certain requirements and considerations apply.

Typically, vendors are responsible for supplying storage providers that integrate with vSphere. Storage providers are implemented through VMware APIs for Storage Awareness (VASA). The VASA architecture extends the Storage Monitoring Service (SMS), shipped with vSphere, and defines a set of functions that vCenter Server and ESXi hosts can use to communicate with VASA Providers.

To use storage providers, follow these requirements:

- Make sure that every storage provider you use is certified by VMware and properly deployed. For information about deploying the storage providers, contact your storage vendor or check the *VMware Compatibility Guide*.
- Make sure that the storage provider is compatible with the vSphere version. See *VMware Compatibility Guide*

When you use storage providers, the following considerations apply:

- Both block storage and file system storage devices can use storage providers.
- Storage providers can run anywhere, except vCenter Server. Typically, the third-party storage provider runs on either the storage array service processor or on a standalone host.
- Multiple vCenter Servers can simultaneously connect to a single instance of a storage provider.
- A single vCenter Server can simultaneously connect to multiple different storage providers. It is possible to have a different storage provider for each type of physical storage device available to your host.

Storage Status Reporting

If you use storage providers, the vCenter Server can collect status characteristics for physical storage devices and display this information in the vSphere Web Client.

The status information includes events and alarms.

- Events indicate important changes in the storage configuration. Such changes might include creation and deletion of a LUN, or a LUN becoming inaccessible due to LUN masking.
- Alarms indicate a change in storage system availability. For example, when you use profile-based storage management, you can specify virtual machine storage requirements. When changes to underlying storage occur that might violate the storage requirements of the virtual machine, an alarm gets triggered.

For more information about events and alarms, see the *vSphere Monitoring and Performance* documentation.

Thin-provisioned LUNs have special reporting requirements. For information about space monitoring on thin-provisioned LUNs, see [“Array Thin Provisioning and VMFS Datastores,”](#) on page 257.

Register Storage Providers

To establish a connection between vCenter Server and a storage provider, you must register the storage provider. Make sure to register a separate storage provider for each host in a cluster.

NOTE If you use Virtual SAN, the storage providers for Virtual SAN are registered and appear on the list of storage providers automatically. Virtual SAN does not support manual registration of storage providers. See the *Administering VMware Virtual SAN* documentation.

Prerequisites

Verify that the storage provider component is installed on the storage side and obtain its credentials from your storage administrator.

Procedure

- 1 Browse to vCenter Server in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage Providers**.
- 3 Click the **Register a new storage provider** icon.
- 4 Type connection information for the storage provider, including the name, URL, and credentials.
- 5 (Optional) To direct vCenter Server to the storage provider certificate, select the **Use storage provider certificate** option and specify the certificate's location.

If you do not select this option, a thumbprint of the certificate is displayed. You can check the thumbprint and approve it.

- 6 Click **OK** to complete the registration.

vCenter Server has registered the storage provider and established a secure SSL connection with it.

Securing Communication with Storage Providers

To communicate with a storage provider, the vCenter Server uses a secure SSL connection. The SSL authentication mechanism requires that both parties, the vCenter Server and the storage provider, exchange SSL certificates and add them to their truststores.

The vCenter Server can add the storage provider certificate to its truststore as part of the storage provider installation. If the certificate is not added during the installation, use one of the following methods to add it when registering the storage provider:

- Direct the vCenter Server to the storage provider certificate. In the **New Storage Provider** dialog box, select the **Use storage provider certificate** option and specify the certificate's location.
- Use a thumbprint of the storage provider certificate. If you do not direct the vCenter Server to use the provider certificate, the certificate thumbprint is displayed. You can check the thumbprint and approve it. The vCenter Server adds the certificate to the truststore and proceeds with the connection.

The storage provider adds the vCenter Server certificate to its truststore when the vCenter Server first connects to the provider.

View Storage Provider Information

After you register a storage provider component with the vCenter Server, the Storage provider appears on the storage providers list.

View general Storage provider information and details for each storage component.

Procedure

- 1 Browse to vCenter Server in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage Providers**.
- 3 In the Storage Providers list, view the storage provider components registered with the vCenter Server.

The list shows general vendor information including the name, URL, and the time of the last view refresh.

- 4 To display additional details, select a specific storage provider from the list.

The details include storage array vendors and array models that the provider supports.

NOTE A single Storage provider can support storage arrays from multiple different vendors.

Unregister Storage Providers

Unregister storage providers that you do not need.



CAUTION You cannot manually unregister storage providers supplied by VMware, such as Virtual SAN storage providers.

Procedure

- 1 Browse to vCenter Server in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage Providers**.
- 3 From the list of storage providers, select the one you want to unregister and click the **Unregister the storage provider** icon.

vCenter Server terminates the connection and removes the storage provider from its configuration.

Update Storage Providers

The vCenter Server periodically updates storage data in its database. The updates are partial and reflect only those storage changes that storage providers communicate to the vCenter Server. When needed, you can perform a full database synchronisation for the selected storage provider.

Procedure

- 1 Browse to vCenter Server in the vSphere Web Client navigator.
- 2 Click the **Manage** tab, and click **Storage Providers**.
- 3 From the list, select the storage provider that you want to synchronise with and click the **Rescan the storage provider** icon.

The vSphere Web Client updates the storage data for the provider.

Using vmkfstools

`vmkfstools` is one of the ESXi Shell commands for managing VMFS volumes and virtual disks. You can perform many storage operations using the `vmkfstools` command. For example, you can create and manage VMFS datastores on a physical partition, or manipulate virtual disk files, stored on VMFS or NFS datastores.

NOTE After you make a change using the `vmkfstools`, the vSphere Web Client might not be updated immediately. You need to use a refresh or rescan operation from the client.

For more information on the ESXi Shell, see *Getting Started with vSphere Command-Line Interfaces*.

This chapter includes the following topics:

- “[vmkfstools Command Syntax](#),” on page 267
- “[vmkfstools Options](#),” on page 268

vmkfstools Command Syntax

Generally, you do not need to log in as the root user to run the `vmkfstools` commands. However, some commands, such as the file system commands, might require the root user login.

The `vmkfstools` command supports the following command syntax:

`vmkfstools conn_options options target.`

Target specifies a partition, device, or path to apply the command option to.

Table 25-1. `vmkfstools` command arguments

Argument	Description
options	One or more command-line options and associated arguments that you use to specify the activity for <code>vmkfstools</code> to perform, for example, choosing the disk format when creating a new virtual disk. After entering the option, specify a target on which to perform the operation. Target can indicate a partition, device, or path.
partition	Specifies disk partitions. This argument uses a <code>disk_ID:P</code> format, where <code>disk_ID</code> is the device ID returned by the storage array and <code>P</code> is an integer that represents the partition number. The partition digit must be greater than zero (0) and should correspond to a valid VMFS partition.

Table 25-1. `vmkfstools` command arguments (Continued)

Argument	Description
device	<p>Specifies devices or logical volumes. This argument uses a path name in the ESXi device file system. The path name begins with <code>/vmfs/devices</code>, which is the mount point of the device file system.</p> <p>Use the following formats when you specify different types of devices:</p> <ul style="list-style-type: none"> ■ <code>/vmfs/devices/disks</code> for local or SAN-based disks. ■ <code>/vmfs/devices/lvm</code> for ESXi logical volumes. ■ <code>/vmfs/devices/generic</code> for generic SCSI devices.
path	<p>Specifies a VMFS file system or file. This argument is an absolute or relative path that names a directory symbolic link, a raw device mapping, or a file under <code>/vmfs</code>.</p> <ul style="list-style-type: none"> ■ To specify a VMFS file system, use this format: <code>/vmfs/volumes/file_system_UUID</code> or <code>/vmfs/volumes/file_system_label</code> ■ To specify a file on a VMFS datastore, use this format: <code>/vmfs/volumes/file_system_label file_system_UUID/[dir]/myDisk.vmdk</code> <p>You do not need to enter the entire path if the current working directory is the parent directory of <code>myDisk.vmdk</code>.</p>

vmkfstools Options

The `vmkfstools` command has several options. Some of the options are suggested for advanced users only.

The long and single-letter forms of the options are equivalent. For example, the following commands are identical.

```
vmkfstools --createfs vmfs5 --blocksize 1m disk_ID:P
vmkfstools -C vmfs5 -b 1m disk_ID:P
```

-v Suboption

The `-v` suboption indicates the verbosity level of the command output.

The format for this suboption is as follows:

```
-v --verbose number
```

You specify the *number* value as an integer from 1 through 10.

You can specify the `-v` suboption with any `vmkfstools` option. If the output of the option is not suitable for use with the `-v` suboption, `vmkfstools` ignores `-v`.

NOTE Because you can include the `-v` suboption in any `vmkfstools` command line, `-v` is not included as a suboption in the option descriptions.

File System Options

File system options allow you to create and manage VMFS datastores. These options do not apply to NFS. You can perform many of these tasks through the vSphere Web Client.

Listing Attributes of a VMFS Volume

Use the `vmkfstools` command to list attributes of a VMFS volume.

```
-P --queryfs
    -h --humanreadable
```

When you use this option on any file or directory that resides on a VMFS volume, the option lists the attributes of the specified volume. The listed attributes include the file system label, if any, the number of extents comprising the specified VMFS volume, the UUID, and a listing of the device names where each extent resides.

NOTE If any device backing VMFS file system goes offline, the number of extents and available space change accordingly.

You can specify the `-h` suboption with the `-P` option. If you do so, `vmkfstools` lists the capacity of the volume in a more readable form, for example, 5k, 12.1M, or 2.1G.

Creating a VMFS Datastore

Use the `vmkfstools` command to create a VMFS datastore.

```
-C --createfs [vmfs5]
    -S --setfsname datastore
```

This option creates a VMFS5 datastore on the specified SCSI partition, such as `disk_ID:P`. The partition becomes the file system's head partition.

NOTE You cannot create a new VMFS3 datastore, but you can use existing VMFS3 datastores. Upgrade your VMFS3 datastores to VMFS5.

You can specify the following suboptions with the `-C` option:

- `-S --setfsname` – Define the volume label of the VMFS datastore you are creating. Use this suboption only in conjunction with the `-C` option. The label you specify can be up to 128 characters long and cannot contain any leading or trailing blank spaces.

NOTE vCenter Server supports the 80 character limit for all its entities. If a datastore name exceeds this limit, the name gets shortened when you add this datastore to vCenter Server.

After you define a volume label, you can use it whenever you specify the VMFS datastore for the `vmkfstools` command. The volume label appears in listings generated for the `ls -l` command and as a symbolic link to the VMFS volume under the `/vmfs/volumes` directory.

To change the VMFS volume label, use the `ln -sf` command. Use the following as an example:

```
ln -sf /vmfs/volumes/UUID /vmfs/volumes/datastore
```

datastore is the new volume label to use for the *UUID* VMFS.

NOTE If your host is registered with vCenter Server, any changes you make to the VMFS volume label get overwritten by vCenter Server. This guarantees that the VMFS label is consistent across all vCenter Server hosts.

Example for Creating a VMFS File System

This example illustrates creating a new VMFS5 datastore named `my_vmfs` on the `naa.ID:1` partition. The file block size is 1MB.

```
vmkfstools -C vmfs5 -S my_vmfs /vmfs/devices/disks/naa.ID:1
```

Extending an Existing VMFS Volume

Use the `vmkfstools` command to add an extent to a VMFS volume.

```
-Z --spanfs span_partition head_partition
```

This option extends the VMFS file system with the specified head partition by spanning it across the partition specified by `span_partition`. You must specify the full path name, for example `/vmfs/devices/disks/disk_ID:1`. Each time you use this option, you extend a VMFS volume with a new extent so that the volume spans multiple partitions.



CAUTION When you run this option, you lose all data that previously existed on the SCSI device you specified in `span_partition`.

Example for Extending a VMFS Volume

In this example, you extend the logical file system by allowing it to span to a new partition.

```
vmkfstools -Z /vmfs/devices/disks/naa.disk_ID_2:1 /vmfs/devices/disks/naa.disk_ID_1:1
```

The extended file system spans two partitions—`naa.disk_ID_1:1` and `naa.disk_ID_2:1`. In this example, `naa.disk_ID_1:1` is the name of the head partition.

Growing an Existing Extent

Instead of adding a new extent to a VMFS datastore, you can grow an existing extent using the `vmkfstools -G` command.

Use the following option to increase the size of a VMFS datastore after the underlying storage had its capacity increased.

```
-G --growfs device device
```

This option grows an existing VMFS datastore or its extent. For example,

```
vmkfstools --growfs /vmfs/devices/disks/disk_ID:1 /vmfs/devices/disks/disk_ID:1
```

Upgrading a VMFS Datastore

If you use a VMFS3 datastore, you must upgrade it to VMFS5.



CAUTION The upgrade is a one-way process. After you have converted a VMFS3 datastore to VMFS5, you cannot revert it back.

When upgrading the datastore, use the following command: `vmkfstools -T|--upgradevmfs /vmfs/volumes/UUID`

NOTE All hosts accessing the datastore must support VMFS5. If any ESX/ESXi host version 4.x or earlier is using the VMFS3 datastore, the upgrade fails and the Mac address of the host that is actively using the datastore is displayed.

Virtual Disk Options

Virtual disk options allow you to set up, migrate, and manage virtual disks stored in VMFS and NFS file systems. You can also perform most of these tasks through the vSphere Web Client.

Supported Disk Formats

When you create or clone a virtual disk, you can use the `-d --diskformat` suboption to specify the format for the disk.

Choose from the following formats:

- `zeroedthick` (default) – Space required for the virtual disk is allocated during creation. Any data remaining on the physical device is not erased during creation, but is zeroed out on demand at a later time on first write from the virtual machine. The virtual machine does not read stale data from disk.
- `eagerzeroedthick` – Space required for the virtual disk is allocated at creation time. In contrast to `zeroedthick` format, the data remaining on the physical device is zeroed out during creation. It might take much longer to create disks in this format than to create other types of disks.
- `thin` – Thin-provisioned virtual disk. Unlike with the `thick` format, space required for the virtual disk is not allocated during creation, but is supplied, zeroed out, on demand at a later time.
- `rdm:device` – Virtual compatibility mode raw disk mapping.
- `rdmp:device` – Physical compatibility mode (pass-through) raw disk mapping.
- `2gbsparse` – A sparse disk with 2GB maximum extent size. You can use disks in this format with hosted VMware products, such as VMware Fusion, Player, Server, or Workstation. However, you cannot power on sparse disk on an ESXi host unless you first re-import the disk with `vmkfstools` in a compatible format, such as `thick` or `thin`.

See “[Migrate Virtual Machines Between Different VMware Products](#),” on page 273.

NFS Disk Formats

The only disk formats you can use for NFS are `thin`, `thick`, `zeroedthick` and `2gbsparse`.

`Thick`, `zeroedthick` and `thin` formats usually behave the same because the NFS server and not the ESXi host determines the allocation policy. The default allocation policy on most NFS servers is `thin`. However, on NFS servers that support Storage APIs - Array Integration, you can create virtual disks in `zeroedthick` format. The `reserve space` operation enables NFS servers to allocate and guarantee space.

For more information on array integration APIs, see [Chapter 22, “Storage Hardware Acceleration,”](#) on page 243.

Creating a Virtual Disk

Use the `vmkfstools` command to create a virtual disk.

```
-c --createvirtualdisk size[kK|mM|gG]
  -a --adaportype [buslogic|lsilogic|ide|lsisas|pvscsi] srcfile
  -d --diskformat [thin|zeroedthick|eagerzeroedthick]
```

This option creates a virtual disk at the specified path on a datastore. Specify the size of the virtual disk. When you enter the value for `size`, you can indicate the unit type by adding a suffix of `k` (kilobytes), `m` (megabytes), or `g` (gigabytes). The unit type is not case sensitive. `vmkfstools` interprets either `k` or `K` to mean kilobytes. If you don't specify a unit type, `vmkfstools` defaults to bytes.

You can specify the following suboptions with the `-c` option.

- `-a` specifies the controller that a virtual machine uses to communicate with the virtual disks. You can choose between BusLogic, LSI Logic, IDE, LSI Logic SAS, and VMware Paravirtual SCSI.

- `-d` specifies disk formats.

Example for Creating a Virtual Disk

This example illustrates creating a two-gigabyte virtual disk file named `rh6.2.vmdk` on the VMFS file system named `myVMFS`. This file represents an empty virtual disk that virtual machines can access.

```
vmkfstools -c 2048m /vmfs/volumes/myVMFS/rh6.2.vmdk
```

Initializing a Virtual Disk

Use the `vmkfstools` command to initialize a virtual disk.

```
-w --writezeros
```

This option cleans the virtual disk by writing zeros over all its data. Depending on the size of your virtual disk and the I/O bandwidth to the device hosting the virtual disk, completing this command might take a long time.



CAUTION When you use this command, you lose any existing data on the virtual disk.

Inflating a Thin Virtual Disk

Use the `vmkfstools` command to inflate a thin virtual disk.

```
-j --inflatedisk
```

This option converts a thin virtual disk to `eagerzeroedthick`, preserving all existing data. The option allocates and zeroes out any blocks that are not already allocated.

Removing Zeroed Blocks

Use the `vmkfstools` command to convert any thin, zeroedthick, or eagerzeroedthick virtual disk to a thin disk with zeroed blocks removed.

```
-K --punchzero
```

This option deallocates all zeroed out blocks and leaves only those blocks that were allocated previously and contain valid data. The resulting virtual disk is in thin format.

Converting a Zeroedthick Virtual Disk to an Eagerzeroedthick Disk

Use the `vmkfstools` command to convert any zeroedthick virtual disk to an eagerzeroedthick disk.

```
-k --eagerzero
```

While performing the conversion, this option preserves any data on the virtual disk.

Deleting a Virtual Disk

This option deletes a virtual disk file at the specified path on the VMFS volume.

```
-U --deletevirtualdisk
```

Renaming a Virtual Disk

This option renames a virtual disk file at the specified path on the VMFS volume.

You must specify the original file name or file path *oldName* and the new file name or file path *newName*.

```
-E --renamevirtualdisk oldName newName
```


Cloning a Virtual Disk or RDM

This option creates a copy of a virtual disk or raw disk you specify.

```
-i --clonevirtualdisk srcfile -d --diskformat [zeroedthick|thin|eagerzeroedthick|rdm:device|rdmp:device|2gbsparse]
```

You can use the `-d|--diskformat` suboption to specify the disk format for the copy you create. A non-root user is not allowed to clone a virtual disk or an RDM.

Example for Cloning a Virtual Disk

This example illustrates cloning the contents of a master virtual disk from the templates repository to a virtual disk file named `myOS.vmdk` on the `myVMFS` file system.

```
vmkfstools -i /vmfs/volumes/myVMFS/templates/gold-master.vmdk /vmfs/volumes/myVMFS/myOS.vmdk
```

You can configure a virtual machine to use this virtual disk by adding lines to the virtual machine configuration file, as in the following example:

```
scsi0:0.present = TRUE
scsi0:0.fileName = /vmfs/volumes/myVMFS/myOS.vmdk
```

Migrate Virtual Machines Between Different VMware Products

Typically, you use VMware Converter to migrate virtual machines from other VMware products into your ESXi system. However, you can use the `vmkfstools -i` command to import virtual disks in 2gbsparse format into ESXi and then attach this disk to a new virtual machine you create in ESXi.

You must import the virtual disk first because you cannot power on disks in 2gbsparse format on the ESXi host.

Procedure

- 1 Import a disk in 2gbsparse format into the ESXi host by running the following command. Make sure to select the disk format compatible with ESXi.

```
vmkfstools -i <input> <output> -d <format>
```

- 2 Use the vSphere Web Client to attach the disk you imported to a virtual machine in ESXi.

For information, see the *vSphere Virtual Machine Administration*.

Extending a Virtual Disk

This option extends the size of a disk allocated to a virtual machine after the virtual machine has been created.

```
-X --extendvirtualdisk newSize [kK|mM|gG]
```

You must power off the virtual machine that uses this disk file before you enter this command. You might have to update the file system on the disk so the guest operating system can recognize and use the new size of the disk and take advantage of the extra space.

NOTE You cannot hot-extend virtual SATA disks, or any virtual disk if the capacity after extension is equal to or greater than 2TB.

You specify the `newSize` parameter in kilobytes, megabytes, or gigabytes by adding a `k` (kilobytes), `m` (megabytes), or `g` (gigabytes) suffix. The unit type is not case sensitive. `vmkfstools` interprets either `k` or `K` to mean kilobytes. If you don't specify a unit type, `vmkfstools` defaults to kilobytes.

The `newSize` parameter defines the entire new size, not just the increment you add to the disk.

For example, to extend a 4g virtual disk by 1g, enter: `vmkfstools -X 5g disk name`.

You can extend the virtual disk to the `eagerzeroedthick` format by using the `-d eagerzeroedthick` option.

NOTE Do not extend the base disk of a virtual machine that has snapshots associated with it. If you do, you can no longer commit the snapshot or revert the base disk to its original size.

Upgrading Virtual Disks

This option converts the specified virtual disk file from ESX Server 2 format to the ESXi format.

`-M --migratevirtualdisk`

Creating a Virtual Compatibility Mode Raw Device Mapping

This option creates a Raw Device Mapping (RDM) file on a VMFS volume and maps a raw LUN to this file. After this mapping is established, you can access the LUN as you would a normal VMFS virtual disk. The file length of the mapping is the same as the size of the raw LUN it points to.

`-r --createrdm device`

When specifying the `device` parameter, use the following format:

`/vmfs/devices/disks/disk_ID:P`

Example for Creating a Virtual Compatibility Mode RDM

In this example, you create an RDM file named `my_rdm.vmdk` and map the `disk_ID` raw disk to that file.

```
vmkfstools -r /vmfs/devices/disks/disk_ID my_rdm.vmdk
```

You can configure a virtual machine to use the `my_rdm.vmdk` mapping file by adding the following lines to the virtual machine configuration file:

```
scsi0:0.present = TRUE
scsi0:0.fileName = /vmfs/volumes/myVMFS/my_rdm.vmdk
```

Creating a Physical Compatibility Mode Raw Device Mapping

This option lets you map a pass-through raw device to a file on a VMFS volume. This mapping lets a virtual machine bypass ESXi SCSI command filtering when accessing its virtual disk. This type of mapping is useful when the virtual machine needs to send proprietary SCSI commands, for example, when SAN-aware software runs on the virtual machine.

`-z --createrdmpassthru device`

After you establish this type of mapping, you can use it to access the raw disk just as you would any other VMFS virtual disk.

When specifying the `device` parameter, use the following format:

`/vmfs/devices/disks/disk_ID`

Listing Attributes of an RDM

This option lets you list the attributes of a raw disk mapping.

`-q --queryrdm`

This option prints the name of the raw disk RDM. The option also prints other identification information, like the disk ID, for the raw disk.

Displaying Virtual Disk Geometry

This option gets information about the geometry of a virtual disk.

```
-g --geometry
```

The output is in the form: `Geometry information C/H/S`, where C represents the number of cylinders, H represents the number of heads, and S represents the number of sectors.

NOTE When you import virtual disks from hosted VMware products to the ESXi host, you might see a disk geometry mismatch error message. A disk geometry mismatch might also be the cause of problems loading a guest operating system or running a newly-created virtual machine.

Checking and Repairing Virtual Disks

Use this option to check or repair a virtual disk in case of an unclean shutdown.

```
-x , --fix [check|repair]
```

Checking Disk Chain for Consistency

With this option, you can check the entire disk chain. You can determine if any of the links in the chain are corrupted or any invalid parent-child relationships exist.

```
-e --chainConsistent
```

Storage Device Options

Device options allows you to perform administrative task for physical storage devices.

Managing SCSI Reservations of LUNs

The `-L` option lets you reserve a SCSI LUN for exclusive use by the ESXi host, release a reservation so that other hosts can access the LUN, and reset a reservation, forcing all reservations from the target to be released.

```
-L --lock [reserve|release|lunreset|targetreset|busreset] device
```



CAUTION Using the `-L` option can interrupt the operations of other servers on a SAN. Use the `-L` option only when troubleshooting clustering setups.

Unless specifically advised by VMware, never use this option on a LUN hosting a VMFS volume.

You can specify the `-L` option in several ways:

- `-L reserve` – Reserves the specified LUN. After the reservation, only the server that reserved that LUN can access it. If other servers attempt to access that LUN, a reservation error results.
- `-L release` – Releases the reservation on the specified LUN. Other servers can access the LUN again.
- `-L lunreset` – Resets the specified LUN by clearing any reservation on the LUN and making the LUN available to all servers again. The reset does not affect any of the other LUNs on the device. If another LUN on the device is reserved, it remains reserved.
- `-L targetreset` – Resets the entire target. The reset clears any reservations on all the LUNs associated with that target and makes the LUNs available to all servers again.
- `-L busreset` – Resets all accessible targets on the bus. The reset clears any reservation on all the LUNs accessible through the bus and makes them available to all servers again.

When entering the *device* parameter, use the following format:

```
/vmfs/devices/disks/disk_ID:P
```

Breaking Device Locks

The `-B` option allows you to forcibly break the device lock on a particular partition.

```
-B --breaklock device
```

When entering the *device* parameter, use the following format:

```
/vmfs/devices/disks/disk_ID:P
```

You can use this command when a host fails in the middle of a datastore operation, such as grow extent, add extent, or resignaturing. When you issue this command, make sure that no other host is holding the lock.

Index

Symbols

* next to path **189**

A

access control **66**
accessing storage **24**
active-active disk arrays **36, 40, 66, 70, 191**
active-passive disk arrays, boot from SAN **50**
adaptive scheme **30**
adding, NFS storage **161**
adding an ESXi host to an Active Directory domain **160**
adding tag-based rules **231**
advanced attributes, hosts **179**
advanced settings
 Disk.EnableNaviReg **60**
 Disk.MaxLUN **125**
 VMkernel.Boot.isPerFileSchedModelActive **200**
alarms **263**
all paths down **129, 130**
all paths down event **126**
allocations, LUN **40**
allocations, LUN **70**
APD
 handling **129**
 Misc.APDHandlingEnable parameter **129**
 Misc.APDTimeout **130**
 with High Availability **131**
applications, layered **31**
applying tags to datastores **229**
array integration, thin provisioning **257**
array-based solution **31**
asterisk next to path **189**
ATS-only locking, upgrading **151**
ATS-only upgrade
 offline **150**
 online **150**
 prerequisites **151**
ATS-only locking, downgrading to ATS +SCSI **152**
authentication **66, 98**
automatic host registration, disabling **60**

B

backups
 considerations **32**
 third-party backup package **33**
best practices, FCoE **46**
bidirectional CHAP **98**
BIOS, enabling for BFS **52**
block devices **206**
boot adapters **51**
boot BIOS prompt, enabling for BFS **52**
boot from DVD-ROM **51**
boot from iSCSI SAN
 configuring HBAs **108**
 configuring iSCSI settings **109**
 guidelines **107**
 hardware iSCSI **108**
 iBFT **109**
 preparing SAN **108**
 software iSCSI **109**
boot from SAN
 benefits **49**
 boot LUN considerations **50**
 configuring Emulex HBAs **52**
 configuring Qlogic HBAs **53**
 configuring storage **50**
 HBA requirements **50**
 host requirements **50**
 overview **49**
 preparing installation **50**
 requirements **50**

C

CHAP
 bidirectional **98**
 disabling **101**
 for discovery targets **100**
 for iSCSI initiators **99**
 for static targets **100**
 one-way **98**
CHAP authentication **66, 98**
CHAP authentication methods **98**
claim rules **189**
cloning storage policies **232**

- compatibility modes
 - physical **206**
 - virtual **206**
- configuration files, virtual machines **131**
- configuration parameters, virtual machines **131**
- configure a swap cache using a virtual flash resource **137**
- configure Flash Read Cache **142**
- configuring, dynamic discovery **73, 80, 86, 97**
- configuring DNS for NFS 4.1 **159**
- conversion of storage profiles **225**
- copying files **170**
- core dump files
 - creating **175**
 - deleting **177**
- creating tags **229**
- creating VVOL datastores **162, 219**
- current multipathing state **190**

D

- data digests **67**
- datastore browser **169**
- datastores
 - adding extents **166**
 - administrative operations **167**
 - configuring on NFS volumes **161**
 - copying files on **170**
 - creating **160**
 - displaying **21**
 - displaying for a parent object **23**
 - increasing capacity **166**
 - managing duplicate **162**
 - mounting **163**
 - moving files on **171**
 - NFS **145**
 - paths **190**
 - refreshing **124**
 - renaming **167**
 - renaming files on **171**
 - review information **22**
 - storage over-subscription **257**
 - unmounting **168**
 - VMFS **145**
 - VVols **162, 219**
- default storage policies
 - provided by VMware **233**
 - user-created **234**
- default storage policy, changing **234**
- deleting storage policy **232**
- dependent iSCSI, networking **88**
- dependent hardware iSCSI
 - and associated NICs **76**
 - configuration **75**

- configuration workflow **74**
- considerations **74**
- device loss, unplanned **126, 128**
- device locks, breaking **276**
- diagnostic partition
 - configuring **174**
 - verifying **175**
- diagnostic partitions **39, 70**
- disaster recovery **28**
- discovery
 - address **97**
 - dynamic **73, 80, 86, 97**
- disk arrays
 - active-active **40, 70, 191**
 - active-passive **40, 70, 191**
- disk chain, consistency **275**
- disk formats
 - thick provisioned **254**
 - thin provisioned **254**
- disk mirroring **173**
- disk timeout **185**
- Disk.EnableNaviReg **60**
- Disk.MaxLUN, and Protocol Endpoints **125**
- disks
 - format **256**
 - inflate **171, 256**
- DRS support for virtual flash **142**
- dump partitions **39, 70**
- DVD-ROM, booting from **51**
- dynamic discovery, configuring **73, 80, 86, 97**
- dynamic discovery addresses **97**
- dynamic disks **173**

E

- editing storage policies **232**
- educational support **9**
- enabling Kerberos users **160**
- esxcli commands, and
 - isPerFileSchedModelActive **201**
- esxcli commands, obtaining VMFS locking information **149**
- ESXi host, and multiple storage arrays **29**
- ESXi NFS protocols **154**
- EUI **64**
- events **263**
- examples
 - vmkfstools -C **270**
 - vmkfstools -Z **270**
- extents
 - adding to datastore **166**
 - growing **166**

F

- failover
 - I/O delay **184**
 - transparent **36, 66**
- failover paths, status **189**
- FC HBA setup **40**
- FC SAN
 - accessing **37**
 - hardware requirements **39**
- FCoE, best practices **46**
- FCoE adapters **45**
- Fibre Channel, concepts **35**
- Fibre Channel over Ethernet **45**
- Fibre Channel SAN
 - best practices **59**
 - preventing problems **59**
- file systems, upgrading **146**
- file-based (VMFS) solution **32**
- files
 - copying **170**
 - moving **171**
- FIP **46**
- firewall, NFS client **156**
- Fixed path policy **188, 191**
- flash device, monitoring **136**
- flash devices
 - best Practices **136**
 - estimating lifetime **136**
 - marking **135**
 - using with vSphere **134**
- Flash Read Cache
 - migrating virtual machines **143**
 - migration settings **143**
- flash virtual disks **134**

G

- GPT **18, 121**
- guidelines, NFS storage **153**

H

- HA support for virtual flash **142**
- hardware acceleration
 - about **243**
 - benefits **243**
 - block storage **244**
 - deleting claim rules **249**
 - enabling **245**
 - NAS **249**
 - NAS status **251**
 - requirements **244**
 - status **244**
 - support details **246**
- hardware acceleration, considerations **251**

- hardware iSCSI, and failover **182**
- hardware iSCSI initiators
 - configuring **71**
 - installing **72**
 - setting up discovery addresses **97**
 - viewing **72**
- hardware iSCSI adapters
 - dependent **65**
 - independent **65**
- HBAs
 - queue depth **39**
 - setup **40**
 - static load balancing **40**
- header digests **67**
- high-tier storage **30**
- host registration, disabling **60**
- host cache, swapping to **139**
- host cache, configuring on a VMFS
 - datastore **139**
- host cache, configuring with virtual flash
 - resource **140**
- host configuration, advanced settings **60**
- host-based failover **181**
- hosts
 - advanced attributes **179**
 - and FC SAN **35**

I

- I/O delay **70, 184**
- iBFT **109**
- iBFT iSCSI boot
 - booting an ESXi host **112**
 - changing boot sequence **111**
 - installing an ESXi host **111**
 - limitations **110**
 - networking best practices **112**
 - setting up ESXi **110**
 - troubleshooting **113**
- IDE **14**
- independent hardware iSCSI adapters, change
 - IP address **73**
- installation
 - preparing for boot from SAN **50**
 - steps **40**
- IP address **64**
- IQN **64**
- iSCSI **15**
- iSCSI initiators
 - configuring CHAP **99**
 - configuring advanced parameters **103**
 - hardware **71**
 - setting up CHAP parameters **98**
- iSCSI SAN
 - accessing **68**

- best practices **115**
- boot **107**
- concepts **63**
- preventing problems **115**
- iSCSI adapter, modifying general properties **72, 76, 82, 87**
- iSCSI adapters
 - about **69**
 - advanced parameters **102**
 - hardware **65**
 - software **65**
- iSCSI alias **64**
- iSCSI boot, iBFT **109**
- iSCSI Boot Firmware Table, *See* iBFT
- iSCSI boot parameters, configuring **111**
- iSCSI names, conventions **64**
- iSCSI networking
 - binding adapters **79, 85, 94**
 - changing policy **79, 85, 93**
 - creating a VMkernel interface **77, 83, 92**
 - managing **95**
 - port binding details **80, 86, 94**
 - troubleshooting **95**
- iSCSI port binding, considerations **91**
- iSCSI ports **64**
- iSCSI SAN restrictions **70**
- iSCSI sessions
 - adding for a target **104**
 - displaying **104**
 - managing **103**
 - removing **105**

J

- jumbo frames
 - enabling for dependent hardware iSCSI **96**
 - enabling for independent hardware iSCSI **96**
 - enabling for software iSCSI **96**
 - using with iSCSI **95**

K

- Kerberos, configuring ESXi hosts **158**

L

- Layer 3 connections **157**
- layered applications **31**
- load balancing **28, 40**
- locations of virtual machines **30**
- locator LED
 - turning off **132**
 - turning on **132**
- loss of network connection, troubleshooting **113**
- lower-tier storage **30**

- LUN decisions
 - adaptive scheme **30**
 - predictive scheme **30**
- LUN masking **35**
- LUNs
 - allocations **40, 70**
 - and VMFS datastores **39**
 - changing number scanned **125**
 - decisions **29**
 - making changes and rescan **124**
 - masking **197**
 - multipathing policy **191**
 - NPIV-based access **41**
 - one VMFS volume per **70**
 - setting multipathing policy **191**

M

- maintenance **28**
- marking, flash devices **135**
- masking LUNs **197**
- MBR **18, 121**
- metadata, RDMs **206**
- metadata consistency, checking with VOMA **177**
- metadata updates **148**
- mid-tier storage **30**
- migrating virtual machines with Flash Read Cache **143**
- migration
 - cross-host Storage vMotion **254**
 - storage **254**
- Most Recently Used path policy **188, 191**
- mounting VMFS datastores **168**
- MPPs
 - displaying **194**
 - See also* multipathing plug-ins
- MRU path policy **191**
- multipathing
 - active paths **189**
 - broken paths **189**
 - considerations **192**
 - disabled paths **189**
 - standby paths **189**
 - viewing the current state of **189**
- multipathing claim rules
 - adding **195**
 - deleting **197**
- multipathing plug-ins, path claiming **189**
- multipathing policy **191**
- multipathing state **190**

N

- N-Port ID Virtualization, *See* NPIV
- NAA **64**

- NAS **15**
 - NAS plug-ins
 - installing **249**
 - uninstalling **250**
 - upgrading **251**
 - Native Multipathing Plug-In **186, 187**
 - network connections, create **77, 83, 91**
 - network adapters, configuring for iBFT iSCSI boot **111**
 - network performance **117**
 - networking, configuring **71**
 - NFS datastores
 - characteristics **21**
 - maximum size **152**
 - unmounting **168**
 - NFS storage
 - adding **161**
 - firewall **155**
 - guidelines **153**
 - setting up **157**
 - NFS 4.1
 - Kerberos credentials **158**
 - Network Time Protocol **159**
 - NFS client, firewall rule set **156**
 - NFS clients, firewall ports **156**
 - NICs, mapping to VMkernel **78, 84, 92**
 - NMP
 - I/O flow **188**
 - path claiming **189**
 - See also* Native Multipathing Plug-In
 - NPIV
 - about **41**
 - assigning WWNs **42**
 - changing WWNs **43**
 - limitations **42**
 - requirements **41**
- O**
- one-way CHAP **98**
- P**
- partition mappings **206**
 - passive disk arrays **40, 70, 191**
 - path claiming **189**
 - path failover
 - and virtual machines **185**
 - array-based **184**
 - host-based **182**
 - path failure rescan **124**
 - path management **181**
 - path policies
 - Fixed **184, 188, 191**
 - Most Recently Used **188, 191**
 - MRU **191**
 - Round Robin **188, 191**
 - path selection policies, changing **191**
 - Path Selection Plug-Ins **188**
 - paths
 - disabling **192**
 - masking **197**
 - preferred **189**
 - unmasking **198**
 - PDL
 - with High Availability **131**
 - See also* permanent device loss
 - per file I/O scheduling
 - about **200**
 - turning off **200**
 - performance
 - checking Ethernet switch statistics **119**
 - network **117**
 - optimizing **60, 116**
 - storage system **116**
 - permanent device loss **126, 128**
 - planned device removal **127**
 - Pluggable Storage Architecture **186**
 - pointer block cache, configuring **179**
 - policy rules, storage-specific data services **226**
 - port binding **182**
 - port redirection **184**
 - Port_ID **36**
 - predictive scheme **30**
 - preferred path **189**
 - profile-driven storage **225**
 - protocol endpoints
 - editing paths **220**
 - managing **220**
 - PSA, *See* Pluggable Storage Architecture
 - PSPs, *See* Path Selection Plug-Ins
- Q**
- Qlogic HBA BIOS, enabling for BFS **53**
 - queue depth **70**
- R**
- RAID devices **206**
 - raw device mapping, *see* RDM **203**
 - RDM
 - advantages **204**
 - and virtual disk files **207**
 - dynamic name resolution **207**
 - overview **203**
 - physical compatibility mode **206**
 - virtual compatibility mode **206**
 - with clustering **207**

- RDMs
 - and snapshots **206**
 - path management **209**
 - reclaiming space **259**
 - remove a virtual flash resource **138**
 - renaming files **171**
 - requirements, boot from SAN **50**
 - rescan
 - LUN creation **124**
 - path masking **124**
 - when path is down **124**
 - rescanning
 - storage **125**
 - storage adapter **125**
 - resignature a VMFS datastore copy **163**
 - resignaturing **162**
 - restrictions **39**
 - Round Robin path policy **188, 191**
 - rule sets **227**
 - rules, and datastore tags **227**
- S**
- SAN
 - backup considerations **32**
 - benefits **27**
 - requirements **39**
 - specifics **28**
 - SAN fabric **35**
 - SAN management software **32**
 - SAN storage performance, optimizing **60, 116**
 - SAS **14**
 - SATA **14**
 - SATPs
 - adding rules **199**
 - displaying **194**
 - See *also* Storage Array Type Plug-Ins
 - scanning, changing number **125**
 - scheduling queues **200**
 - SCSI, vmkfstools **267**
 - SCSI controllers **13**
 - sense codes **126**
 - server performance **61, 116**
 - set up virtual flash resource **137**
 - setup steps **40**
 - software iSCSI
 - and failover **182**
 - diagnostic partition **174**
 - networking **88**
 - software FCoE
 - and VMkernel **46**
 - activating adapters **47**
 - booting **55**
 - software FCoE boot
 - best practices **56**
 - configuring parameters **56**
 - ESXi installation **57**
 - requirements **55**
 - setting up **56**
 - software FCoE installation, troubleshooting **57**
 - software iSCSI adapter
 - configuring **81**
 - disabling **87**
 - software iSCSI initiator, enabling **81**
 - software iSCSI boot, changing settings **113**
 - software iSCSI initiators, setting up discovery addresses **97**
 - standard switches **78, 84, 92**
 - static targets, removing **98**
 - static discovery addresses **97**
 - storage
 - access for virtual machines **24**
 - adapters **20**
 - introduction **13**
 - local **14**
 - networked **15**
 - provisioning **253**
 - rescan **125**
 - supported vSphere features **24**
 - types **14**
 - used by virtual machines **255**
 - storage adapter, rescan **125**
 - storage devices
 - attaching **128**
 - detaching **128**
 - disconnections **126**
 - displaying **195**
 - displaying for a host **19, 122**
 - displaying for an adapter **19, 122**
 - hardware acceleration status **246**
 - managing **121**
 - marking as local flash **134**
 - naming **123**
 - paths **190**
 - viewing **18, 121**
 - storage filters
 - disabling **172**
 - host rescan **173**
 - RDM **173**
 - same host and transports **173**
 - VMFS **173**
 - storage policies
 - cloning **232**
 - editing **232**
 - noncompliant **237**
 - reapplying **238**

- rule sets **227**
- virtual machines **233**
- storage profiles, conversion **225**
- storage provider, requirements **262**
- storage adapters
 - viewing **20**
 - viewing in vSphere Web Client **20**
- Storage APIs, Storage Awareness **261**
- storage area network **63**
- Storage Array Type Plug-Ins **187**
- storage arrays, performance **60**
- storage capabilities **262**
- storage capabilities and storage policies **226**
- storage compliance
 - storage policy **236**
 - virtual machine **237**
- storage device
 - connection status **130**
 - renaming **124**
- storage devices, marking as local **135**
- storage policy
 - applying to a virtual machine **235**
 - compliance **236**
 - deleting **232**
 - enabling **228**
- storage providers
 - and VVols **213**
 - SSL connection **264**
 - unregistering **264**
 - updating **265**
 - viewing **264**
- storage providers, registering **263**
- storage space **253**
- storage status **262**
- storage systems
 - performance **116**
 - types **36, 66**
- storage topology **262**
- storage virtualization **13**
- storage, and VMkernel **239**
- STP **46**
- swap cache, configure with virtual flash
 - resource **137**
- swap to host cache **139**

T

- tag-based rules, adding **231**
- tags
 - applying to datastores **229**
 - creating **229**
- tape devices **40**
- targets **17, 65**
- targets vs. LUNs **65**
- technical support **9**

- thick provisioning **253**
- thin disks, creating **255**
- thin provisioning, over-subscription **253**
- thin-provisioned LUNs
 - identify **258**
 - reporting **258**
- third-party backup package **33**
- third-party management applications **32**
- TimeoutValue parameter **39, 70**
- troubleshooting
 - changing iSCSI boot parameters **113**
 - loss of network connection **113**

U

- unplanned device loss **126, 128**
- updated information **11**
- uploading files **170**
- USB **14**
- use cases **28**

V

- VAAI claim rules
 - defining **248**
 - deleting **249**
 - VAAI filter **247**
 - VAAI plug-in **247**
- VAAI filter, displaying **245**
- VAAI plug-ins, displaying **245**
- VFFS volumes, about **137**
- VFFS volume, setting up **137**
- virtual datastores
 - characteristics **21**
 - default policies **233**
 - mounting **168**
 - unmounting **168**
- virtual flash
 - disabling **138**
 - DRS support **142**
 - HA support **142**
- Virtual SAN datastores, characteristics **21**
- Virtual Volumes
 - changing default policy **223**
 - guidelines **215**
- virtual datastore, changing default policy **223**
- virtual disk, repair **275**
- virtual disk files
 - copying **170**
 - renaming **171**
- virtual disks
 - extending **273**
 - formats **254**
 - supported formats **271**
- virtual flash resource
 - configure a swap cache **137**
 - considerations **137**

- remove **138**
- setup **137**
- virtual flash resource, configuring host
 - cache **140**
- virtual machine I/O, queues **200**
- virtual machine storage policies, default **233**
- virtual machine storage policy, compliance **237**
- virtual machine storage profiles, *See* storage policies
- virtual machines
 - accessing FC SAN **37**
 - accessing iSCSI SAN **68**
 - assigning WWNs to **42**
 - configuration files **131**
 - configuration parameters **131**
 - I/O delay **184**
 - locations **30**
 - with RDMs **208**
- virtual ports (VPORTs) **41**
- Virtual SAN, default policies **233**
- virtual volumes
 - and virtual disks **212**
 - assigning VM storage policy **222**
 - VM storage policies **221**
- Virtual Volumes functionality
 - about
 - See* VVols functionality
- VM Component Protection **131**
- VM storage polices, workflow **228**
- VM storage policies
 - creating **228**
 - defining storage-specific rules **230**
 - managing **228**
 - reviewing compatible datastores **232**
 - rules **226**
- VM storage policy
 - assigning **234**
 - starting creation **230**
- VMFS
 - checking metadata consistency **177**
 - conversion **270**
 - locking **149**
 - one volume per LUN **70**
 - resignaturing **162**
 - vmkfstools **267**
- VMFS datastores
 - adding extents **166**
 - changing signatures **163**
 - characteristics **21**
 - creating on Fibre Channel storage **160**
 - creating on iSCSI storage **160**
 - creating on SCSI disk **160**
 - creating on storage device **160**
 - deleting **169**
 - disk formats **147**
 - increasing **165**
 - increasing capacity **166**
 - mounting **168**
 - sharing **148**
 - unmounting **168**
 - upgrading **164**
- VMFS datastore, configuring host cache **139**
- VMFS resignaturing **162**
- VMFS3, upgrading **165**
- VMFS5, improvements over VMFS3 **146**
- VMFS5 and VMFS3, differences **146**
- VMkernel interfaces **78, 84, 92**
- vmkfstools
 - breaking locks **276**
 - cloning disks **273**
 - creating RDMs **274**
 - creating virtual disks **271**
 - deleting virtual disks **272**
 - device options **275**
 - disk chain **275**
 - extending virtual disks **273**
 - file system options **269**
 - geometry **275**
 - inflating thin disks **272**
 - initializing virtual disks **272**
 - migrating virtual disks **273**
 - overview **267**
 - RDM attributes **274**
 - removing zeroed blocks **272**
 - renaming virtual disks **272**
 - SCSI reservations **275**
 - syntax **267**
 - upgrading virtual disks **274**
 - virtual disk options **271**
 - virtual disks conversion **272**
- vmkfstools -C command **269**
- vmkfstools -G command **270**
- vmkfstools -P command **269**
- vmkfstools -v command **268**
- vmkfstools -Z command **270**
- vmkfstools command options **268**
- vmkfstools examples
 - cloning disks **273**
 - creating RDMs **274**
 - creating virtual disks **272**
- vMotion **27, 28, 40, 70**
- VMware DRS, using with vMotion **70**
- VMware HA **27**
- VMware NMP
 - I/O flow **188**
 - See also* Native Multipathing Plug-In

VMware On-disk Metadata Analyser, *See* VOMA
 VMware PSPs, *See* Path Selection Plug-Ins
 VMware SATPs, *See* Storage Array Type Plug-Ins

VOMA **177**

VVOL **211, 215**

VVOL datastores
 characteristics **21**
 unmounting **168**

VVOL functionality
 and storage protocols **215**
 concepts **212**
 prerequisites **217**
 protocol endpoints **214**
 storage containers **213**
 workflow **218**

See also Virtual Volumes

VVOL storage providers **219**

VVOL VASA providers **213**

VVols, VM storage policies **221**

VVols datastores
 creating VMs **221**
 mounting **168**

VVols functionality
 about **211**
 and NTP server **217**
 and storage protocols **215**
 architecture **216**
 characteristics **215**
 datastores **214**
 limitations **215**
 prerequisites **217**
 protocol endpoints **214**
 storage containers **213**
 workflow **218**

VVols functionality, concepts **212**

VVols protocol endpoints, editing paths **220**

VVols storage providers, registering **219**

VVols VASA providers **213**

W

Windows guest OS timeout **185**

World Wide Names, *See* WWNs

World Wide Port Names, *See* WWPNs

WWNNs **42**

WWNs
 assigning to virtual machines **42**
 changing **43**

WWPNs **36, 42**

Z

zoning **35, 36**

