

# vSphere Command-Line Interface Concepts and Examples

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-001470-00

**vmware**<sup>®</sup>

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](mailto:docfeedback@vmware.com)

Copyright © 2008 - 2015 VMware, Inc. All rights reserved. [Copyright and trademark information](#)

**VMware, Inc.**  
3401 Hillview Ave.  
Palo Alto, CA 94304  
[www.vmware.com](http://www.vmware.com)

# Contents

About This Book	9
<b>1 vSphere CLI Command Overviews</b>	<b>11</b>
Introduction	11
Documentation	12
Command-Line Help	12
List of Available Host Management Commands	12
Targets and Protocols for vCLI Host Management Commands	14
Supported Platforms for vCLI Commands	15
Commands with an esxcfg Prefix	16
ESXCLI Overview	16
ESXCLI Commands Available on Different ESXi Hosts	16
Trust Relationship Requirement for ESXCLI Commands	17
Downloading and Installing the vCenter Server Certificate	17
Using the --cacertsfile Option	17
Using the --thumbprint Option	17
Using the Credential Store	17
Using ESXCLI Output	18
Connection Options for vCLI Host Management Commands	18
Connection Options for DCLI Commands	18
vCLI Host Management Commands and Lockdown Mode	19
<b>2 Managing Hosts</b>	<b>21</b>
Stopping, Rebooting, and Examining Hosts	21
Stopping and Rebooting Hosts with ESXCLI	21
Stopping, Rebooting, and Examining Hosts with vicfg-hostops	21
Entering and Exiting Maintenance Mode	22
Entering and Exiting Maintenance Mode with ESXCLI	22
Entering and Exiting Maintenance Mode with vicfg-hostops	23
Backing Up Configuration Information with vicfg-cfgbackup	23
Backup Tasks	23
Backing Up Configuration Data	24
Restoring Configuration Data	24
Using vicfg-cfgbackup from vMA	24
Managing VMkernel Modules	24
Managing Modules with esxcli system module	25
Managing Modules with vicfg-module	25
Using vicfg-authconfig for Active Directory Configuration	25
Updating Hosts	26
<b>3 Managing Files</b>	<b>27</b>
Introduction to Virtual Machine File Management	27
Managing the Virtual Machine File System with vmkfstools	28
Upgrading VMFS3 Volumes to VMFS5	29
Managing VMFS Volumes	29
Managing Duplicate VMFS Datastores	29
Mounting Datastores with Existing Signatures	29

Mounting and Unmounting with ESXCLI	30
Mounting and Unmounting with vicfg-volume	30
Resignaturing VMFS Copies	30
Resignaturing a VMFS Copy with ESXCLI	31
Resignaturing a VMFS Copy with vicfg-volume	31
Reclaiming Unused Storage Space	31
Using vifs to View and Manipulate Files on Remote ESXi Hosts	32
Examples	34
Example Scenario	35

## 4 Managing Storage 37

Introduction to Storage	37
How Virtual Machines Access Storage	38
Datastores	39
Storage Device Naming	39
Device Identifiers	40
Legacy Identifiers	40
Examining LUNs	40
Target and Device Representation	40
Examining LUNs with esxcli storage core	41
Examining LUNs with vicfg-scsidevs	42
Detaching Devices and Removing a LUN	43
Working with Permanent Device Loss	43
Managing Paths	44
Multipathing with Local Storage and FC SANs	44
Listing Path Information	45
Listing Path Information with ESXCLI	45
Listing Path Information with vicfg-mpath	46
Changing the State of a Path	46
Changing Path State with ESXCLI	47
Changing Path State with vicfg-mpath	47
Managing Path Policies	47
Multipathing Considerations	48
Changing Path Policies	49
Changing Path Policies with ESXCLI	49
Changing Path Policies with vicfg-mpath	49
Setting Policy Details for Devices that Use Round Robin	50
Managing NFS/NAS Datastores	50
Capabilities Supported by NFS/NAS	51
Adding and Deleting NAS File Systems	51
Managing NAS File Systems with ESXCLI	51
Managing NAS File Systems with vicfg-nas	52
Monitoring and Managing SAN Storage	52
Monitoring and Managing Virtual SAN Storage	53
To retrieve Virtual SAN information	53
To manage a Virtual SAN cluster	53
To add and remove Virtual SAN storage	54
Monitoring vSphere Flash Read Cache	54
Monitoring and Managing Virtual Volumes	55
Migrating Virtual Machines with svmotion	55
Storage vMotion Uses	56
Storage vMotion Requirements and Limitations	56
Running svmotion in Interactive Mode	56
Running svmotion in Noninteractive Mode	56

Configuring FCoE Adapters	57
Scanning Storage Adapters	58
Retrieving SMART Information	58
<b>5 Managing iSCSI Storage</b>	<b>59</b>
iSCSI Storage Overview	59
Discovery Sessions	60
Discovery Target Names	61
Protecting an iSCSI SAN	61
Protecting Transmitted Data	61
Securing iSCSI Ports	62
Setting iSCSI CHAP	62
Command Syntax for <code>esxcli iscsi</code> and <code>vicfg-iscsi</code>	63
<code>esxcli iscsi</code> Command Syntax	63
Key to <code>esxcli iscsi</code> Short Options	64
<code>vicfg-iscsi</code> Command Syntax	65
iSCSI Storage Setup with <code>ESXCLI</code>	68
Setting Up Software iSCSI with <code>ESXCLI</code>	68
Setting Up Dependent Hardware iSCSI with <code>ESXCLI</code>	70
Setting Up Independent Hardware iSCSI with <code>ESXCLI</code>	72
iSCSI Storage Setup with <code>vicfg-iscsi</code>	73
Setting Up Software iSCSI with <code>vicfg-iscsi</code>	73
Setting Up Dependent Hardware iSCSI with <code>vicfg-iscsi</code>	75
Setting Up Independent Hardware iSCSI with <code>vicfg-iscsi</code>	76
Listing and Setting iSCSI Options	77
Listing iSCSI Options with <code>ESXCLI</code>	77
Setting MTU with <code>ESXCLI</code>	78
Listing and Setting iSCSI Options with <code>vicfg-iscsi</code>	78
Listing and Setting iSCSI Parameters	78
Listing and Setting iSCSI Parameters with <code>ESXCLI</code>	79
Returning Parameters to Default Inheritance	80
Listing and Setting iSCSI Parameters with <code>vicfg-iscsi</code>	80
Returning Parameters to Default Inheritance	81
Enabling iSCSI Authentication	82
Enabling iSCSI Authentication with <code>ESXCLI</code>	82
Enabling iSCSI Authentication with <code>vicfg-iscsi</code>	83
Setting Up Ports for iSCSI Multipathing	83
Managing iSCSI Sessions	84
Introduction to iSCSI Session Management	84
Listing iSCSI Sessions	85
Logging in to iSCSI Sessions	85
Removing iSCSI Sessions	85
<b>6 Managing Third-Party Storage Arrays</b>	<b>87</b>
Managing NMP with <code>esxcli storage nmp</code>	87
Device Management with <code>esxcli storage nmp device</code>	87
<code>esxcli storage nmp device list</code>	88
<code>esxcli storage nmp device set</code>	88
Listing Paths with <code>esxcli storage nmp path</code>	88
Managing Path Selection Policy Plugins with <code>esxcli storage nmp psp</code>	88
Retrieving PSP Information	89
Setting Configuration Parameters for Third-Party Extensions	89
Fixed Path Selection Policy Operations	89
Retrieving the Preferred Path	89
Setting the Preferred Path	90

- Customizing Round Robin Setup 90
      - Retrieving Path Selection Settings 90
      - Specifying Conditions for Path Changes 91
  - Managing SATPs 91
    - Retrieving Information About SATPs 91
    - Adding SATP Rules 91
    - Removing SATP Rules 92
    - Retrieving and Setting SATP Configuration Parameters 93
  - Path Claiming with esxcli storage core claiming 93
    - Using the Reclaim Troubleshooting Command 94
    - Unclaiming Paths or Sets of Paths 94
  - Managing Claim Rules 95
    - Adding Claim Rules 95
    - Converting ESX 3.5 LUN Masks to Claim Rule Format 97
    - Removing Claim Rules 98
    - Listing Claim Rules 98
    - Loading Claim Rules 98
    - Moving Claim Rules 98
    - Running Path Claiming Rules 99
- 7 Managing Users 101**
  - Users in the vSphere Environment 101
  - vicfg-user Command Syntax 101
  - Managing Users with vicfg-user 102
  - Assigning Permissions with ESXCLI 104
- 8 Managing Virtual Machines 105**
  - vmware-cmd Overview 105
    - Connection Options for vmware-cmd 106
    - General Options for vmware-cmd 106
    - Format for Specifying Virtual Machines 106
  - Listing and Registering Virtual Machines 106
  - Retrieving Virtual Machine Attributes 107
  - Managing Virtual Machine Snapshots with vmware-cmd 108
    - Taking Virtual Machine Snapshots 109
    - Reverting and Removing Snapshots 109
  - Powering Virtual Machines On and Off 109
  - Connecting and Disconnecting Virtual Devices 110
  - Working with the AnswerVM API 111
  - Forcibly Stopping Virtual Machines with EXCLI 111
- 9 Managing vSphere Networking 113**
  - Introduction to vSphere Networking 113
    - Networking Using vSphere Standard Switches 114
    - Networking Using vSphere Distributed Switches 115
  - Retrieving Basic Networking Information 115
  - Network Troubleshooting 116
  - Setting Up vSphere Networking with vSphere Standard Switches 117
    - Setting Up Virtual Switches and Associating a Switch with a Network Interface 117
    - Retrieving Information About Virtual Switches 118
      - Retrieving Information about Virtual Switches with ESXCLI 118
      - Retrieving Information about Virtual Switches with vicfg-vswitch 118
  - Adding and Deleting Virtual Switches 119
    - Adding and Deleting Virtual Switches with ESXCLI 119

Adding and Deleting Virtual Switches with vicfg-vswitch	119
Setting Switch Attributes with esxcli network vswitch standard	119
Setting Switch Attributes with vicfg-vswitch	120
Checking, Adding, and Removing Port Groups	120
Managing Port Groups with ESXCLI	120
Managing Port Groups with vicfg-vswitch	120
Managing Uplinks and Port Groups	121
Connecting and Disconnecting Uplink Adapters and Port Groups with ESXCLI	121
Connecting and Disconnecting Uplinks and Port Groups with vicfg-vswitch	121
Setting the Port Group VLAN ID	121
Setting the Port Group VLAN ID with ESXCLI	121
Setting the Port Group VLAN ID with vicfg-vswitch	122
Managing Uplink Adapters	122
Managing Uplink Adapters with esxcli network nic	122
Specifying Multiple Uplinks with ESXCLI	123
Managing Uplink Adapters with vicfg-nics	124
Linking and Unlinking Uplink Adapters with ESXCLI	124
Linking and Unlinking Uplink Adapters with vicfg-vswitch	124
Adding and Modifying VMkernel Network Interfaces	125
Managing VMkernel Network Interfaces with ESXCLI	125
Managing VMkernel Network Interfaces with vicfg-vmknic	126
Setting Up vSphere Networking with vSphere Distributed Switch	128
Managing Standard Networking Services in the vSphere Environment	128
Setting the DNS Configuration	128
Setting the DNS Configuration with ESXCLI	128
Setting the DNS Configuration with vicfg-dns	130
Adding and Starting an NTP Server	131
Managing the IP Gateway	131
Setting Up IPsec	132
Using IPsec with ESXi	132
Managing Security Associations	133
Managing Security Policies	134
Managing the ESXi Firewall	135
Monitoring VXLAN	136
	137

## 10 Monitoring ESXi Hosts 139

Using resxtop for Performance Monitoring	139
Managing Diagnostic Partitions	139
Diagnostic Partition Creation	140
Diagnostic Partition Management	140
Managing Core Dumps	140
Managing Local Core Dumps with ESXCLI	140
Managing Core Dumps with ESXi Dump Collector	141
Managing Core Dumps with vicfg-dumppart	141
Configuring ESXi Syslog Services	142
Managing ESXi SNMP Agents	143
Configuring SNMP Communities	144
Configuring the SNMP Agent to Send Traps	144
Configuring a Trap Destination with ESXCLI	144
Configuring a Trap Destination with vicfg-snmp	145
Configuring the SNMP Agent for Polling	145
Retrieving Hardware Information	146

Index 147



# About This Book

---

The *Getting Started with vSphere Command-Line Interfaces* documentation explains how to use the commands in the VMware vSphere® Command-Line Interface (vCLI) and includes command overviews and examples.

## Intended Audience

This book is for experienced Windows or Linux system administrators who are familiar with vSphere administration tasks and data center operations and know how to use commands in scripts.

## Document Feedback

VMware welcomes your suggestions for improving our documentation. If you have comments, send your feedback to [docfeedback@vmware.com](mailto:docfeedback@vmware.com) or click on the **Send Us Feedback** button in the documentation center.

## Related Documentation

The *vSphere Command-Line Interface Reference*, available in the vSphere Documentation Center, includes reference information for `vicfg-` commands and ESXCLI commands.

*Getting Started with vSphere Command-Line Interfaces* includes information about available CLIs, enabling the ESXi Shell, and installing and running vCLI and DCLI commands.

*Command-Line Management in vSphere 5 and vSphere 6 for Service Console Users* is for customers who currently use the ESX Service Console.

The vSphere SDK for Perl documentation explains how you can use the vSphere SDK for Perl and related utility applications to manage your vSphere environment. The documentation includes an *Installation Guide*, a *Programming Guide*, and a reference to the vSphere SDK for Perl Utility Applications.

Background information for the tasks discussed in this manual is available in the vSphere documentation set. The vSphere documentation consists of the combined vCenter Server and ESXi documentation and includes information about managing storage, networking virtual machines, and more.

## Technical Support and Education Resources

The following sections describe the technical support resources available to you. To access the current version of this book and other books, go to <http://www.pubs.vmware.com>.

### Online and Telephone Support

To use online support to submit technical support requests, view your product and contract information, and register your products, go to <http://www.vmware.com/support>.

Customers with appropriate support contracts should use telephone support for the fastest response on priority 1 issues. Go to [http://www.vmware.com/support/phone\\_support](http://www.vmware.com/support/phone_support).

## Support Offerings

To find out how VMware support offerings can help meet your business needs, go to <http://www.vmware.com/support/services>.

## VMware Professional Services

VMware Education Services courses offer extensive hands-on labs, case study examples, and course materials designed to be used as on-the-job reference tools. Courses are available onsite, in the classroom, and live online. For onsite pilot programs and implementation best practices, VMware Consulting Services provides offerings to help you assess, plan, build, and manage your virtual environment. To access information about education classes, certification programs, and consulting services, go to <http://www.vmware.com/services>.

# vSphere CLI Command Overviews

This chapter introduces the command set, presents supported commands for different versions of vSphere, lists connection options, and discusses vCLI and lockdown mode.

This chapter includes the following topics:

- [“Introduction”](#) on page 11
- [“List of Available Host Management Commands”](#) on page 12
- [“Targets and Protocols for vCLI Host Management Commands”](#) on page 14
- [“Commands with an esxcfg Prefix”](#) on page 16
- [“ESXCLI Overview”](#) on page 16
- [“Connection Options for vCLI Host Management Commands”](#) on page 18
- [“Connection Options for DCLI Commands”](#) on page 18
- [“vCLI Host Management Commands and Lockdown Mode”](#) on page 19

## Introduction

The commands in the vSphere CLI package allow you to perform vSphere configuration tasks using commands from vCLI package installed on supported platforms, or using commands from vMA. The package consists of several command sets.

**Table 1-1.** Components of the vSphere CLI Command Set

vCLI Commands	Description
ESXCLI commands	Manage many aspects of an ESXi host. You can run ESXCLI commands remotely or in the ESXi Shell. You can also run ESXCLI commands from the vSphere PowerCLI prompt by using the <code>Get-EsxCli</code> cmdlet.
vicfg- commands	Set of commands for many aspects of host management Eventually, these commands will be replaced by ESXCLI commands. A set of esxcfg- commands that precisely mirrors the vicfg- commands is also included in the vCLI package. f
Other commands (vmware-cmd, vifs, vmkfstools)	Commands implemented in Perl that do not have a vicfg- prefix. These commands are scheduled to be deprecated or replaced by ESXCLI commands.
DCLI commands	Manage VMware SDDC services. DCLI is a CLI client to the vCloud Suite SDK interface for managing VMware SDDC services. A DCLI command talks to a vCloud Suite API endpoint to locate relevant information, and then executes the command and displays result to the user.

You can install the vSphere CLI command set on a supported Linux or Windows system. See *Getting Started with vSphere Command-Line Interfaces*. You can also deploy the vSphere Management Assistant (vMA) to an ESXi system of your choice.

After installation, run vCLI commands from the Linux or Windows system or from vMA.

- Manage ESXi hosts with other vCLI commands by specifying connection options such as the target host, user, and password or a configuration file. See “[Connection Options for vCLI Host Management Commands](#)” on page 18.
- Manage vCenter services with DCLI commands by specifying a target vCenter Server system and authentication options. See *Getting Started with vSphere Command-Line Interfaces* for a list of connection options.

## Documentation

*Getting Started with vSphere Command-Line Interfaces* includes information about available CLIs, enabling the ESXi Shell, and installing and running vCLI commands. An appendix supplies the namespace and command hierarchies for ESXCLI.

Reference information for vCLI and DCLI commands is available on the vCLI documentation page <http://www.vmware.com/support/developer/vcli/> and in the vSphere Documentation Center for the product version that you are using.

- *vSphere Command-Line Interface Reference* is a reference to `vicfg-` and related vCLI commands and includes reference information for ESXCLI commands. All reference information is generated from the help.
- A reference to `esxstop` and `resxstop` is included in the *Resource Management* documentation.
- The *DCLI Reference* is included separately from the *vSphere Command-Line Interface Reference*. All reference information is generated from the help.

## Command-Line Help

Available command-line help differs for the different command sets.

Command set	Available Command-Line Help
vicfg- commands	Run <code>&lt;vicfg-cmd&gt; --help</code> for an overview of each options. Run Pod2Html with a vicfg- command as input and pipe the output to a file for more detailed help information. <code>pod2html vicfg-authconfig.pl &gt; vicfg-authconfig.html</code> This output corresponds to the information available in the <i>vSphere Command-Line Interface Reference</i> .
ESXCLI commands	Run <code>--help</code> at any level of the hierarchy for information about both commands and namespaces available from that level.
DCLI commands	Run <code>--help</code> for any command or namespace to display the input options, whether the option is required, and the input option type. For namespaces, <code>--help</code> displays all available child namespaces and commands. Run <code>dcli --help</code> to display usage information for DCLI.

## List of Available Host Management Commands

Table 1-2 lists vCLI host management commands in alphabetical order and the corresponding ESXCLI command if available. For ESXCLI, new commands and namespaces are added with each release. See the *Release Notes* for the corresponding release for information.

Functionality of DCLI command set that is being added in vSphere 6.0 is different from these commands. They are not included in the table.

**Table 1-2.** vCLI and ESXCLI Commands

vCLI 4.1 Command	vCLI 5.1 and later Command	Comment
esxcli	esxcli (new syntax)	All vCLI 4.1 commands have been renamed. Significant additions have been made to ESXCLI. Many tasks previously performed with a vicfg- command is now performed with ESXCLI.
resxtop	resxtop (No ESXCLI equivalent) Supported only on Linux.	Monitors in real time how ESXi hosts use resources. Runs in interactive or batch mode. See <a href="#">“Using resxtop for Performance Monitoring”</a> on page 139. See the <i>vSphere Resource Management</i> documentation for a detailed reference.
svmotion	svmotion (No ESXCLI equivalent) Must run against a vCenter Server system.	Moves a virtual machine’s configuration file, and, optionally, its disks, while the virtual machine is running. See <a href="#">“Migrating Virtual Machines with svmotion”</a> on page 55.
vicfg-advcfg	esxcli system settings advanced	Performs advanced configuration. The advanced settings are a set of VMkernel options. These options are typically in place for specific workarounds or debugging. Use this command as instructed by VMware.
vicfg-authconfig	vicfg-authconfig (No ESXCLI equivalent).	Remotely configures Active Directory settings for an ESXi host. See <a href="#">“Using vicfg-authconfig for Active Directory Configuration”</a> on page 25.
vicfg-cfgbackup	vicfg-cfgbackup (No ESXCLI equivalent), Cannot run against a vCenter Server system.	Backs up the configuration data of an ESXi system and restores previously saved configuration data. See <a href="#">“Backing Up Configuration Information with vicfg-cfgbackup”</a> on page 23.
vicfg-dns	esxcli network ip dns	Specifies an ESXi host’s DNS (Domain Name Server) configuration. See <a href="#">“Setting the DNS Configuration”</a> on page 128.
vicfg-dumppart	esxcli system coredump	Sets both the partition (esxcli system coredump partition) and the network (esxcli system coredump network) to use for core dumps. Use this command to set up ESXi Dump Collector. <a href="#">“Managing Diagnostic Partitions”</a> on page 139.
vicfg-hostops	esxcli system maintenancemode esxcli system shutdown	Manages hosts. <a href="#">“Stopping, Rebooting, and Examining Hosts”</a> on page 21. <a href="#">“Entering and Exiting Maintenance Mode”</a> on page 22.
vicfg-ipsec	esxcli network ip ipsec	Sets up IPsec (Internet Protocol Security), which secures IP communications coming from and arriving at ESXi hosts. ESXi hosts support IPsec using IPv6. See <a href="#">“Setting Up IPsec”</a> on page 132.
vicfg-iscsi	esxcli iscsi	Manages hardware and software iSCSI storage. See <a href="#">“Managing iSCSI Storage”</a> on page 59.
vicfg-module	esxcli system module	Enables VMkernel options. Use this command with the options listed in this document, or as instructed by VMware. See <a href="#">“Managing VMkernel Modules”</a> on page 24.
vicfg-mpath vicfg-mpath35	esxcli storage core path	Configures storage arrays. <a href="#">“Managing Paths”</a> on page 44.
vicfg-nas	esxcli storage nfs	Manages NAS/NFS filesystems. See <a href="#">“Managing NFS/NAS Datastores”</a> on page 50.
vicfg-nics	esxcli network nic	Manages the ESXi host’s uplink adapters. See <a href="#">“Managing Uplink Adapters”</a> on page 122.
vicfg-ntp	vicfg-ntp (No ESXCLI equivalent)	Defines the NTP (Network Time Protocol) server. See <a href="#">“Adding and Starting an NTP Server”</a> on page 131.
vicfg-rescan	esxcli storage core adapter rescan	Rescans the storage configuration. See <a href="#">“Scanning Storage Adapters”</a> on page 58.
vicfg-route	esxcli network ip route)	Manages the ESXi host’s route entry. See <a href="#">“Managing the IP Gateway”</a> on page 131.

**Table 1-2.** vCLI and ESXCLI Commands (Continued)

vCLI 4.1 Command	vCLI 5.1 and later Command	Comment
vicfg-scsidevs	esxcli storage core device	Finds and examines available LUNs. See <a href="#">“Examining LUNs”</a> on page 40.
vicfg-snmpp	esxcli system snmp	Manages the SNMP agent. <a href="#">“Managing ESXi SNMP Agents”</a> on page 143. Using SNMP in a vSphere environment is discussed in detail in the <i>vSphere Monitoring and Performance</i> documentation. New options added in vCLI 5.0. Expanded SNMP support added in vCLI 5.1.
vicfg-syslog	esxcli system syslog	Specifies log settings for ESXi hosts including local storage policies and server and port information for network logging. See <a href="#">“Configuring ESXi Syslog Services”</a> on page 142. The <i>vCenter Server and Host Management</i> documentation explains how to set up system logs using the vSphere Web Client.
vicfg-user	vicfg-user (No ESXCLI equivalent)	Creates, modifies, deletes, and lists local direct access users and groups of users. See <a href="#">“Managing Users”</a> on page 101. The <i>vSphere Security</i> documentation discusses security implications of user management and custom roles.
vicfg-vmknic	esxcli network ip interface	Adds, deletes, and modifies VMkernel network interfaces. See <a href="#">“Adding and Modifying VMkernel Network Interfaces”</a> on page 125.
vicfg-volume	esxcli storage filesystem	Supports resignaturing the copy of a VMFS volume, and mounting and unmounting the copy. See <a href="#">“Managing Duplicate VMFS Datastores”</a> on page 29.
vicfg-vswitch	esxcli network vswitch	Adds or removes virtual switches or modifies virtual switch settings. See <a href="#">“Setting Up Virtual Switches and Associating a Switch with a Network Interface”</a> on page 117.
vifs	vifs (No ESXCLI equivalent)	Performs file system operations such as retrieving and uploading files on the ESXi system. See <a href="#">“Managing the Virtual Machine File System with vmkfstools”</a> on page 28.
vihostupdate	esxcli software vib	Updates legacy ESXi hosts to a different version of the same major release. You cannot run <code>vihostupdate</code> against ESXi 5.0 and later hosts. See <a href="#">“Managing VMkernel Modules”</a> on page 24.
vmkfstools	vmkfstools (No ESXCLI equivalent)	Creates and manipulates virtual disks, file systems, logical volumes, and physical storage devices on an ESXi host. See <a href="#">“Managing the Virtual Machine File System with vmkfstools”</a> on page 28.
vmware-cmd	vmware-cmd (No ESXCLI equivalent)	Performs virtual machine operations remotely. This includes, for example, creating a snapshot, powering the virtual machine on or off, and getting information about the virtual machine. See <a href="#">“Managing Virtual Machines”</a> on page 105.

## Targets and Protocols for vCLI Host Management Commands

Most vCLI commands are used to manage or retrieve information about one or more ESXi hosts. They can target an ESXi host or a vCenter Server system. When you target a vCenter Server system, you can use `--vihost` to specify the ESXi host to run the command against. The only exception is `svmotion`, which you can run against vCenter Server systems, but not against ESXi systems.

The following commands must have an ESXi system, not a vCenter Server system as a target.

- `vifs`
- `vicfg-user`
- `vicfg-cfgbackup`
- `vihostupdate`
- `vmkfstools`

The `resxtop` command requires an HTTPS connection. All other commands support HTTP and HTTPS.

## Supported Platforms for vCLI Commands

You cannot run the `vihostupdate` command against an ESXi 5.0 or later system.

You cannot run `vicfg-syslog --setserver` or `vicfg-syslog --setport` with an ESXi 5.0 or later target.

Table 1-3 lists platform support for the different vCLI commands.

**Table 1-3.** Platform Support for vCLI 5.x Commands

Command	ESXi 5.x and 6.0	VC 5.x and 6.0	ESXi 4.x	ESX 4.x	VC 4.x
DCLI	No	No	No	No	No
esxcli	Yes	Yes	Yes	Yes	No
resxtop	Yes (from Linux)	Yes (from Linux)	Yes (from Linux)	Yes (from Linux)	Yes (from Linux)
svmotion	No	Yes	No	No	Yes
vicfg-advcfg	Yes	Yes	Yes	Yes	Yes
vicfg-authconfig	Yes	Yes	Yes	Yes	Yes
vicfg-cfgbackup	Yes	No	Yes	No	No
vicfg-dns	Yes	Yes	Yes	Yes	Yes
vicfg-dumppart	Yes	Yes	Yes	Yes	Yes
vicfg-hostops	Yes	Yes	Yes	Yes	Yes
vicfg-ipsec	Yes	No	Yes	Yes	No
vicfg-iscsi	Yes	Yes	Yes	Yes	Yes
vicfg-module	Yes	Yes	Yes	Yes	Yes
vicfg-mpath	Yes	Yes	Yes	Yes	Yes
vicfg-nas	Yes	Yes	Yes	Yes	Yes
vicfg-nics	Yes	Yes	Yes	Yes	Yes
vicfg-ntp	Yes	Yes	Yes	Yes	Yes
vicfg-rescan	Yes	Yes	Yes	Yes	Yes
vicfg-route	Yes	Yes	Yes	Yes	Yes
vicfg-scsidevs	Yes	Yes	Yes	Yes	Yes
vicfg-snmpp	Yes	No	Yes	Yes	No
vicfg-syslog	No	No for 5.0 target	Yes	No	Yes
vicfg-user	Yes	No	Yes	Yes	No
vicfg-vmknic	Yes	Yes	Yes	Yes	Yes
vicfg-volume	Yes	Yes	Yes	Yes	Yes
vicfg-vswitch	Yes	Yes	Yes	Yes	Yes
vifs	Yes	No	Yes	Yes	No
vihostupdate	Use <code>esxcli software vib</code> instead.		Yes	Yes	No
vmkfstools	Yes	No	Yes	Yes	No
vmware-cmd	Yes	Yes	Yes	Yes	Yes
vicfg-mpath35	No	No	No	No	No
vihostupdate35	No	No	No	No	No

## Commands with an esxcfg Prefix

To facilitate easy migration of shell scripts that use `esxcfg-` commands, the vCLI package includes a copy of each `vicfg-` command that uses an `esxcfg-` prefix.

---

**IMPORTANT** VMware recommends that you use ESXCLI or the vCLI commands with the `vicfg` prefix. Commands with the `esxcfg` prefix are available mainly for compatibility reasons and are now obsolete.

vCLI `esxcfg-` commands are equivalent to `vicfg-` commands, but not completely equivalent to the deprecated `esxcfg-` service console commands.

---

[Table 1-4](#) lists all vCLI `vicfg-` commands for which a vCLI command with an `esxcfg` prefix is available.

**Table 1-4.** Commands with an `esxcfg` Prefix

Command with <code>vicfg</code> prefix	Command with <code>esxcfg</code> prefix
<code>vicfg-advcfg</code>	<code>esxcfg-advcfg</code>
<code>vicfg-cfgbackup</code>	<code>esxcfg-cfgbackup</code>
<code>vicfg-dns</code>	<code>esxcfg-dns</code>
<code>vicfg-dumppart</code>	<code>esxcfg-dumppart</code>
<code>vicfg-module</code>	<code>esxcfg-module</code>
<code>vicfg-mpath</code>	<code>esxcfg-mpath</code>
<code>vicfg-nas</code>	<code>esxcfg-nas</code>
<code>vicfg-nics</code>	<code>esxcfg-nics</code>
<code>vicfg-ntp</code>	<code>esxcfg-ntp</code>
<code>vicfg-rescan</code>	<code>esxcfg-rescan</code>
<code>vicfg-route</code>	<code>esxcfg-route</code>
<code>vicfg-scsidevs</code>	<code>esxcfg-scsidevs</code>
<code>vicfg-snmp</code>	<code>esxcfg-snmp</code>
<code>vicfg-syslog</code>	<code>esxcfg-syslog</code>
<code>vicfg-vmknic</code>	<code>esxcfg-vmknic</code>
<code>vicfg-volume</code>	<code>esxcfg-volume</code>
<code>vicfg-vswitch</code>	<code>esxcfg-vswitch</code>

## ESXCLI Overview

This section gives an overview of ESXCLI commands and how to use them. See *Getting Started with vSphere Command-Line Interfaces* for details.

### ESXCLI Commands Available on Different ESXi Hosts

When you run an ESXCLI vCLI command, you must know the commands supported on the target host. For example, if you run commands against ESXi 5.x hosts, ESXCLI 5.x commands are supported. If you run commands against ESXi 6.0 hosts, ESXCLI 6.0 commands are supported.

Some commands or command outputs are determined by the host type. In addition, VMware partners might develop custom ESXCLI commands that you can run on hosts where the partner VIB has been installed.

Run `esxcli --server <target> --help` for a list of namespaces supported on the target. You can drill down into the namespaces for additional help.

---

**IMPORTANT** ESXCLI on ESX 4.x hosts does not support targeting a vCenter Server system. You can therefore not run ESXCLI commands with `--server` pointing to a vCenter Server system even if you install vCLI 5.0.

---



## Trust Relationship Requirement for ESXCLI Commands

Starting with vSphere 6.0, ESXCLI checks whether a trust relationship exists between the machine where you run the ESXCLI command and the ESXi host. An error results if the trust relationship does not exist.

To establish the trust relationship, you have these options.

### Downloading and Installing the vCenter Server Certificate

You can download the vCenter Server root certificate using a Web browser and add it to the trusted certificates on the machine where you plan on running ESXCLI commands.

#### To download the certificate

- 1 Type the URL of the vCenter Server system or vCenter Server Virtual Appliance into a Web Browser.
- 2 Click the **Download trusted root certificates** link.
- 3 Change the extension of the downloaded file to `.zip`. (The file is a ZIP file of all certificates in the TRUSTED\_ROOTS store).
- 4 Extract the ZIP file.

The result is a `certs` folder. The folder includes files with the extension `.0`, `.1`, and so on, which are certificates, and files with the extension `.r0`, `r1`, and so on which are CRL files associated with the certificates.

- 5 Add the trusted root certificates to the list of trusted roots. The process differs depending on the platform you are on.

You can now run ESXCLI commands against any host that is managed by the trusted vCenter Server without supplying additional information if you specify the vCenter Server in the `--server` option and the ESXi host in the `--vhost` option.

### Using the `--cacertsfile` Option

Using a certificate to establish the trust relationship is the most secure option. You can specify the certificate with the `--cacertsfile` parameter or the `VI_CACERTFILE` variable.

### Using the `--thumbprint` Option

You can supply the thumbprint for the target server (ESXi host or vCenter Server system) in the `--thumbprint` parameter (`VI_THUMBPRINT` variable).

When you run a command, ESXCLI checks first whether a certificate file is available. If not, ESXCLI checks whether a thumbprint of the target server is available. If not, an error like the following results:

```
Connect to sof-40583-srv failed. Server SHA-1 thumbprint:
5D:01:06:63:55:9D:DF:FE:38:81:6E:2C:FA:71:BC:Usin63:82:C5:16:51 (not trusted).
```

You can run the command with the thumbprint to establish the trust relationship, or add the thumbprint to the `VI_THUMBPRINT` variable. For example, using the thumbprint of the ESXi host above, you can run the following command:

```
esxcli --server myESXi --username user1 --password 'my_password' --thumbprint
5D:01:06:63:55:9D:DF:FE:38:81:6E:2C:FA:71:BC:63:82:C5:16:51 storage nfs list
```

### Using the Credential Store

Your vCLI installation includes a credential store. You can manage the credential store with the `credstore-admin` utility application, which is located in the `/Perl/apps/general` directory inside the VMware vSphere CLI directory.

---

**IMPORTANT** Updating the credential store is a two-step process. First you add the user and password for the server, and then you add the thumbprint for the server.

---

**To establish trust for a user with the credential store**

- 1 Add the user and password for the target ESXi host to the local credential store.  

```
credstore_admin.pl add --server <esxi_HOSTNAME_OR_IP> --username <user> --password <pwd>
```
- 2 Add the thumbprint for the target ESXi host. This thumbprint was returned in the error when you attempted to connect to the host.  

```
credstore_admin.pl add --server <esxi_HOSTNAME_OR_IP> --thumbprint <thumbprint>
```
- 3 If you are using a non-default credential store file, you have to pass it in with the `--credstore` option. Otherwise, this user will be able to access the host without authentication going forward.

**Using ESXCLI Output**

Many ESXCLI commands generate output you might want to use in your application. You can run `esxcli` with the `--formatter` dispatcher option and send the resulting output as input to a parser.

The `--formatter` options supports three values, `csv`, `xml`, and `keyvalue` and is used before any namespace.

```
esxcli --formatter=csv storage filesystem list
```

Lists all file system information in CSV format.

You can pipe the output to a file.

```
esxcli --formatter=keyvalue storage filesystem list > myfilesystemlist.txt
```

---

**IMPORTANT** Always use a formatter for consistent output.

---

**Connection Options for vCLI Host Management Commands**

You can run host management commands such as ESXCLI commands, `vicfg-` commands, and other commands with several different connection options. You can target hosts directly or target a vCenter Server system and specify the host you want to manage. If you are targeting a vCenter Server system, specify the Platform Services Controller, which includes the vCenter Single Sign-On service, for best security.

---

**IMPORTANT** For connections to ESXi 6.0 hosts, vCLI supports both the IPv4 protocol and the IPv6 protocol. For earlier versions, vCLI supports only IPv4. In all cases, you can configure IPv6 on the target host with several of the networking commands.

---

See the *Getting Started with vSphere Command-Line Interfaces* documentation for a complete list and examples.

**Connection Options for DCLI Commands**

DCLI is a CLI client to the vCloud Suite SDK interface for managing VMware SDDC services. A DCLI command talks to a vCloud Suite SDK endpoint to get the vCloud Suite SDK command information, executes the command, and displays result to the user.

You can run DCLI commands locally or from an administration server.

- Run DCLI on the Linux shell of a vCenter Server Virtual Appliance.
- Install vCLI on a supported Windows or Linux system and target a vCenter Server windows installation or a vCenter Server Virtual Appliance. You have to provide endpoint information to successfully run commands.

DCLI commands support other connection options than other commands in the command set.

See the *Getting Started with vSphere Command-Line Interfaces* documentation for a complete list and examples.

## vCLI Host Management Commands and Lockdown Mode

For additional security, an administrator can place one or more hosts managed by a vCenter Server system in lockdown mode. Lockdown mode affects login privileges for the ESXi host. See the *vSphere Security* document in the vSphere 6.0 Documentation Center for a detailed discussion of normal lockdown mode and strict lockdown mode, and of how to enable and disable them.

To make changes to ESXi systems in lockdown mode, you must go through a vCenter Server system that manages the ESXi system as the user `vpxuser` and include both the `--server` and the `--vihost` parameter. .

```
esxcli --server MyVC --vihost MyESXi storage filesystem list
```

The command prompts for the vCenter Server system user name and password.

The following commands cannot run against vCenter Server systems and are therefore not available in lockdown mode:

- `vifs`
- `vicfg-user`
- `vicfg-cfgbackup`
- `vihostupdate`
- `vmkfstools`

If you have problems running a command on an ESXi host directly (without specifying a vCenter Server target), check whether lockdown mode is enabled on that host.



# Managing Hosts

---

Host management commands can stop and reboot ESXi hosts, back up configuration information, and manage host updates. You can also use a host management command to make your host join an Active Directory domain or exit from a domain.

The chapter includes the following topics:

- [“Stopping, Rebooting, and Examining Hosts”](#) on page 21
- [“Entering and Exiting Maintenance Mode”](#) on page 22
- [“Backing Up Configuration Information with vicfg-cfgbackup”](#) on page 23
- [“Managing VMkernel Modules”](#) on page 24
- [“Using vicfg-authconfig for Active Directory Configuration”](#) on page 25
- [“Updating Hosts”](#) on page 26

For information on updating ESXi 5.0 hosts with the `esxcli software` command and on changing the host acceptance level to match the level of a VIB that you might want to use for an update, see the *vSphere Upgrade* documentation in the vSphere 5.0 Documentation Center.

## Stopping, Rebooting, and Examining Hosts

You can stop, reboot, and examine hosts with ESXCLI or with `vicfg-hostops`.

### Stopping and Rebooting Hosts with ESXCLI

You can shut down or reboot an ESXi host using the vSphere Web Client or vCLI commands (ESXCLI or `vicfg-hostops`).

Shutting down a managed host disconnects it from the vCenter Server system, but does not remove the host from the inventory. You can shut down a single host or all hosts in a data center or cluster. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

To shut down a host, run `esxcli system shutdown poweroff`. You must specify the `--reason` option and supply a reason for the shutdown. A `--delay` option allows you to specify a delay interval, in seconds.

To reboot a host, run `system shutdown reboot`. You must specify the `--reason` option and supply a reason for the shutdown. A `--delay` option allows you to specify a delay interval, in seconds.

### Stopping, Rebooting, and Examining Hosts with vicfg-hostops

You can shut down or reboot an ESXi host using the vSphere Web Client, or ESXCLI or the `vicfg-hostops` vCLI command.

Shutting down a managed host disconnects it from the vCenter Server system, but does not remove the host from the inventory. You can shut down a single host or all hosts in a data center or cluster. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- **Single host.** Run `vicfg-hostops` with `--operation shutdown`.
  - If the host is in maintenance mode, run the command without the `--force` option.
 

```
vicfg-hostops <conn_options> --operation shutdown
```
  - If the host is not in maintenance mode, use `--force` to shut down the host and all running virtual machines.
 

```
vicfg-hostops <conn_options> --operation shutdown --force
```
- **All hosts in data center or cluster.** To shut down all hosts in a cluster or data center, specify `--cluster` or `--datacenter`.
 

```
vicfg-hostops <conn_options> --operation shutdown --cluster <my_cluster>
vicfg-hostops <conn_options> --operation shutdown --datacenter <my_datacenter>
```

You can reboot a single host or all hosts in a data center or cluster.

- **Single host.** Run `vicfg-hostops` with `--operation reboot`.
  - If the host is in maintenance mode, run the command without the `--force` option.
 

```
vicfg-hostops <conn_options> --operation reboot
```
  - If the host is not in maintenance mode, use `--force` to shut down the host and all running virtual machines.
 

```
vicfg-hostops <conn_options> --operation reboot --force
```
- **All hosts in data center or cluster.** You can specify `--cluster` or `--datacenter` to reboot all hosts in a cluster or data center.
 

```
vicfg-hostops <conn_options> --operation reboot --cluster <my_cluster>
vicfg-hostops <conn_options> --operation reboot --datacenter <my_datacenter>
```

You can display information about a host by running `vicfg-hostops` with `--operation info`.

```
vicfg-hostops <conn_options> --operation info
```

The command returns the host name, manufacturer, model, processor type, CPU cores, memory capacity, and boot time. The command also returns whether vMotion is enabled and whether the host is in maintenance mode.

## Entering and Exiting Maintenance Mode

You can instruct your host to enter or exit maintenance mode with ESXCLI or with `vicfg-hostops`.

### Entering and Exiting Maintenance Mode with ESXCLI

You place a host in maintenance mode to service it, for example, to install more memory. A host enters or leaves maintenance mode only as the result of a user request.

`esxcli system maintenanceMode set` allows you to enable or disable maintenance mode.

When you run the `vicfg-hostops` vCLI command, you can specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

#### To enter and exit maintenance mode

- 1 Run `esxcli <conn_options> system maintenanceMode set --enable true` to enter maintenance mode.

After all virtual machines on the host have been suspended or migrated, the host enters maintenance mode. You cannot deploy or power on a virtual machine on hosts in maintenance mode.

- 2 Run `esxcli <conn_options> system maintenanceMode set --enable false` to have a host existing maintenance mode.

If you attempt to exit maintenance mode when the host is no longer in maintenance mode, an error informs you that maintenance mode is already disabled.

## Entering and Exiting Maintenance Mode with `vicfg-hostops`

You place a host in maintenance mode to service it, for example, to install more memory. A host enters or leaves maintenance mode only as the result of a user request.

`vicfg-hostops` suspends virtual machines by default, or powers off the virtual machine if you run `vicfg-hostops --action poweroff`.

---

**NOTE** `vicfg-hostops` does not work with VMware DRS. Virtual machines are always suspended.

---

The host is in a state of Entering Maintenance Mode until all running virtual machines are suspended or migrated. When a host is entering maintenance mode, you cannot power on virtual machines on it or migrate virtual machines to it.

When you run the `vicfg-hostops` vCLI command, you can specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

### To enter maintenance mode

- 1 Run `vicfg-hostops <conn_options> --operation enter` to enter maintenance mode.
- 2 Run `vicfg-hostops <conn_options> --operation info` to check whether the host is in maintenance mode or in the Entering Maintenance Mode state.

After all virtual machines on the host have been suspended or migrated, the host enters maintenance mode. You cannot deploy or power on a virtual machine on hosts in maintenance mode.

You can put all hosts in a cluster or data center in maintenance mode by using the `--cluster` or `--datacenter` option. Do not use those options unless suspending all virtual machines in that cluster or data center is no problem.

You can later run `vicfg-hostops <conn_options> --operation exit` to exit maintenance mode.

## Backing Up Configuration Information with `vicfg-cfgbackup`

After you configure an ESXi host, you can back up the host configuration data. Always back up your host configuration after you change the configuration or upgrade the ESXi image.

---

**IMPORTANT** The `vicfg-cfgbackup` command is available only for ESXi hosts. The command is not available through a vCenter Server system connection. No equivalent ESXCLI command is supported.

---

### Backup Tasks

During a configuration backup, the serial number is backed up with the configuration. The number is restored when you restore the configuration. The number is not preserved when you run the Recovery CD (ESXi Embedded) or perform a repair operation (ESXi Installable).

You can back up and restore configuration information as follows.

- 1 Back up the configuration by using the `vicfg-cfgbackup` command.
- 2 Run the Recovery CD or repair operation
- 3 Restore the configuration by using the `vicfg-cfgbackup` command.

When you restore a configuration, you must make sure that all virtual machines on the host are stopped.

## Backing Up Configuration Data

You can back up configuration data by running `vicfg-cfgbackup` with the `-s` option.

```
vicfg-cfgbackup <conn_options> -s /tmp/ESXi_181842_backup.txt
```

For the backup filename, include the number of the build that is running on the host that you are backing up. If you are running vCLI on vMA, the backup file is saved locally on vMA. Backup files can safely be stored locally because virtual appliances are stored in the `/vmfs/volumes/<datastore>` directory on the host, which is separate from the ESXi image and configuration files.

## Restoring Configuration Data

If you have created a backup, you can later restore ESXi configuration data. When you restore configuration data, the number of the build running on the host must be the same as the number of the build that was running when you created the backup file. To override this requirement, include the `-f` (force) option.

### To restore ESXi configuration data

- 1 Power off all virtual machines that are running on the host that you want to restore.
- 2 Log in to a host on which vCLI is installed, or log in to vMA.
- 3 Run `vicfg-cfgbackup` with the `-l` flag to load the host configuration from the specified backup file. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.
  - If you run the following command, you are prompted for confirmation.
 

```
vicfg-cfgbackup <conn_options> -l /tmp/ESXi_181842_backup.tgz
```
  - If you run the following command, you are not prompted for confirmation.
 

```
vicfg-cfgbackup <conn_options> -l /tmp/ESXi_181842_backup.tgz -q
```

To restore the host to factory settings, run `vicfg-cfgbackup` with the `-r` option:

```
vicfg-cfgbackup <conn_options> -r
```

## Using vicfg-cfgbackup from vMA

To back up a host configuration, you can run `vicfg-cfgbackup` from a vMA instance. The vMA instance can run on the target host (the host that you are backing up or restoring), or on a remote host.

To restore a host configuration, you must run `vicfg-cfgbackup` from a vMA instance running on a remote host. The host must be in maintenance mode, which means all virtual machines (including vMA) must be suspended on the target host.

For example, a backup operation for two ESXi hosts (host1 and host2) with vMA deployed on both hosts works as follows:

- To back up one of the host’s configuration (host1 or host2), run `vicfg-cfgbackup` from the vMA appliance running on either host1 or host2. Use the `--server` option to specify the host for which you want backup information. The information is stored on vMA.
- To restore the host1 configuration, run `vicfg-cfgbackup` from the vMA appliance running on host2. Use the `--server` option to point to host1 to restore the configuration to that host.
- To restore the host2 configuration, run `vicfg-cfgbackup` from the vMA appliance running on host1. Use the `--server` option to point to host2 to restore the configuration to that host.

## Managing VMkernel Modules

The `esxcli system module` and `vicfg-module` commands support setting and retrieving VMkernel module options.



`vicfg-module` and `esxcli system module` commands are implementations of the deprecated `esxcfg-module` service console command. The two commands support most of the options `esxcfg-module` supports. `vicfg-module` and `esxcli system module` are commonly used when VMware Technical Support, a Knowledge Base article, or VMware documentation instruct you to do so.

## Managing Modules with `esxcli system module`

Not all VMkernel modules have settable module options. The following example illustrates how to examine and enable a VMkernel module. Specify one of the connection options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

### To examine, enable, and set a VMkernel modules

- 1 List information about the module.

```
esxcli <conn_options> system module list -module=module_name
```

The system returns the name, type, value, and description of the module.

- 2 (Optional) List all enabled or loaded modules.

```
esxcli <conn_options> system module list --enabled=true
esxcli <conn_options> system module list --loaded=true
```

- 3 Enable the model.

```
esxcli <conn_options> system module set --module=module_name --enabled=true
```

- 4 Set the parameter.

```
esxcli system module parameters set --module module_name
--parameter-string="parameter_string"
```

- 5 Verify that the module is configured.

```
esxcli <conn_options> system module parameters list --module=module_name
```

## Managing Modules with `vicfg-module`

Not all VMkernel modules have settable module options. The following example illustrates how to examine and enable a VMkernel modules. Specify one of the connection options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

### To examine and set a VMkernel modules

- 1 Run `vicfg-module --list` to list the modules on the host.

```
vicfg-module <conn_options> --list
```

- 2 Run `vicfg-module --set-options` with connection options, the option string to be passed to a module, and the module name. For example:

```
vicfg-module <conn_options> --set-options '<parameter_name>=<value>' <module_name>
```

To retrieve the option string that is configured to be passed to a module when the module is loaded, run `vicfg-module --get-options`. This string is not necessarily the option string currently in use by the module.

```
vicfg-module <conn_options> --get-options module_name
```

Verifies that a module is configured.

## Using `vicfg-authconfig` for Active Directory Configuration

ESXi can be integrated with Active Directory. Active Directory provides authentication for all local services and for remote access through the vSphere Web Services SDK, vSphere Web Client, PowerCLI, and vSphere CLI. You can configure Active Directory settings with the vSphere Web Client, as discussed in the *vCenter Server and Host Management* documentation, or use `vicfg-authconfig`.

`vicfg-authconfig` allows you to remotely configure Active Directory settings on ESXi hosts. You can list supported and active authentication mechanisms, list the current domain, and join or part from an Active Directory domain. Before you run the command on an ESXi host, you must prepare the host.

---

**IMPORTANT** All hosts that join Active Directory must also be managed by an NTP Server to avoid issues with clock skews and Kerberos tickets.

---

### To prepare ESXi hosts for Active Directory Integration

- 1 Make sure the ESXi system and the Active Directory server are using the same time zone by configuring ESXi and AD to use same NTP server.

The ESXi system's time zone is always set to UTC.

- 2 Configure the ESXi system's DNS to be in the Active Directory domain.

You can run `vicfg-authconfig` to add the host to the domain. A user who runs `vicfg-authconfig` to configure Active Directory settings must have the appropriate Active Directory permissions, and must have administrative privileges on the ESXi host. You can run the command directly against the host or against a vCenter Server system, specifying the host with `--vihost`.

### To set up Active Directory

- 1 Install the ESXi host, as explained in the *vSphere Installation and Setup* documentation.
- 2 Install Windows Active Directory on a Windows Server that runs Windows 2000, Windows 2003, or Windows 2008. See the Microsoft Web site for instructions and best practices.
- 3 Synchronize time between the ESXi system and Windows Active Directory (AD).
- 4 Test that the Windows AD Server can ping the ESXi host by using the host name.

```
ping <ESX_hostname>
```

- 5 Run `vicfg-authconfig` to add the host to the Active Directory domain.

```
vicfg-authconfig --server=<ESXi Server IP Address>
--username=<ESXi Server Admin Username>
--password=<ESXi Server Admin User's Password>
--authscheme AD --joindomain <AD Domain Name>
--adusername=<Active Directory Administrator User Name>
--adpassword=<Active Directory Administrator User's Password>
```

The system prompts for user names and passwords if you do not specify them on the command line. Passwords are not echoed to the screen.

- 6 Check that a `Successfully Joined <Domain Name>` message appears.
- 7 Verify the ESXi host is in the intended Windows AD domain.

```
vicfg-authconfig --server XXX.XXX.XXX.XXX --authscheme AD -c
```

You are prompted for a user name and password for the ESXi system.

## Updating Hosts

When you add custom drivers or patches to a host, the process is called an update.

- Update ESXi 4.0 and ESXi 4.1 hosts with the `vihostupdate` command, as discussed in the *vSphere Command-Line Interface Installation and Reference Guide* included in the vSphere 4.1 documentation set.
- Update ESXi 5.0 hosts with `esxcli software vib` commands discussed in the *vSphere Upgrade* documentation included in the vSphere 5.0 documentation set. You cannot run the `vihostupdate` command against ESXi 5.0 or later.
- Update ESXi 5.1 hosts with `esxcli software vib` commands discussed in the *vSphere Upgrade* documentation included in the vSphere 5.1 documentation set.

- Update ESXi 5.5 hosts with `esxcli software vib` commands discussed in the *vSphere Upgrade* documentation included in the vSphere 5.5 documentation set.
- Update ESXi 6.0 hosts with `esxcli software vib` commands discussed in the *vSphere Upgrade* documentation included in the vSphere 6.0 documentation set.



# Managing Files

---

The vSphere CLI includes two commands for file manipulation. `vmkfstools` allows you to manipulate VMFS (Virtual Machine File System) and virtual disks. `vifs` supports remote interaction with files on your ESXi host.

---

**NOTE** See “[Managing Storage](#)” on page 37 for information about storage manipulation commands.

---

This chapter includes the following topics:

- “[Introduction to Virtual Machine File Management](#)” on page 27
- “[Managing the Virtual Machine File System with `vmkfstools`](#)” on page 28
- “[Upgrading VMFS3 Volumes to VMFS5](#)” on page 29
- “[Managing VMFS Volumes](#)” on page 29
- “[Reclaiming Unused Storage Space](#)” on page 31
- “[Using `vifs` to View and Manipulate Files on Remote ESXi Hosts](#)” on page 32

## Introduction to Virtual Machine File Management

You can use the vSphere Web Client or vCLI commands to access different types of storage devices that your ESXi host discovers and to deploy datastores on those devices.

---

**NOTE** 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. Datastores can be used for storing ISO images, virtual machine templates, and floppy images. The vSphere Web Client uses the term datastore exclusively. This manual uses the term datastore and VMFS (or NFS) volume to refer to the same logical container on the physical device.

---

Depending on the type of storage you use, datastores can be backed by the following file system formats:

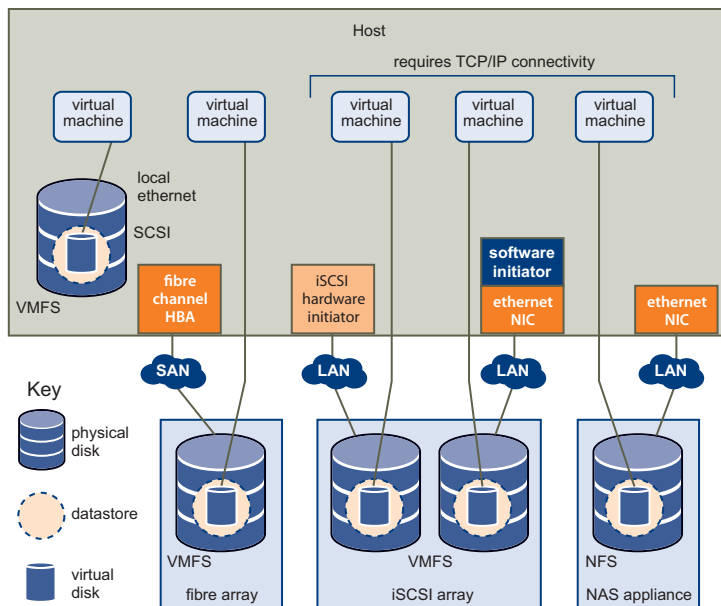
- **Virtual Machine File System (VMFS).** High-performance file system that is optimized for storing virtual machines. Your host can deploy a VMFS datastore on any SCSI-based local or networked storage device, including Fibre Channel and iSCSI SAN equipment. As an alternative to using the VMFS datastore, your virtual machine can have direct access to raw devices and use a mapping file (RDM) as a proxy.

You manage VMFS and RDMs with the vSphere Web Client, or the `vmkfstools` command.

- **Network File System (NFS).** The 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 version 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 volumes do not need to be formatted with a local file system, such as VMFS. You can mount the volumes directly and use them to store and boot virtual machines in the same way that you use VMFS datastores. The host can access a designated NFS volume located on an NFS server, mount the volume, and use it for any storage needs.

You manage NAS storage devices from the vSphere Web Client or with the `esxcli storage nfs` command. The diagram below illustrates different types of storage, but it is for conceptual purposes only. It is not a recommended configuration.

**Figure 3-1.** Virtual Machines Accessing Different Types of Storage



## Managing the Virtual Machine File System with `vmkfstools`

VMFS datastores primarily serve as repositories for virtual machines. You can store multiple virtual machines on the same VMFS volume. 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.

In addition, you can use the VMFS datastores to store other files, such as virtual machine templates and ISO images. VMFS supports file and block sizes that enable virtual machines to run data-intensive applications, including databases, ERP, and CRM, in virtual machines. See the *vSphere Storage* documentation.

You use the `vmkfstools` vCLI to create and manipulate virtual disks, file systems, logical volumes, and physical storage devices on an ESXi host. You can use `vmkfstools` to create and manage a virtual machine file system (VMFS) on a physical partition of a disk and to manipulate files, such as virtual disks, stored on VMFS-3 and NFS. You can also use `vmkfstools` to set up and manage raw device mappings (RDMs).

---

**IMPORTANT** The `vmkfstools` vCLI supports most but not all of the options that the `vmkfstools` ESXi Shell command supports. See VMware Knowledge Base article 1008194.

---

You cannot run `vmkfstools` with `--server` pointing to a vCenter Server system.

---

The *vSphere Storage* documentation includes a complete reference to the `vmkfstools` command that you can use in the ESXi Shell. You can use most of the same options with the `vmkfstools` vCLI command. Specify one of the connection options listed in “[Connection Options for vCLI Host Management Commands](#)” on page 18 in place of `<conn_options>`.

The following options supported by the `vmkfstools` ESXi Shell command are not supported by the `vmkfstools` vCLI command.

- `--breaklock -B`
- `--chainConsistent -e`
- `--eagerzero -k`
- `--fix -x`
- `--lock -L`

- `--migratevirtualdisk -M`
- `--parseimage -Y`
- `--punchzero -K`
- `--snapshotdisk -I`
- `--verbose -v`

## Upgrading VMFS3 Volumes to VMFS5

vSphere 5.0 supports VMFS5 volumes, which have improved scalability and performance. You can upgrade from VMFS3 to VMFS5 by using the vSphere Web Client, the `vmkfstools` ESXi Shell command, or the `esxcli storage vmfs upgrade` command. Pass the volume label or the volume UUID to the ESXCLI command.

---

**IMPORTANT** You cannot upgrade VMFS3 volumes to VMFS5 with the `vmkfstools` command included in vSphere CLI.

---

## Managing VMFS Volumes

Different commands are available for listing, mounting, and unmounting VMFS volumes and for listing, mounting, and unmounting VMFS snapshot volumes.

- Managing VMFS volumes

`esxcli storage filesystem list` shows all volumes, mounted and unmounted, that are resolved, that is, that are not snapshot volumes.

`esxcli storage filesystem unmount` unmounts a currently mounted filesystem. Use this command for snapshot volumes or resolved volumes.

- Managing snapshot volumes

`esxcli storage vmfs snapshot` commands can be used for listing, mounting, and resignaturing snapshot volumes. See [“Mounting Datastores with Existing Signatures”](#) on page 29 and [“Resignaturing VMFS Copies”](#) on page 30.

## Managing Duplicate VMFS Datastores

Each VMFS datastore created in a LUN has a unique UUID that is stored in the file system superblock. When the LUN is replicated or when a snapshot is made, the resulting LUN copy is identical, byte-for-byte, to the original LUN. As a result, if the original LUN contains a VMFS datastore with UUID X, the LUN copy appears to contain an identical VMFS datastore, or a VMFS datastore copy, with the same UUID X.

ESXi hosts can determine whether a LUN contains the VMFS datastore copy, and either mount the datastore copy with its original UUID or change the UUID to resignature the datastore.

When a LUN contains a VMFS datastore copy, you can mount the datastore with the existing signature or assign a new signature. The *vSphere Storage* documentation discusses volume resignaturing in detail.

## Mounting Datastores with Existing Signatures

You can mount a VMFS datastore copy without changing its signature if the original is not mounted. For example, you can 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 can mount the datastore copy and power on the virtual machines at the secondary site.

---

**IMPORTANT** You can mount a VMFS datastore only if it does not conflict with an already mounted VMFS datastore that has the same UUID.

---

When you mount the VMFS datastore, ESXi allows both read and write operations to the datastore that resides on the LUN copy. The LUN copy must be writable. The datastore mounts are persistent and valid across system reboots.

You can mount a datastore with `vicfg-volume` (see [“To mount a datastore with vicfg-volume”](#) on page 30) or with `ESXCLI` (see [“To mount a datastore with ESXCLI”](#) on page 30).

## Mounting and Unmounting with ESXCLI

The `esxcli storage filesystem` commands support mounting and unmounting volumes. You can also specify whether to persist the mounted volumes across reboots by using the `--no-persist` option.

Use the `esxcli storage filesystem` command to list mounted volumes, mount new volumes, and unmount a volume. Specify one of the connection options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

### To mount a datastore with ESXCLI

- 1 List all volumes that have been detected as snapshots.

```
esxcli <conn_options> storage filesystem list
```

- 2 Run `esxcli storage filesystem mount` with the volume label or volume UUID.

By default, the volume is mounted persistently; use `--no-persist` to mount nonpersistently.

```
esxcli <conn_options> storage filesystem volume mount
--volume-label=<label>|--volume-uuid=<VMFS-UUID>
```

This command fails if the original copy is online.

You can later run `esxcli storage filesystem volume unmount` to unmount the snapshot volume.

```
esxcli <conn_options> storage filesystem volume unmount
--volume-label=<label>|--volume-uuid=<VMFS-UUID>
```

## Mounting and Unmounting with vicfg-volume

Use the `vicfg-volume` command to list mounted volumes, mount new volumes, and unmount a volume. Specify one of the connection options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

### To mount a datastore with vicfg-volume

- 1 List all volumes that have been detected as snapshots or replicas.

```
vicfg-volume <conn_options> --list
```

- 2 Run `vicfg-volume --persistent-mount` with the VMFS-UUID or label as an argument to mount a volume.

```
vicfg-volume <conn_options> --persistent-mount <VMFS-UUID|label>
```

This command fails if the original copy is online.

You can later run `vicfg-volume --unmount` to unmount the snapshot or replica volume.

```
vicfg-volume <conn_options> --unmount <VMFS-UUID|label>
```

The `vicfg-volume` command supports resignaturing a snapshot volume and mounting and unmounting the volume. You can also make the mounted volume persistent across reboots and query a list of snapshot volumes and original volumes.

## Resignaturing VMFS Copies

Use datastore resignaturing to retain the data stored on the VMFS datastore copy. When resignaturing a VMFS copy, the ESXi host assigns a new UUID and a new label to the copy, and mounts the copy as a datastore distinct from the original. Because ESXi prevents you from resignaturing the mounted datastore, unmount the datastore before resignaturing.

The default format of the new label assigned to the datastore is `snap-<snapID>-<oldLabel>`, where `<snapID>` is an integer and `<oldLabel>` is the label of the original datastore.



When you perform datastore resignaturing, consider the following points:

- Datastore resignaturing is irreversible.
- The LUN copy that contains the VMFS datastore that you resignature is no longer treated as a LUN copy.
- 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 LUN snapshots.

You can resignature a VMFS copy with ESXCLI (see [“Resignaturing a VMFS Copy with ESXCLI”](#) on page 31) or with `vicfg-volume` see [“Resignaturing a VMFS Copy with vicfg-volume”](#) on page 31.

### Resignaturing a VMFS Copy with ESXCLI

The `esxcli storage vmfs snapshot` commands support resignaturing a snapshot volume. Specify one of the connection options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

#### To resignature a VMFS copy with ESXCLI

- 1 List unresolved snapshots or replica volumes.

```
esxcli <conn_options> storage vmfs snapshot list
```

- 2 (Optional) Unmount the copy.

```
esxcli <conn_options> storage filesystem unmount
```

- 3 Run the `resignature` command.

```
esxcli <conn_options> storage vmfs snapshot resignature
    --volume-label=<label>|--volume-uuid=<id>
```

The command returns to the prompt or signals an error.

After resignaturing, you might have to do the following:

- If the resignatured datastore contains virtual machines, update references to the original VMFS datastore in the virtual machine files, including `.vmx`, `.vmdk`, `.vmsd`, and `.vmsn`.
- To power on virtual machines, register them with the vCenter Server system.

### Resignaturing a VMFS Copy with vicfg-volume

You can use `vicfg-volume` to mount, unmount, and resignature VMFS volumes.

#### To resignature a VMFS copy with vicfg-volume

- 1 Make sure the copy is not mounted.
- 2 Run `vicfg-volume` with the `resignature` option.

```
vicfg-volume <conn_options> --resignature <VMFS-UUID|label>
```

The command returns to the prompt or signals an error.

## Reclaiming Unused Storage Space

When VMFS datastores reside on thin-provisioned LUNs, you can use ESXCLI commands to reclaim the unused logical blocks of a thin-provisioned LUN formatted with VMFS.

When you run the commands, you must specify the volume label (`--volume-label`) or the volume id (`--volume-uuid`) but you cannot specify both.

In each iteration, the command issues unmap commands to the number of file system blocks that are specified by the optional `reclaim-unit` argument, which defaults to 200. For newly created VMFS-5 file systems, the filesystem block size is always 1 MB. For VMFS-3 filesystems or VMFS-5 filesystems that were upgraded from VMFS-3, the filesystem block size could be one of 1, 2, 4, 8 MB.

The following examples illustrate how to use the command.

```
# esxcli storage vmfs unmap --volume-label datastore1 --reclaim-unit 100
# esxcli storage vmfs unmap -l datastore1 -n 100
# esxcli storage vmfs unmap --volume-uuid 515615fb-1e65c01d-b40f-001d096dbf97 --reclaim-unit 500
# esxcli storage vmfs unmap -u 515615fb-1e65c01d-b40f-001d096dbf97 -n 500

# esxcli storage vmfs unmap -l datastore1
# esxcli storage vmfs unmap -u 515615fb-1e65c01d-b40f-001d096dbf97
```

## Using vifs to View and Manipulate Files on Remote ESXi Hosts

You can use the `vifs` utility for datastore file management.



**CAUTION** If you manipulate files directly, your vSphere setup might end up in an inconsistent state. Use the vSphere Web Client or one of the other vCLI commands to manipulate virtual machine configuration files and virtual disks.

The `vifs` command performs common operations such as copy, remove, get, and put on ESXi files and directories. The command is supported against ESXi hosts but not against vCenter Server systems.

Some similarities between `vifs` and DOS or UNIX/Linux file system management utilities exist, but there are many differences. For example, `vifs` does not support wildcard characters or current directories and, as a result, relative pathnames. Use `vifs` only as documented.

Instead of using the `vifs` command, you can browse datastore contents and host files by using a Web browser. Connect to the following location:

```
http://ESX_host_IP_Address/host
http://ESX_host_IP_Address/folder
```

You can view data center and datastore directories from this root URL. For example:

```
http://<ESXi_addr>/folder?dcPath=ha-datacenter
http://<ESXi_host_name>/folder?dcPath=ha-datacenter
```

The ESXi host prompts for a user name and password.

The `vifs` command supports different operations for the following groups of files and directories. Different operations are available for each group, and you specify locations with a different syntax. The behavior differs for vSphere 4.x and vSphere 5.0.

	vSphere 4.x	vSphere 5.0
Host	Host configuration files. You must specify the file's unique name identifier. Specify host locations by using the <code>/host/&lt;path&gt;</code> syntax.	Host configuration files. You must specify the file's unique name identifier. Specify host locations by using the <code>/host/&lt;path&gt;</code> syntax. You cannot list subdirectories of <code>/host</code> .

	vSphere 4.x	vSphere 5.0
Temp	The /tmp directory and files in that directory. Specify temp locations by using the /tmp/dir/subdir syntax.	Not supported.
Datstores	Datastore files and directories. You have two choices for specifying a datastore: <ul style="list-style-type: none"> <li>■ Datastore prefix style: '[ds_name] relative_path'. For example: '[myStorage1] testvms/VM1/VM1.vmx' (Linux) or "[myStorage1] testvms/VM1/VM1.vmx" (Windows)</li> <li>■ URL style: /folder/dir/subdir/file?dsName=&lt;name&gt;. For example: '/folder/testvms/VM1/VM1.vmx?dsName=myStorage1' (Linux) or "/folder/testvms/VM1/VM1.vmx?dsName=myStorage1" (Windows)</li> </ul> The two example paths refer to a virtual machine configuration file for the virtual machine VM1 in the testvms/VM1 directory of the myStorage1 datastore.	

To avoid problems with directory names that use special characters or spaces, enclose the path in quotes for both operating systems.

When you run `vifs`, you can specify the operation name and argument and one of the standard connection options. Use aliases, symbolic links, or wrapper scripts to simplify the invocation syntax.

**IMPORTANT** The concepts of working directory and last directory or file operated on are not supported with `vifs`.

### Options

`vifs` command-specific options allow you to retrieve and upload files from the remote host and perform a number of other operations. All `vifs` options work on datastore files or directories. Some options also work on host files and files in the `temp` directory. You must also specify connection options.

Command	Description	Target	Syntax
<code>--copy</code> <code>-c &lt;source&gt;</code> <code>&lt;target&gt;</code>	Copies a file in a datastore to another location in a datastore. The <code>&lt;source&gt;</code> must be a remote source path, the <code>&lt;target&gt;</code> a remote target path or directory. The <code>--force</code> option replaces existing destination files.	Datastore Temp	<code>copy src_file_path</code> <code>dst_directory_path</code> <code>[--force]</code> <code>copy src_file_path</code> <code>dst_file_path [--force]</code>
<code>--dir</code> <code>-D &lt;remote_dir&gt;</code>	Lists the contents of a datastore directory.	Datastore Temp	<code>dir</code> <code>datastore_directory_path</code>
<code>--force</code> <code>-F</code>	Overwrites the destination file. Used with <code>--move</code> and <code>--copy</code> .	Datastore Temp	<code>copy src_file_path</code> <code>dst_file_path [--force]</code>
<code>--get</code> <code>-g &lt;remote_path&gt;</code> <code>&lt;local_path&gt;</code>	Downloads a file from the ESXi host to the machine on which you run vCLI. This operation uses HTTP GET.	Datastore Host	<code>get src_dstore_file_path</code> <code>dst_local_file_path</code> <code>get src_dstore_dir_path</code> <code>dst_local_file_path</code>
<code>--listdc</code> <code>-C</code>	Lists the data center paths available on an ESXi system.	Datastore Host	
<code>--listds</code> <code>-S</code>	Lists the datastore names on the ESXi system. When multiple data centers are available, use the <code>--dc (-Z)</code> argument to specify the name of the data center from which you want to list the datastore.	Datastore Host	<code>vifs --listds</code>
<code>--mkdir</code> <code>-M &lt;remote_dir&gt;</code>	Creates a directory in a datastore. This operation fails if the parent directory of <code>dst_datastore_file_path</code> does not exist.	Datastore Temp	<code>mkdir dst_directory_path</code>

Command	Description	Target	Syntax
<pre>--move -m &lt;source&gt; &lt;target&gt;</pre>	<p>Moves a file in a datastore to another location in a datastore. The &lt;source&gt; must be a remote source path, the &lt;target&gt; a remote target path or directory.</p> <p>The <code>--force</code> option replaces existing destination files.</p>	Datastore Temp	<pre>move src_file_path dst_directory_path [--force] move src_file_path dst_file_path [--force]</pre>
<pre>--put -p &lt;local_path&gt; &lt;remote_path&gt;</pre>	<p>Uploads a file from the machine on which you run vCLI to the ESXi host. This operation uses HTTP PUT.</p> <p>This command can replace existing host files but cannot create new files.</p>	Datastore Host Temp	<pre>put src_local_file_path dst_file_path put src_local_file_path dst_directory_path</pre>
<pre>--rm -r &lt;remote_path&gt;</pre>	<p>Deletes a datastore file.</p>	Datastore Temp	<pre>rm dst_file_path</pre>
<pre>--rmdir -R &lt;remote_dir&gt;</pre>	<p>Deletes a datastore directory. This operation fails if the directory is not empty.</p>	Datastore Temp	<pre>rmdir dst_directory_path</pre>

## Examples

You can use `vifs` to interact with the remote ESXi or vCenter Server system in a variety of ways. Specify one of the connection options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`. The examples illustrate use on a Linux system, use double quotes instead of single quotes when on a Windows system.

### Listing Remote Information

- List all data centers on a vCenter Server system with `--listdc`, using `--server` to point to the vCenter Server system.

```
vifs --server <my_vc>--username administrator --password <pswd> --listdc
```

- List all datastores on a vCenter Server system with `--listds`.

```
vifs --server <my_vc> --username administrator --password <pswd> --dc kw-dev --listds
```

- List all datastores on an ESXi host with `--listds`.

```
vifs --server <my_ESXi> --username root --password <pswd> --listds
```

The command lists the names of all datastores on the specified server.

You can use each name that has been returned to refer to datastore paths by using square bracket notation, as follows:

```
'[my_datastore] dir/subdir/file'
```

- List the content of a directory in a datastore.

```
vifs --server <my_ESXi> --username root --password <pswd>--dir '[Storage1]'
```

```
vifs --server <my_ESXi> --username root --password <pswd> --dir '[Storage1] WindowsXP'
```

The command lists the directory content. In this example, the command lists the contents of a virtual machine directory.

```
Content Listing
-----
vmware-37.log
vmware-38.log
...
vmware.log
...
winxpPro-sp2.vmdk
winxpPro-sp2.vmx
winxpPro-sp2.vmx
...
```

- List the contents of one of the datastores.

```
vifs <conn_options> --dir '[osdc-cx700-02]'
```

The command lists the complete contents of the datastore.

### Working with Directories and Files on the Remote Server

- Create a new directory in a datastore with `--mkdir <remote_dir>`.

```
vifs --server <my_ESXi> --username root --password <pswd> --mkdir '[Storage1] test'
```

- Remove a directory with `--rmdir <remote_dir>`.

```
vifs --server <my_ESXi> --username root --password <pswd> --rmdir '[Storage1] test'.
```

- Forcibly remove a directory with `--rmdir --force <remote_dir>`.

```
vifs --server <my_ESXi> --username root --password <pswd> --rmdir '[Storage1] test2' --force
```

- Update a file on the remote server with `--put <local_path> <remote_path>`.

```
vifs --server <my_ESXi> --username root --password <pswd>
--put /tmp/testfile '[Storage1] test/testfile'
```

- Retrieve a file from the remote server with `--get <remote_path> <local_path>|<local_dir>`. The command overwrites the local file if it exists. If you do not specify a file name, the file name of the remote file is used.

```
vifs --server <my_ESXi> --username root --password <pswd> --get '[Storage1] test/testfile'
/tmp/tfile
vifs --server <my_ESXi> --username root --password <pswd> --get '[Storage1] test/testfile'
/tmp
```

- Delete a file on the remote server with `-rm <remote_path>`.

```
vifs --server <my_ESXi> --username root --password <pswd> --rm '[Storage1] test2/testfile'
```

- Forcibly remove a file on the remote server with `--rm <remote_path> --force`.

```
vifs --server <my_ESXi> --username root --password <pswd> --rm '[Storage1] test2/testfile2'
--force
```

- Move a file from one location on the remote server to another location with `--move <remote_source_path> <remote_target_path>`. If you specify a file name, the file is moved and renamed at the same time.

```
vifs --server <my_ESXi> --username root --password <pswd> --move '[Storage1] test/tfile'
'[Storage1] newfile'
```

If the target file already exists on the remote server, the command fails unless you use `--force`.

```
vifs --server <my_ESXi> --username root --password <pswd> --move '[Storage1] test/tfile2'
'[Storage1] test2/tfile' --force
```

- Create a copy of a file on the remote server at a different location on the remote server.

```
vifs --server <my_ESXi> --username root --password <pswd> --copy '[Storage1] test/tfile'
'[Storage1] test/tfile2'
```

If the target file already exists on the remote server, the command fails unless you use `--force`.

```
vifs --server <my_ESXi> --username root --password <pswd> --copy '[Storage1] test/tfile'
'[Storage1] test/tfile2' --force
```

### Example Scenario

The following example scenario illustrates other uses of `vifs`. Specify one of the connection options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

#### To manage files and directories on the remote ESXi system

- 1 Create a directory in the datastore.

```
vifs <conn_options> --mkdir '[osdc-cx700-03] vcli_test'
```

You must specify the precise path; there is no concept of a relative path.

- 2 Place a file that is on the system from which you are running the commands into the newly created directory.

```
vifs <conn_options> --put /tmp/test_doc '[osdc-cx700-03] vcli_test/test_doc'
```

- 3 Move a file into a virtual machine directory.

```
vifs <conn_options> --move '[osdc-cx700-03] vcli_test/test_doc'  
'[osdc-cx700-03] winxpPro-sp2/test_doc'
```

A message indicates success or failure.

- 4 Retrieve one of the files from the remote ESXi system.

The following example retrieves a log file for analysis.

```
vifs <conn_options> --get '[osdc-cx700-03] winxpPro-sp2/vmware.log' ~user1/vmware.log
```

- 5 Clean up by removing the file and directory you created earlier.

```
vifs <conn_options> --rm '[osdc-cx700-03] vcli_test/test_doc'  
vifs <conn_options> --rmdir '[osdc-cx700-03] vcli_test'
```

# Managing Storage

---

A 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.

To store virtual disk files and manipulate the files, a host requires dedicated storage space. ESXi storage is storage space on a variety of physical storage systems, local or networked, that a host uses to store virtual machine disks.

This chapter includes the following topics:

- [“Introduction to Storage”](#) on page 37
- [“Examining LUNs”](#) on page 40
- [“Detaching Devices and Removing a LUN”](#) on page 43
- [“Working with Permanent Device Loss”](#) on page 43
- [“Managing Paths”](#) on page 44
- [“Managing Path Policies”](#) on page 47
- [“Managing NFS/NAS Datastores”](#) on page 51
- [“Monitoring and Managing SAN Storage”](#) on page 53
- [“Monitoring and Managing Virtual SAN Storage”](#) on page 53
- [“Monitoring vSphere Flash Read Cache”](#) on page 54
- [“Monitoring and Managing Virtual Volumes”](#) on page 55
- [“Migrating Virtual Machines with svmotion”](#) on page 56
- [“Configuring FCoE Adapters”](#) on page 57
- [“Scanning Storage Adapters”](#) on page 58
- [“Retrieving SMART Information”](#) on page 58

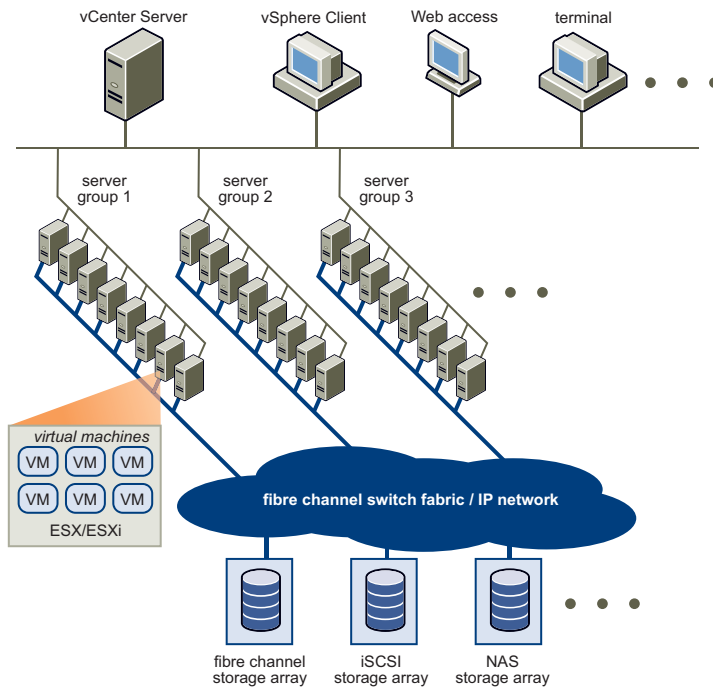
[Chapter 5, “Managing iSCSI Storage,”](#) on page 59 discusses iSCSI storage management. [Chapter 6, “Managing Third-Party Storage Arrays,”](#) on page 87 explains how to manage the Pluggable Storage Architecture, including Path Selection Plugin (PSP) and Storage Array Type Plugin (SATP) configuration.

For information on masking and unmasking paths with ESXCLI, see the *vSphere Storage* documentation.

## Introduction to Storage

Fibre Channel SAN arrays, iSCSI SAN arrays, and NAS arrays are widely used storage technologies supported by VMware vSphere to meet different data center storage needs. The storage arrays are connected to and shared between groups of servers through storage area networks. This arrangement allows aggregation of the storage resources and provides more flexibility in provisioning them to virtual machines.

**Figure 4-1.** vSphere Data Center Physical Topology



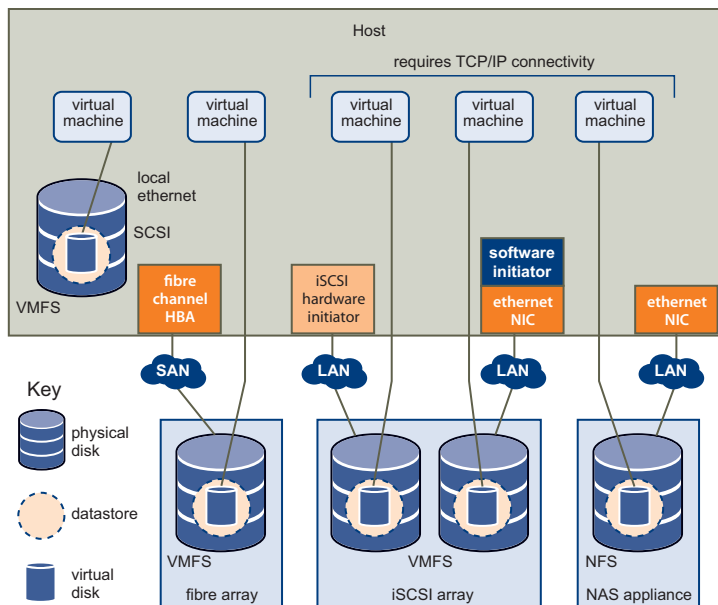
## How Virtual Machines Access Storage

A virtual disk hides the physical storage layer from the virtual machine’s operating system. Regardless of the type of storage device that your host uses, the virtual disk always appears to the virtual machine as a mounted SCSI device. As a result, you can run operating systems that are not certified for specific storage equipment, such as SAN, in the virtual machine.

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.

Figure 4-2 depicts five virtual machines that use different types of storage to illustrate the differences between each type.

**Figure 4-2.** Virtual Machines Accessing Different Types of Storage





You can use vCLI commands to manage the virtual machine file system and storage devices.

- **VMFS.** Use `vmkfstools` to create, modify, and manage VMFS virtual disks and raw device mappings. See [“Managing the Virtual Machine File System with vmkfstools”](#) on page 28 for an introduction and the *vSphere Storage* documentation for a detailed reference.
- **Datstores.** Several commands allow you to manage datstores and are useful for multiple protocols.
  - **LUNs.** Use `esxcli storage core` or `vicfg-scsidevs` commands to display available LUNs and mappings for each VMFS volume to its corresponding partition. See [“Examining LUNs”](#) on page 40.
  - **Path management.** Use `esxcli storage core` or `vicfg-mpath` commands to list information about Fibre Channel or iSCSI LUNs and to change a path’s state. See [“Managing Paths”](#) on page 44. Use the `ESXCLI` command to view and modify path policies. See [“Managing Path Policies”](#) on page 47.
  - **Rescan.** Use `esxcli storage core` or `vicfg-rescan adapter rescan` to perform a rescan operation each time you reconfigure your storage setup. See [“Scanning Storage Adapters”](#) on page 58.
- **Storage devices.** Several commands manage only specific storage devices.
  - **NFS storage.** Use `esxcli storage nfs` or `vicfg-nas` to manage NAS storage devices. See [“Managing NFS/NAS Datstores”](#) on page 51.
  - **iSCSI storage.** Use `esxcli iscsi` or `vicfg-iscsi` to manage both hardware and software iSCSI. See [“Managing iSCSI Storage”](#) on page 59.
- **Software-defined storage.** vSphere supports several types of software-defined storage.
  - **Virtual SAN storage.** Use commands in the `esxcli vsan` namespace to manage Virtual SAN. See [“Monitoring and Managing Virtual SAN Storage”](#) on page 53.
  - **Virtual Flash storage.** Use commands in the `esxcli storage vflash` namespace to manage VMware vSphere Flash Read Cache.
  - **Virtual volumes.** Virtual volumes offer a different layer of abstraction than datstores. As a result, finer-grained management is possible.. Use commands in the `esxcli storage vvol` namespace.

## Datstores

ESXi hosts use storage space on a variety of physical storage systems, including internal and external devices and networked storage. A host can discover storage devices to which it has access and format them as datstores. Each datstore is a special logical container, analogous to a file system on a logical volume, where the host places virtual disk files and other virtual machine files. Datstores hide specifics of each storage product and provide a uniform model for storing virtual machine files.

Depending on the type of storage you use, datstores can be backed by the following file system formats:

- **Virtual Machine File System (VMFS).** High-performance file system optimized for storing virtual machines. Your host can deploy a VMFS datstore on any SCSI-based local or networked storage device, including Fibre Channel and iSCSI SAN equipment.

As an alternative to using the VMFS datstore, your virtual machine can have direct access to raw devices and use a mapping file (RDM) as a proxy. See [“Managing the Virtual Machine File System with vmkfstools”](#) on page 28.

- **Network File System (NFS).** File system on a NAS storage device. ESXi supports NFS version 3 over TCP/IP. The host can access a designated NFS volume located on an NFS server, mount the volume, and use it for any storage needs.

## Storage Device Naming

Each storage device, or LUN, is identified by several device identifier 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. SCSI INQUIRY device identifiers 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 for information on Page 83.

- **Path-based identifier.** If the device does not provide the information on Page 83 of the T10 committee SCSI-3 documentation, 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 identifiers.

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:

```
vmhba<adapter>:C<channel>:T<target>:L<LUN>
```

- vmhba<adapter> is the name of the storage adapter. The name refers to the physical adapter on the host, not the SCSI controller used by the virtual machines.
- C<channel> is the storage channel number. Software iSCSI adapters and dependent hardware adapters use the channel number to show multiple paths to the same target.
- T<target> is the target number. Target numbering is determined by the host and might change if the mappings of targets that are visible to the host change. Targets that are shared by different hosts might not have the same target number.
- L<LUN> is the LUN number that shows the position of the LUN within the target. The number is provided by the storage system. If a target has only one LUN, the LUN number is always zero (0).

## Legacy Identifiers

In addition to the SCSI INQUIRY or mpx identifiers, ESXi generates an alternative legacy name, called VML name, for each device. Use the device UID instead.

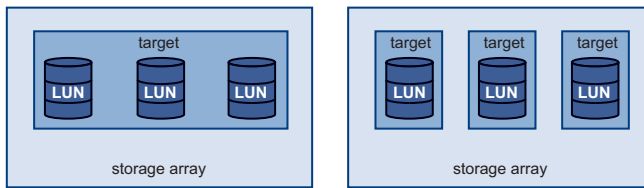
## Examining LUNs

A LUN (Logical Unit Number) is an identifier for a disk volume in a storage array target.

### Target and Device Representation

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

Different storage vendors present their storage systems to ESXi hosts in different ways. Some vendors present a single target with multiple LUNs on it. Other vendors, especially iSCSI vendors, present multiple targets with one LUN each.

**Figure 4-3. Target and LUN Representations**

In [Figure 4-3](#), three LUNs are available in each configuration. On the left, the host sees one target, but that target has three LUNs that can be used. Each LUN represents an individual storage volume. On the right, the host sees three different targets, each having one LUN.

## Examining LUNs with `esxcli storage core`

Use `esxcli storage core` to display information about available LUNs on ESXi 5.0.

You can run one of the following commands to examine LUNs. Specify one of the connection options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- List all logical devices known on this system with detailed information.

```
esxcli <conn_options> storage core device list
```

The command lists device information for all logical devices on this system. The information includes the name (UUID), device type, display name, and multipathing plugin. Specify the `--device` option to only list information about a specific device. See [“Storage Device Naming”](#) on page 39 for background information.

```
naa.5000c50037b3967e
  Display Name: <name> (naa.5000c50037b3967e)
  Has Settable Display Name: true
  Size: 953869
  Device Type: Direct-Access
...
naa.500000e014e7a4e0
  Display Name: <name> (naa.500000e014e7a4e0)
  Has Settable Display Name: true
  Size: 70007
  Device Type: Direct-Access
...
mpx.vmhba0:C0:T0:L0
  Display Name: Local <name> CD-ROM (mpx.vmhba0:C0:T0:L0)
  Has Settable Display Name: false
  Size: 0
  Device Type: CD-ROM
```

- List a specific logical device with its detailed information.

```
esxcli <conn_options> storage core device list -d mpx.vmhba32:C0:T1:L0
```

- List all device unique identifiers.

```
esxcli <conn_options> storage core device list
```

The command lists the primary UID for each device (`naa.xxx` or other primary name) and any other UIDs for each UID (VML name). You can specify `--device` to only list information for a specific device.

- Print mappings for VMFS volumes to the corresponding partition, path to that partition, VMFS UUID, extent number, and volume names.

```
esxcli <conn_option> storage filesystem list
```

- Print HBA devices with identifying information.

```
esxcli <conn_options> storage core adapter list
```

The return value includes adapter and UID information.

- Print a mapping between HBAs and the devices it provides paths to.

```
esxcli <conn_options> storage core path list
```

## Examining LUNs with vicfg-scsidevs

Use `vicfg-scsidevs` to display information about available LUNs on ESXi 4.x hosts.

---

**IMPORTANT** You can run `vicfg-scsidevs --query` and `vicfg-scsidevs --vmfs` against ESXi version 3.5. The other options are supported only against ESXi version 4.0 and later.

---

You can run one of the following commands to examine LUNs. Specify one of the connection options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- List all logical devices known on this system with detailed information.

```
vicfg-scsidevs <conn_options> --list
```

The command lists device information for all logical devices on this system. The information includes the name (UUID), device type, display name, and multipathing plugin. Specify the `--device` option to only list information about a specific device. The following example shows output for two devices; the actual listing might include multiple devices and the precise format differs between releases.

```
mpx.vmhba2:C0:T1:L0
  Device Type: cdrom
  Size: 0 MB
  Display Name: Local HL-DT-ST (mpx.vmhba2:C0:T1:L0)
  Plugin: NMP
  Console Device: /vmfs/devices/cdrom/mpx.vmhba2:C0:T1:L0
  Devfs Path: /vmfs/devices/cdrom/mpx.vmhba2:C0:T1:L0
  Vendor: SONY      Model: DVD-ROM GDRXX8XX Revis: 3.00
  SCSI Level: 5 Is Pseudo: Status:
  Is RDM Capable: Is Removable:
  Other Names:
    vml.000N000000XXXdXXXXXXXXXaXXXaXX
    VAAI Status: nnnn

naa.60060...
  Device Type: disk
  Size: 614400 MB
  Display Name: DGC Fibre Channel Disk (naa.60060...)
  ...
```

- List all logical devices with abbreviated information.

```
vicfg-scsidevs <conn_options> --compact-list
```

The information includes the device ID, device type, size, plugin, and device display name.

- List all device unique identifiers.

```
vicfg-scsidevs <conn_options> --uids
```

The command lists the primary UID for each device (`naa.xxx` or other primary name) and any other UIDs for each UID (VML name). You can specify `--device` to only list information for a specific device.

- List a specific logical device with its detailed information.

```
vicfg-scsidevs <conn_options> -l -d mpx.vmhba32:C0:T1:L0
```

- Print mappings for VMFS volumes to the corresponding partition, path to that partition, VMFS uuid, extent number, and volume names.

```
vicfg-scsidevs <conn_options> --vmfs
```

- Print HBA devices with identifying information.

```
vicfg-scsidevs <conn_options> --hbas
```

The return value includes the adapter ID, driver ID, adapter UID, PCI, vendor, and model.

- Print a mapping between HBAs and the devices it provides paths to.

```
vicfg-scsidevs <conn_options> --hba-device-list
```

## Detaching Devices and Removing a LUN

Before you can remove a LUN, you must detach the corresponding device by using the vSphere Web Client, or the `esxcli storage core device set` command. Detaching a device brings a device offline. Detaching a device does not impact path states. If the LUN is still visible, the path state is not set to dead.

### To detach a device and remove a LUN

- 1 Migrate virtual machines from the device you plan to detach.

For information on migrating virtual machines, see the *vCenter Server and Host Management* documentation.

- 2 Unmount the datastore deployed on the device. See [“Mounting and Unmounting with ESXCLI”](#) on page 30.

If the unmount fails, ESXCLI returns an error. If you ignore that error, you will get an error in step 4 when you attempt to detach a device with a VMFS partition still in use.

- 3 If the unmount failed, check whether the device is in use.

```
esxcli storage core device world list -d <device>
```

If a VMFS volume is using the device indirectly, the world name includes the string `idle0`. If a virtual machine uses the device as an RDM, the virtual machine process name is displayed. If any other process is using the raw device, the information is displayed.

- 4 Detach the storage device.

```
esxcli storage core device set -d naa.xxx... --state=off
```

Detach is persistent across reboots and device unregistration. Any device that is detached remains detached until a manual attach operation. Rescan does not bring persistently detached devices back online. A persistently detached device comes back in the off state.

ESXi maintains the persistent information about the device's offline state even if the device is unregistered. You can remove the device information by running `esxcli storage core device detached remove -d naa.12`.

- 5 (Optional) To troubleshoot the detach operation, list all devices that were detached manually.

```
esxcli storage core device detached list
```

- 6 Perform a rescan.

```
esxcli <conn_options> storage core adapter rescan
```

When you have completed storage reconfiguration, you can reattach the storage device, mount the datastore, and restart the virtual machines.

### To reattach the device

- 1 (Optional) Check that the device is detached.

```
esxcli storage core device detached list
```

- 2 Attach the device.

```
esxcli storage core device set -d naa.XXX --state=on
```

- 3 Mount the datastore and restart virtual machines. See [“Mounting Datastores with Existing Signatures”](#) on page 29.

## Working with Permanent Device Loss

With earlier ESXi releases, an APD (All Paths Down) event results when the LUN becomes unavailable. The event is difficult for administrators because they do not have enough information about the state of the LUN to know which corrective action is appropriate.

In ESXi 5.0, the ESXi host can determine whether the cause of an All Paths Down (APD) event is temporary, or whether the cause is permanent device loss. A PDL status occurs when the storage array returns SCSI sense codes indicating that the LUN is no longer available or that a severe, unrecoverable hardware problem exist with it. ESXi has an improved infrastructure that can speed up operations of upper-layer applications in a device loss scenario.

---

**IMPORTANT** Do not plan for APD/PDL events, for example, when you want to upgrade your hardware. Instead, perform an orderly removal of LUNs from your ESXi server, which is described in [“Detaching Devices and Removing a LUN”](#) on page 43, perform the operation, and add the LUN back.

---

### To Remove a PDL LUN

How you remove a PDL LUN depends on whether it was in use.

- If the LUN that goes into PDL is not in use by any user process or by the VMkernel, the LUN disappears by itself after a PDL.
- If the LUN was in use when it entered PLD, delete the LUN manually by following the process described in [“Detaching Devices and Removing a LUN”](#) on page 43.

### To Reattach a PDL LUN

- 1 Return the LUN to working order.
- 2 Remove any users of the device.

You cannot bring a device back without removing active users. The ESXi host cannot know whether the device that was added back has changed. ESXi must be able to treat the device similarly to a new device being discovered.

- 3 Perform a rescan to get the device back in working order.

## Managing Paths

To maintain a constant connection between an ESXi host and its storage, ESXi supports multipathing. With multipathing you can use more than one physical path for transferring data between the ESXi host and the external storage device.

In case of failure of an element in the SAN network, such as an HBA, switch, or cable, the ESXi host can fail over to another physical path. On some devices, multipathing also offers load balancing, which redistributes I/O loads between multiple paths to reduce or eliminate potential bottlenecks.

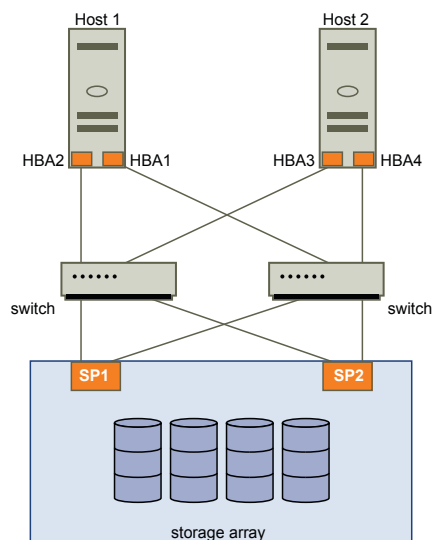
The storage architecture in vSphere 4.0 and later supports a special VMkernel layer, Pluggable Storage Architecture (PSA). The PSA is an open modular framework that coordinates the simultaneous operation of multiple multipathing plugins (MPPs). You can manage PSA using ESXCLI commands. See [“Managing Third-Party Storage Arrays”](#) on page 87. This section assumes you are using only PSA plugins included in vSphere by default.

### Multipathing with Local Storage and FC SANs

In a simple multipathing local storage topology, you can use one ESXi host with two HBAs. The ESXi host connects to a dual-port local storage system through two cables. This configuration ensures fault tolerance if one of the connection elements between the ESXi host and the local storage system fails.

To support path switching with FC SAN, the ESXi host typically has two HBAs available from which the storage array can be reached through one or more switches. Alternatively, the setup can include one HBA and two storage processors so that the HBA can use a different path to reach the disk array.

In [Figure 4-4](#), multiple paths connect each host with the storage device. For example, if HBA1 or the link between HBA1 and the 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 4-4.** FC Multipathing

If SP1 or the link between SP1 and the switch breaks, SP2 takes over and provides the connection between the switch and the storage device. This process is called SP failover. ESXi multipathing supports HBA and SP failover.

After you have set up your hardware to support multipathing, you can use the vSphere Web Client or vCLI commands to list and manage paths. You can perform the following tasks.

- List path information with `vicfg-mpath` or `esxcli storage core path`. See [“Listing Path Information”](#) on page 45.
- Change path state with `vicfg-mpath` or `esxcli storage core path`. See [“Changing the State of a Path”](#) on page 46.
- Change path policies with ESXCLI. See [“Setting Policy Details for Devices that Use Round Robin”](#) on page 50.
- Mask paths with ESXCLI. See the vSphere Storage documentation.
- Manipulate the rules that match paths to multipathing plugins to newly discovered devices with `esxcli claimrule`. See [“Managing Claim Rules”](#) on page 95.
- Run or rerun claim rules or unclaim paths. See [“Managing Claim Rules”](#) on page 95.
- Rescan with `vicfg-rescan`. See [“Scanning Storage Adapters”](#) on page 58.

## Listing Path Information

You can list path information with ESXCLI or with `vicfg-mpath`.

### Listing Path Information with ESXCLI

You can run `esxcli storage core path` to display information about Fibre Channel or iSCSI LUNs.

---

**IMPORTANT** Use industry-standard device names, with format `eui.xxx` or `naa.xxx` to ensure consistency. Do not use VML LUN names unless device names are not available.

Names of virtual machine HBAs are not guaranteed to be valid across reboots.

---

You can display information about paths by running `esxcli storage core path`. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- List all devices with their corresponding paths, state of the path, adapter type, and other information.

- ```
esxcli <conn_options> storage core path list
```
- Limit the display to only a specified path or device.
 

```
esxcli <conn_options> storage core path list --path <path>
esxcli <conn_options> storage core path list --device <device>
```
  - List the statistics for the SCSI paths in the system. You can list all paths or limit the display to a specific path.
 

```
esxcli <conn_options> storage core path stats get
esxcli <conn_options> storage core path stats get --path <path>
```
  - List detailed information for the paths for the device specified with `--device`.
 

```
esxcli <conn_options> storage core path list -d <naa.xxxxxx>
```
  - List all adapters.
 

```
esxcli <conn_options> storage core adapter list
```
  - Rescan all adapters.
 

```
esxcli <conn_options> storage core adapter rescan
```

### Listing Path Information with `vicfg-mpath`

You can run `vicfg-mpath` to list information about Fibre Channel or iSCSI LUNs.

---

**IMPORTANT** Use industry-standard device names, with format `eui.xxx` or `naa.xxx` to ensure consistency. Do not use VML LUN names unless device names are not available.

Names of virtual machine HBAs are not guaranteed to be valid across reboots.

---

You can display information about paths by running `vicfg-mpath` with one of the following options. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- List all devices with their corresponding paths, state of the path, adapter type, and other information.
 

```
vicfg-mpath <conn_options> --list-paths
```
- Display a short listing of all paths.
 

```
vicfg-mpath <conn_options> --list-compact
```
- List all paths with adapter and device mappings.
 

```
vicfg-mpath <conn_options> --list-map
```
- List paths and detailed information by specifying the path UID (long path). The path UID is the first item in the `vicfg-mpath --list display`.
 

```
vicfg-mpath <conn_options> --list
-P sas.5001c231c79c4a00-sas.1221000001000000-naa.5000c5000289c61b
```
- List paths and detailed information by specifying the path runtime name.
 

```
vicfg-mpath <conn_options> -l -P vmhba32:C0:T0:L0
```

The return information includes the runtime name, device, device display name, adapter, adapter identifier, target identifier, plugin, state, transport, and adapter and target transport details.
- List detailed information for the paths for the device specified with `--device`.
 

```
vicfg-mpath <conn_options> -l -d mpv.vmhba32:C0:T1:L0
vicfg-mpath <conn_options> --list --device naa.60060...
```

### Changing the State of a Path

You can change the state of a path with ESXCLI or with `vicfg-mpath`.



## Changing Path State with ESXCLI

You can temporarily disable paths for maintenance or other reasons, and enable the path when you need it again. You can disable paths with ESXCLI. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

If you are changing a path’s state, the change operation fails if I/O is active when the path setting is changed. Reissue the command. You must issue at least one I/O operation before the change takes effect.

### To disable a path with ESXCLI

- 1 (Optional) List all devices and corresponding paths.

```
esxcli <conn_options> storage core path list
```

The display includes information about each path’s state.

- 2 Set the state of a LUN path to off.

```
esxcli <conn_options> storage core path set --state off --path vmhba32:C0:T1:L0
```

When you are ready, set the path state to active again.

```
esxcli <conn_options> storage core path set --state active --path vmhba32:C0:T1:L0
```

## Changing Path State with vicfg-mpath

You can disable paths with `vicfg-mpath`. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

If you are changing a path’s state, the change operation fails if I/O is active when the path setting is changed. Reissue the command. You must issue at least one I/O operation before the change takes effect.

### To disable a path with vicfg-mpath

- 1 (Optional) List all devices and corresponding paths.

```
vicfg-mpath <conn_options> --list-paths
```

The display includes information about each path’s state.

- 2 Set the state of a LUN path to off.

```
vicfg-mpath <conn_options> --state off --path vmhba32:C0:T1:L0
```

When you are ready, set the path state to active again.

```
vicfg-mpath <conn_options> --state active --path vmhba32:C0:T1:L0
```

## Managing Path Policies

For each storage device managed by NMP (not PowerPath), an ESXi host uses a path selection policy. If you have a third-party PSP installed on your host, its policy also appears on the list. The following path policies are supported by default.

**Table 4-1.** Supported Path Policies

| Policy        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| VMW_PSP_FIXED | <p>The host uses the designated preferred path, if it has been configured. Otherwise, the host selects the first working path discovered at system boot time. If you want the host to use a particular preferred path, specify it through the vSphere Web Client, or by using <code>esxcli storage nmp psp fixed deviceconfig set</code>. See “<a href="#">Changing Path Policies</a>” on page 49.</p> <p>The default policy for active-active storage devices is VMW_PSP_FIXED.</p> <p><b>Note:</b> 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.</p> |
| VMW_PSP_MRU   | <p>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 active-passive storage devices.</p> <p>The VMW_PSP_MRU ranking capability allows you to assign ranks to individual paths. To set ranks to individual paths, use the <code>esxcli storage nmp psp generic pathconfig set</code> command. For details, see the VMware knowledge base article 2003468.</p>                                                                                                                        |
| VMW_PSP_RR    | <p>The host uses an automatic path selection algorithm that rotates through all active paths when connecting to active-passive arrays, or through all available paths when connecting to active-active arrays. Automatic path selection implements load balancing across the physical paths available to your host. Load balancing is the process of spreading I/O requests across the paths. The goal is to optimize throughput performance such as I/O per second, megabytes per second, or response times.</p> <p>VMW_PSP_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.</p>                                          |

The type of array and the path policy determine the behavior of the host.

**Table 4-2.** Path Policy Effects

| Policy             | Active/Active Array                                                      | Active/Passive Array                                                                                                                       |
|--------------------|--------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Most Recently Used | Administrator action is required to fail back after path failure.        | Administrator action is required to fail back after path failure.                                                                          |
| Fixed              | VMkernel resumes using the preferred path when connectivity is restored. | VMkernel attempts to resume by using the preferred path. This action can cause path thrashing or failure when another SP now owns the LUN. |
| Round Robin        | No fail back.                                                            | Next path in round robin scheduling is selected.                                                                                           |

## Multipathing Considerations

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 remains 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.

## Changing Path Policies

You can change path policies with ESXCLI or with `vicfg-mpath`.

### Changing Path Policies with ESXCLI

You can change the path policy with ESXCLI. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

#### To change the path policy with ESXCLI

- 1 Ensure your device is claimed by the NMP plugin. Only NMP devices allow you to change the path policy.

```
esxcli <conn_options> storage nmp device list
```

- 2 Retrieve the list of path selection policies on the system to see which values are valid for the `--psp` option when you set the path policy.

```
esxcli storage core plugin registration list --plugin-class="PSP"
```

- 3 Set the path policy using `esxcli`.

```
esxcli <conn_options> storage nmp device set --device naa.xxx --psp VMW_PSP_RR
```

See [Table 4-1, “Supported Path Policies,”](#) on page 48.

- 4 (Optional) If you specified the VMW\_PSP\_FIXED policy, you must make sure the preferred path is set correctly.

- a Check which path is the preferred path for a device.

```
esxcli <conn_options> storage nmp psp fixed deviceconfig get --device naa.xxx
```

- b If necessary, change the preferred path.

```
esxcli <conn_options> storage nmp psp fixed deviceconfig set --device naa.xxx --path
vmhba3:C0:T5:L3
```

The command sets the preferred path to `vmhba3:C0:T5:L3`. Run the command with `--default` to clear the preferred path selection.

### Changing Path Policies with vicfg-mpath

You can change the path policy with `vicfg-mpath`. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

#### To change the path policy with vicfg-mpath

- 1 List all multipathing plugins loaded into the system.

```
vicfg-mpath <conn_options> --list-plugins
```

At a minimum, this command returns NMP (Native Multipathing Plugin) and MASK\_PATH. If other MPP plugins have been loaded, they are listed as well.

- 2 Set the path policy by using ESXCLI.

```
esxcli <conn_options> nmp device set --device naa.xxx --psp VMW_PSP_RR
```

See [Table 4-1, “Supported Path Policies,”](#) on page 48.

- 3 (Optional) If you specified the VMW\_PSP\_FIXED policy, you must make sure the preferred path is set correctly.

- a First check which path is the preferred path for a device.

```
esxcli <conn_options> storage nmp psp fixed deviceconfig get -d naa.xxxx
```

- b If necessary, change the preferred path.

```
esxcli <conn_options> storage nmp psp fixed deviceconfig set --device naa.xxx --path
vmhba3:C0:T5:L3
```

The command sets the preferred path to vmhba3:C0:T5:L3

## Setting Policy Details for Devices that Use Round Robin

ESXi hosts can use multipathing for failover. With certain storage devices, ESXi hosts can also use multipathing for load balancing. To achieve better load balancing across paths, administrators can specify that the ESXi host should switch paths under certain circumstances. Different settable options determine when the ESXi host switches paths and what paths are chosen. Only a limited number of storage arrays support round robin.

You can use `esxcli nmp roundrobin` to retrieve and set round robin path options on a device controlled by the roundrobin PSP. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

No `vicfg-` command exists for performing the operations. The ESXCLI commands for setting round robin path options have changed. The commands supported in ESXi 4.x are no longer supported.

### To view and manipulate round robin path selection settings with ESXCLI

- 1 Retrieve path selection settings for a device that is using the roundrobin PSP.

```
esxcli <conn_options> storage nmp psp roundrobin deviceconfig get --device na.xxx
```

- 2 Set the path selection. You can specify when the path should change, and whether unoptimized paths should be included.

- Use `--bytes` or `--iops` to specify when the path should change, as in the following examples:

```
esxcli <conn_options> storage nmp psp roundrobin deviceconfig set --type "bytes" -B 12345
--device naa.xxx
```

Sets the device specified by `--device` to switch to the next path each time 12345 bytes have been sent along the current path.

```
esxcli <conn_options> storage nmp psp roundrobin deviceconfig set --type=iops --iops
4200 --device naa.xxx
```

Sets the device specified by `--device` to switch after 4200 I/O operations have been performed on a path.

- Use `useano` to specify that the round robin PSP should include paths in the active, unoptimized state in the round robin set (1) or that the PSP should use active, unoptimized paths only if no active optimized paths are available (0). If you do not include this option, the PSP includes only active optimized paths in the round robin path set.

## Scheduling Queues for Virtual Machine I/O

By default, vSphere provides a mechanism that creates scheduling queues for each virtual machine file. Each file has individual bandwidth controls. This mechanism ensures that the I/O for a particular virtual machine goes into its own separate queue and does not interfere with the I/O of other virtual machines.

This capability is enabled by default. You can turn it off by using the `esxcli system settings kernel set -s isPerFileSchedModelActive` option.

**To enable or disable per file I/O scheduling**

- Run `esxcli system settings kernel set -s isPerFileSchedModelActive -v FALSE` to disable per file scheduling.
- Run `esxcli system settings kernel set -s isPerFileSchedModelActive -v TRUE` to enable per file scheduling.

## Managing NFS/NAS Datastores

ESXi hosts can access a designated NFS volume located on a NAS (Network Attached Storage) server, can mount the volume, and can use it for its storage needs. You can use NFS volumes to store and boot virtual machines in the same way that you use VMFS datastores.

### Capabilities Supported by NFS/NAS

An NFS client built into the ESXi hypervisor 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 does not need 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 also 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 hosts support the following shared storage capabilities on NFS volumes:

- VMware vMotion and Storage vMotion
- High Availability (HA), Fault Tolerance, and Distributed Resource Scheduler (DRS)
- ISO images, which are presented as CD-ROMs to virtual machines
- Virtual machine snapshots
- Host profiles
- 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. See “Hardware Acceleration on NAS Devices” in the vSphere Storage documentation.

In addition to storing virtual disks on NFS datastores, you can also use NFS as a central repository for ISO images, virtual machine templates, and so on.

To use NFS as a shared repository, you create a directory on the NFS server and then mount the directory as a datastore on all hosts. 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.

### Adding and Deleting NAS File Systems

You can list, add, and delete a NAS file system with ESXCLI or with `vicfg-nas`.

#### Managing NAS File Systems with ESXCLI

You can use ESXCLI as a vCLI command with connection options (see “[Connection Options for vCLI Host Management Commands](#)” on page 18) or in the ESXi shell.

**To manage a NAS file system**

- 1 List all known NAS file systems.

```
esxcli <conn_options> storage nfs list
```

For each NAS file system, the command lists the mount name, share name, and host name and whether the file system is mounted.

If no NAS file systems are available, the system does not return a NAS filesystem and returns to the command prompt.

- 2 Add a new NAS file system to the ESXi host. Specify the NAS server with `--host`, the volume to use for the mount with `--volume-name`, and the share name on the remote system to use for this NAS mount point with `--share`.

```
esxcli <conn_options> storage nfs add --host=dir42.eng.vmware.com --share=/<mount_dir>
--volume-name=nfsstore-dir42
```

This command adds an entry to the known NAS file system list and supplies the share name of the new NAS file system. You must supply the host name, share name, and volume name for the new NAS file system.

- 3 Add a second NAS file system with read-only access.

```
esxcli <conn_options> storage nfs add --host=dir42.eng.vmware.com --share=/home
--volume-name=FileServerHome2 --readonly
```

- 4 Delete one of the NAS file systems.

```
esxcli <conn_options> storage nfs remove --volume-name=FileServerHome2
```

This command unmounts the NAS file system and removes it from the list of known file systems.

**Managing NAS File Systems with vicfg-nas**

You can use `vicfg-nas` as a vCLI command with connection options. See [“Connection Options for vCLI Host Management Commands”](#) on page 18.

**To manage a NAS file system**

- 1 List all known NAS file systems.

```
vicfg-nas <conn_options> -l
```

For each NAS file system, the command lists the mount name, share name, and host name and whether the file system is mounted. If no NAS file systems are available, the system returns the following message:

```
No NAS datastore found
```

- 2 Add a new NAS file system to the ESXi host.

```
vicfg-nas <conn_options --add --nasserver dir42.eng.vmware.com -s /<mount_dir>
nfsstore-dir42
```

This command adds an entry to the known NAS file system list and supplies the share name of the new NAS file system. You must supply the host name and the share name for the new NAS file system.

- 3 Add a second NAS file system with read-only access.

```
vicfg-nas <conn_options> -a -y --n esx42nas2 -s /home FileServerHome2
```

- 4 Delete one of the NAS file systems.

```
vicfg-nas <conn_options> -d FileServerHome1
```

This command unmounts the NAS file system and removes it from the list of known file systems.

## Monitoring and Managing SAN Storage

The `esxcli storage san` commands help administrators troubleshoot issues with I/O devices and fabric, and include Fibre Channel, FCoE, iSCSI, SAS protocol statistics. The commands allow you to retrieve device information and I/O statistics from those device. You can also issue Loop Initialization Primitives (LIP) to FC/FCoE devices and you can reset SAS devices.

For FC and FCoE devices, you can retrieve FC events such as RSCN, LINKUP, LINKDOWN, Frame Drop and FCoE CVL. The commands log a warning in the VMkernel log if it encounters too many Link Toggling or frame drops

The following example examines and resets SAN storage through a FibreChannel adapter. Instead of `fc`, the information retrieval commands can also use `iscsi`, `fcoe`, and `sas`.

### To monitor and manage FibreChannel SAN storage

- 1 List adapter attributes.  
`esxcli storage san fc list`
- 2 Retrieve all events for a Fibre Channel I/O device.  
`esxcli storage san fc events get`
- 3 Clear all I/O Device Management events for the specified adapter.  
`esxcli storage san fc events clear --adapter adapter`
- 4 Reset the adapter  
`esxcli storage san fc reset`

## Monitoring and Managing Virtual SAN Storage

Virtual SAN is a distributed layer of software that runs natively as a part of the ESXi hypervisor. Virtual SAN aggregates local or direct-attached storage disks of a host cluster and creates a single storage pool shared across all hosts of the cluster.

While supporting VMware features that require shared storage, such as HA, vMotion, and DRS, Virtual SAN eliminates the need for an external shared storage and simplifies storage configuration and virtual machine provisioning activities.

You can use ESXCLI commands to retrieve virtual SAN information, manage Virtual SAN clusters, perform network management, add storage, set the policy, and perform other monitoring and management tasks. Type `esxcli vsan --help` for a complete list of commands.

### To retrieve Virtual SAN information

- 1 Verify which VMkernel adapters are used for Virtual SAN communication.  
`esxcli vsan network list`
- 2 List storage disks that were claimed by Virtual SAN.  
`esxcli vsan storage list`
- 3 Get Virtual SAN cluster information.  
`esxcli vsan cluster get`

You can activate Virtual SAN when you create host clusters or enable Virtual SAN on existing clusters. When enabled, Virtual SAN aggregates all local storage disks available on the hosts into a single datastore shared by all hosts. You can later expand the datastore by adding storage devices or hosts to the cluster.

You can run these commands in the ESXi Shell for a host, or the command affects the target host that you specify as part of the vCLI connection options.

### To manage a Virtual SAN cluster

- 1 Join the target host to a given Virtual SAN cluster.  

```
esxcli vsan cluster join --cluster-uuid <uuid>
```

The UUID of the cluster is required.
- 2 Verify that the target host is joined to a Virtual SAN cluster.  

```
esxcli vsan cluster get
```
- 3 Remove the target host from the Virtual SAN cluster.  

```
esxcli vsan cluster leave
```

### To add and remove Virtual SAN storage

- 1 Add an hdd (or data disk) for use by Virtual SAN.  

```
esxcli vsan storage add --disks <device_name>
```

The command expects an empty disk, which will be partitioned or formatted. Specify a device name, for example, `mpx.vmhba2:C0:T1:L0`.
- 2 Add an sdd disk for use by Virtual SAN.  

```
esxcli vsan storage add --ssd <device_name>
```

The command expects an empty disk, which will be partitioned or formatted. Specify a device name, for example, `mpx.vmhba2:C0:T1:L0`.
- 3 List the Virtual SAN storage configuration. You can display the complete list, or filter to show only a single device.  

```
esxcli vsan storage list --device <device>
```
- 4 Remove disks or disk groups.  

You can remove disks or disk groups only when Virtual SAN is in manual mode. For the automatic disk claim mode, the remove action is not supported

  - Remove an individual Virtual SAN disk.  

```
esxcli vsan storage remove --disk <device_name>
```

Instead of specifying the device name, you can specify the UUID if you include the `--uuid` option.
  - Remove a disk group's SSD and each of its backing HDD drives from Virtual SAN usage.  

```
esxcli vsan storage remove --ssd <device_name>
```

Instead of specifying the device name, you can specify the UUID if you include the `--uuid` option. Any SSD that you remove from Virtual SAN becomes available for such features as Flash Read Cache.

## Monitoring vSphere Flash Read Cache

Flash Read Cache™ lets you accelerate virtual machine performance through the use of host resident flash devices as a cache. The *vSphere Storage* documentation discusses vSphere Flash Read Cache in some detail.

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.

---

**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

---

You can manage vSphere Flash Read Cache from the vSphere Web Client. You can monitor Flash Read Cache by using commands in the `esxcli storage vflash` namespace. The following table lists available commands. See the *vSphere Command-Line Interface Reference* or the online help for a list of options to each command.

**Table 4-3.** Commands for Monitoring vSphere Flash Read Cache

| Command                                       | Description                       |
|-----------------------------------------------|-----------------------------------|
| <code>storage vflash cache get</code>         | Get individual vflash cache info. |
| <code>storage vflash cache list</code>        | List individual vflash caches.    |
| <code>storage vflash cache stats get</code>   | Get vflash cache statistics.      |
| <code>storage vflash cache stats reset</code> | Reset vflash cache statistics.    |
| <code>storage vflash device list</code>       | List vflash SSD devices.          |
| <code>storage vflash module get</code>        | Get vflash module info.           |
| <code>storage vflash module list</code>       | List vflash modules.              |
| <code>storage vflash module stats get</code>  | Get vflash module statistics      |

## Monitoring and Managing 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. The vSphere Storage documentation discusses Virtual Volumes in some detail and explains how to manage them using the vSphere Web Client.

The following `esxcli` commands are available for managing display information about virtual volumes and for unbinding all Virtual Volumes from all vendor providers. See the *vSphere Storage* documentation for information on creating Virtual Volumes and configuring multipathing and SCSI-based endpoints.

**Table 4-4.** VVol Commands

| Command                                            | Description                                                                                     |
|----------------------------------------------------|-------------------------------------------------------------------------------------------------|
| <code>storage vvol daemon unbindall</code>         | Unbind all Virtual Volume instances from all storage providers that are known to the ESXi host. |
| <code>storage vvol protocolendpoint list</code>    | List the VVol protocol endpoints currently known to the ESXi host.                              |
| <code>storage vvol storagecontainer list</code>    | List the VVol storage containers currently known to the ESX host.                               |
| <code>storage vvol storagecontainer restore</code> | Restore storage containers of vendor providers that are registered on the host.                 |
| <code>storage vvol vasacontext get</code>          | Get the VASA context (VC UUID).                                                                 |
| <code>storage vvol vendorprovider list</code>      | List the vendor providers registered on the host.                                               |
| <code>storage vvol vendorprovider restore</code>   | Restore the vendor providers that are registered on the host.                                   |

## Migrating Virtual Machines with svmotion

Storage vMotion moves a virtual machine's configuration file, and, optionally, its disks, while the virtual machine is running. You can perform Storage vMotion tasks from the vSphere Web Client or with the `svmotion` command.

---

**IMPORTANT** No ESXCLI command for Storage vMotion is available.

---

You can place the virtual machine and all of its disks in a single location, or choose separate locations for the virtual machine configuration file and each virtual disk. You cannot change the virtual machine's execution host during a migration with `svmotion`.

### Storage vMotion Uses

Storage vMotion has several uses in administering your vSphere environment.

- Upgrade ESXi without virtual machine downtime in situations where virtual machine disks must be moved to shared storage to allow migration with vMotion.
- Perform storage maintenance and reconfiguration. You can use Storage vMotion to move virtual machines off a storage device to allow maintenance or reconfiguration of the storage device without virtual machine downtime.
- Redistribute storage load. You can use Storage vMotion to manually redistribute virtual machines or virtual disks to different storage volumes to balance capacity or improve performance.

### Storage vMotion Requirements and Limitations

You can migrate virtual machine disks with Storage vMotion if the virtual machine and its host meet the following resource and configuration requirements:

- For ESXi 5.0 and later hosts, you can migrate virtual machines that have snapshots. For earlier versions of ESXi, you cannot migrate virtual machines that have snapshots.
- Virtual machine disks must be in persistent mode or be raw device mappings (RDMs). For physical and virtual compatibility mode RDMs, you can migrate the mapping file only. For virtual compatibility mode RDMs, you can use the vSphere Web Client to convert to thick-provisioned or thin-provisioned disks during migration as long as the destination is not an NFS datastore. You cannot use the `svmotion` command to perform this conversion.
- The host on which the virtual machine is running must have a license that includes Storage vMotion.
- The host on which the virtual machine is running must have access to both the source and target datastores.
- A particular host can be involved in up to four migrations with vMotion or Storage vMotion at one time. See "Limits on Simultaneous Migrations" in the *vCenter Server and Host Management* documentation for details.

If you use the vSphere Web Client for migration with `svmotion`, the system performs several compatibility checks. These checks are not supported by the `svmotion` vCLI command.

### Running svmotion in Interactive Mode

You can run `svmotion` in interactive mode using the `--interactive` option. The command prompts you for the information it needs to complete the storage migration.

```
svmotion <conn_options> --interactive
```

When you use `--interactive`, all other options are ignored.

---

**IMPORTANT** When responding to the prompts, use quotes around input strings with special characters.

---

## Running svmotion in Noninteractive Mode

---

**IMPORTANT** When you run `svmotion`, `--server` must point to a vCenter Server system.

---

In noninteractive mode, the `svmotion` command uses the following syntax:

```
svmotion [standard vCLI options] --datacenter=<datacenter_name>
  --vm <VM config datastore path>:<new datastore>
  [--disks <virtual disk datastore path>:<new datastore>,
  <virtual disk datastore path>:<new datastore>]
```

Square brackets indicate optional elements, not datastores.

The `--vm` option specifies the virtual machine and its destination. By default, all virtual disks are relocated to the same datastore as the virtual machine. This option requires the current virtual machine configuration file location. See [“To determine the path to the virtual machine configuration file and disk file”](#) on page 57.

The `--disks` option relocates individual virtual disks to different datastores. The `--disks` option requires the current virtual disk datastore path as an option. See [“To determine the path to the virtual machine configuration file and disk file”](#) on page 57.

### To determine the path to the virtual machine configuration file and disk file

- 1 Run `vmware-cmd -l` to list all virtual machine configuration files (VMX files).

```
vmware-cmd -H <vc_server> -U <login_user> -P <login_password> -h <esx_host> -l
```

- 2 Choose the VMX file for the virtual machine of interest.

By default, the virtual disk file has the same name as the VMX file but has a `.vmdk` extension.

- 3 (Optional) Use `vifs` to verify that you are using the correct VMDK file.

### To relocate a virtual machine’s storage (including disks)

- 1 Determine the path to the virtual machine configuration file.

- 2 Run `svmotion`:

```
svmotion
--url=https://myvc.mycorp.com/sdk --datacenter=DC1
--vm="[storage1] myvm/myvm.vmx:new_datastore"
```

The example is for Windows. Use single quotes on Linux.

### To relocate a virtual machine’s configuration file, but leave virtual disks

- 1 Determine the path to the virtual disk files and the virtual machine configuration file.

- 2 Run `svmotion`, for example:

```
svmotion
<conn_options>
--datacenter='My DC'
--vm='[old_datastore] myvm/myvm.vmx:new_datastore'
--disks='[old_datastore] myvm/myvm_1.vmdk:old_datastore, [old_datastore] myvm/myvm_2.vmdk:old_datastore'
```

This command relocates the virtual machine's configuration file to `new_datastore`, but leaves the two disks (`myvm_1.vmdk` and `myvm_2.vmdk`) in `old_datastore`. The example is for Linux. Use double quotes on Windows. The square brackets surround the datastore name and do not indicate an optional element.

## Configuring FCoE Adapters

ESXi can use Fibre Channel over Ethernet (FCoE) adapters to access Fibre Channel storage.

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 10 Gbit lossless Ethernet to deliver Fibre Channel traffic.

To use FCoE, you need to install FCoE adapters. The adapters that VMware supports generally fall into two categories, hardware FCoE adapters and software FCoE adapters.

- **Hardware FCoE Adapters.** Hardware FCoE adapters include completely offloaded specialized Converged Network Adapters (CNAs) that contain network and Fibre Channel functionalities on the same card. When such an adapter is installed, your host detects and can use both CNA components. In the vSphere Web Client, the networking component appears as a standard network adapter (vmnic) and the Fibre Channel component as a FCoE adapter (vmhba). You do not have to configure a hardware FCoE adapter to be able to use it.
- **Software FCoE Adapters.** A software FCoE adapter is a software code that performs some of the FCoE processing. The adapter can be used with a number of NICs that support partial FCoE offload. Unlike the hardware FCoE adapter, the software adapter must be activated.

## Scanning Storage Adapters

You must perform a rescan operation each time you reconfigure your storage setup. You can scan using the vSphere Web Client, the `vicfg-rescan` vCLI command, or the `esxcli storage core adapter rescan` command.

- `esxcli storage core adapter rescan` supports the following additional options:
  - `-a|--all` or `-A|--adapter=<string>` – Scan all adapters or a specified adapter.
  - `-S|--skip-claim` – Skip claiming of new devices by the appropriate multipath plugin.
  - `-F|--skip-fs-scan` – Skip filesystem scan
  - `-t|--type` – Specify the type of scan to perform. The command either scans for all changes (`all`) or for added, deleted, or updated adapters (`add`, `delete`, `update`)
- `vicfg-rescan` supports only a simple rescan operation on a specified adapter.

### To rescan a storage adapter with `vicfg-rescan`

Run `vicfg-rescan`, specifying the adapter name.

```
vicfg-rescan <conn_options> vmhba1
```

The command returns an indication of success or failure, but no detailed information.

### To rescan a storage adapter with `ESXCLI`

The following command scans a specific adapter and skips the filesystem scan that is performed by default.

```
esxcli <conn_options> storage core adapter rescan --adapter=vmhba33 --skip-claim
```

The command returns an indication of success or failure, but no detailed information.

## Retrieving SMART Information

You can use `ESXCLI` to retrieve information related to SMART. SMART is a monitoring system for computer hard disks that reports information about the disks.

```
esxcli storage core device smart get -d device
```

What the command returns depends on the level of SMART information that the device supports. If no information is available for a parameter, the output displays N/A, as in the following sample output.

| Parameter                | Value | Threshold | Worst |
|--------------------------|-------|-----------|-------|
| Health Status            | OK    | N/A       | N/A   |
| Media Wearout Indicator  | N/A   | N/A       | N/A   |
| Write Error Count        | N/A   | N/A       | N/A   |
| Read Error Count         | 119   | 6         | 74    |
| Power-on Hours           | 57    | 0         | 57    |
| Power Cycle Count        | 100   | 20        | 100   |
| Reallocated Sector Count | 100   | 36        | 100   |

|                              |     |     |     |
|------------------------------|-----|-----|-----|
| Raw Read Error Rate          | 119 | 6   | 74  |
| Drive Temperature            | 38  | 0   | 49  |
| Driver Rated Max Temperature | 62  | 45  | 51  |
| Write Sectors TOT Count      | 200 | 0   | 200 |
| Read Sectors TOT Count       | 100 | 0   | 253 |
| Initial Bad Block Count      | N/A | N/A | N/A |



# Managing iSCSI Storage

---

ESXi systems include iSCSI technology to access remote storage using an IP network. You can use the vSphere Web Client, commands in the `esxcli iscsi` namespace, or the `vicfg-iscsi` command to configure both hardware and software iSCSI storage for your ESXi system.

This chapter includes the following topics:

- [“iSCSI Storage Overview”](#) on page 59
- [“Protecting an iSCSI SAN”](#) on page 61
- [“Command Syntax for esxcli iscsi and vicfg-iscsi”](#) on page 63
- [“iSCSI Storage Setup with ESXCLI”](#) on page 68
- [“iSCSI Storage Setup with vicfg-iscsi”](#) on page 73
- [“Listing and Setting iSCSI Options”](#) on page 77
- [“Listing and Setting iSCSI Parameters”](#) on page 78
- [“Enabling iSCSI Authentication”](#) on page 82
- [“Setting Up Ports for iSCSI Multipathing”](#) on page 83
- [“Managing iSCSI Sessions”](#) on page 84

See the *vSphere Storage* documentation for additional information.

## iSCSI Storage Overview

With iSCSI, SCSI storage commands that your virtual machine issues to its virtual disk are converted into TCP/IP protocol packets and transmitted to a remote device, or target, on which the virtual disk is located. To the virtual machine, the device appears as a locally attached SCSI drive.

To access remote targets, the ESXi host uses iSCSI initiators. Initiators transport SCSI requests and responses between ESXi and the target storage device on the IP network. ESXi supports these types of initiators:

- **Software iSCSI adapter.** VMware code built into the VMkernel. Allows an ESXi host to connect to the iSCSI storage device through standard network adapters. The software initiator handles iSCSI processing while communicating with the network adapter.
- **Hardware iSCSI adapter.** Offloads all iSCSI and network processing from your host. Hardware iSCSI adapters are broken into two types.
  - **Dependent hardware iSCSI adapter.** Leverages the VMware iSCSI management and configuration interfaces.
  - **Independent hardware iSCSI adapter.** Leverages its own iSCSI management and configuration interfaces.

See the *vSphere Storage* documentation for details on setup and failover scenarios.

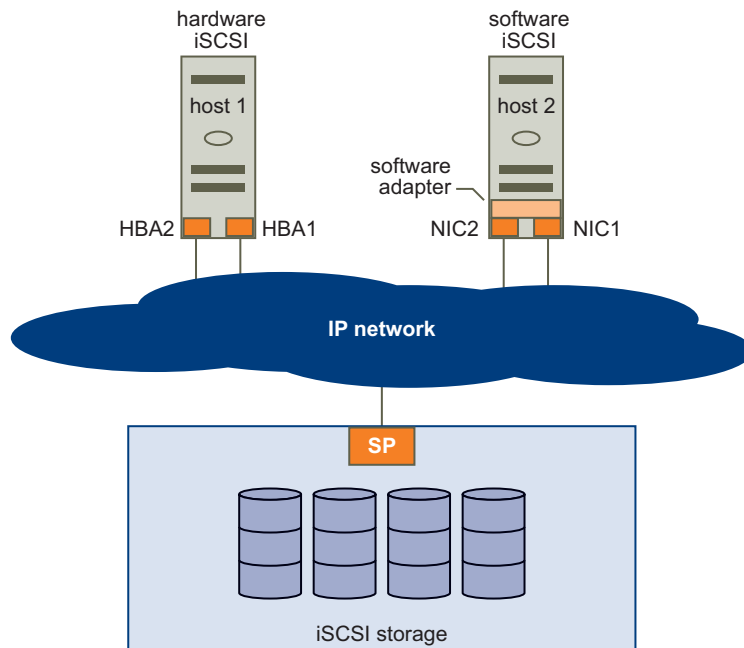
You must configure iSCSI initiators for the host to access and display iSCSI storage devices.

Figure 5-1 depicts hosts that use different types of iSCSI initiators.

- The host on the left uses an independent hardware iSCSI adapter to connect to the iSCSI storage system.
- The host on the right uses software iSCSI.

Dependent hardware iSCSI can be implemented in different ways and is not shown. 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.

**Figure 5-1.** iSCSI Storage



## Discovery Sessions

A discovery session is part of the iSCSI protocol. The discovery session returns the set of targets that you can access on an iSCSI storage system. ESXi systems support dynamic and static discovery.

- **Dynamic discovery.** Also known as Send Targets discovery. Each time the ESXi host contacts a specified iSCSI storage server, it sends a Send Targets request to the server. In response, the iSCSI storage server supplies a list of available targets to the ESXi host. Monitor and manage with `esxcli iscsi adapter discovery sendtarget` or `vicfg-iscsi` commands.
- **Static discovery.** The ESXi host does not have to perform discovery. Instead, the ESXi host uses the IP addresses or domain names and iSCSI target names (IQN or EUI format names) to communicate with the iSCSI target. Monitor and manage with `esxcli iscsi adapter discovery statictarget` or `vicfg-iscsi` commands.

For either case, you set up target discovery addresses so that the initiator can determine which storage resource on the network is available for access. You can do this setup with dynamic discovery or static discovery. With dynamic discovery, all targets associated with an IP address or host name and the iSCSI name are discovered. With static discovery, you must specify the IP address or host name and the iSCSI name of the target you want to access. The iSCSI HBA must be in the same VLAN as both ports of the iSCSI array.



## Discovery Target Names

The target name is either an IQN name or an EUI name.

- The IQN name uses the following format:

```
iqn.yyyy-mm.{reversed domain name}:id_string
```

For example: `iqn.2007-05.com.mydomain:storage.tape.sys3.abc`

The ESXi host generates an IQN name for software iSCSI and dependent hardware iSCSI adapters. You can change that default IQN name.

- The EUI name is described in IETF rfc3720 as follows:

The IEEE Registration Authority provides a service for assigning globally unique identifiers [EUI]. The EUI-64 format is used to build a global identifier in other network protocols. For example, Fibre Channel defines a method of encoding it into a `WorldWideName`.

The format is `eui.` followed by an EUI-64 identifier (16 ASCII-encoded hexadecimal digits).

For example:

```
Type  EUI-64 identifier (ASCII-encoded hexadecimal)
+---+-----+
|  |         |
eui.02004567A425678D
```

The IEEE EUI-64 iSCSI name format can be used when a manufacturer is registered with the IEEE Registration Authority and uses EUI-64 formatted worldwide unique names for its products.

Check in the UI of the storage array whether an array uses an IQN name or an EUI name.

## Protecting an iSCSI SAN

Your iSCSI configuration is only as secure as your IP network. By enforcing good security standards when you set up your network, you help safeguard your iSCSI storage.

### Protecting Transmitted Data

A primary security risk in iSCSI SANs is that an attacker might sniff transmitted storage data. Neither the iSCSI adapter nor the ESXi host iSCSI initiator encrypts the data that it transmits to and from the targets, making the data vulnerable to sniffing attacks. You must therefore take additional measures to prevent attackers from easily seeing iSCSI data.

Allowing your virtual machines to share virtual switches and VLANs with your iSCSI configuration potentially exposes iSCSI traffic to misuse by a virtual machine attacker. To help ensure that intruders cannot listen to iSCSI transmissions, make sure that none of your virtual machines can see the iSCSI storage network.

Protect your system by giving the iSCSI SAN a dedicated virtual switch.

- If you use an independent hardware iSCSI adapter, make sure that the iSCSI adapter and ESXi physical network adapter are not inadvertently connected outside the host. Such a connection might result from sharing a switch.
- If you use dependent hardware or software iscsi adapter, which uses ESXi networking, configure iSCSI storage through a different virtual switch than the one used by your virtual machines.

You can also configure your iSCSI SAN on its own VLAN to improve performance and security. Placing your iSCSI configuration on a separate VLAN ensures that no devices other than the iSCSI adapter can see transmissions within the iSCSI SAN. With a dedicated VLAN, network congestion from other sources cannot interfere with iSCSI traffic.

## Securing iSCSI Ports

When you run iSCSI devices, the ESXi host does not open ports that listen for network connections. This measure reduces the chances that an intruder can break into the ESXi host through spare ports and gain control over the host. Therefore, running iSCSI does not present an additional security risks at the ESXi host end of the connection.

An iSCSI target device must have one or more open TCP ports to listen for iSCSI connections. If security vulnerabilities exist in the iSCSI device software, your data can be at risk through no fault of the ESXi system. To lower this risk, install all security patches that your storage equipment manufacturer provides and limit the devices connected to the iSCSI network.

## Setting iSCSI CHAP

iSCSI storage systems authenticate an initiator using a name and key pair. ESXi systems support Challenge Handshake Authentication Protocol (CHAP), which VMware recommends for your SAN implementation. The ESXi host and the iSCSI storage system must have CHAP enabled and must have common credentials. During iSCSI login, the iSCSI storage system exchanges its credentials with the ESXi system and checks them.

You can set up iSCSI authentication by using the vSphere Web Client, as discussed in the *vSphere Storage* documentation or by using the `esxcli` command, discussed in “[Enabling iSCSI Authentication](#)” on page 82. To use CHAP authentication, you must enable CHAP on both the initiator side and the storage system side. After authentication is enabled, it applies for targets to which no connection has been established, but does not apply to targets to which a connection is established. After the discovery address is set, the new volumes to which you add a connection are exposed and can be used.

For software iSCSI and dependent hardware iSCSI, ESXi hosts support per-discovery and per-target CHAP credentials. For independent hardware iSCSI, ESXi hosts support only one set of CHAP credentials per initiator. You cannot assign different CHAP credentials for different targets.

When you configure independent hardware iSCSI initiators, ensure that the CHAP configuration matches your iSCSI storage. If CHAP is enabled on the storage array, it must be enabled on the initiator. If CHAP is enabled, you must set up the CHAP authentication credentials on the ESXi host to match the credentials on the iSCSI storage.

### Supported CHAP Levels

To set CHAP levels with `esxcli iscsi adapter setauth` or `vicfg-iscsi`, specify one of the values in [Table 5-1](#) for `<level>`. Only two levels are supported for independent hardware iSCSI.

Mutual CHAP is supported for software iSCSI and for dependent hardware iSCSI, but not for independent hardware iSCSI.

---

**IMPORTANT** Ensure that CHAP is set to `chapRequired` before you set mutual CHAP, and use compatible levels for CHAP and mutual CHAP. Use different passwords for CHAP and mutual CHAP to avoid security risks.

---

**Table 5-1.** Supported Levels for CHAP

| Level                        | Description                                                                                                             | Supported                                                                |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| <code>chapProhibited</code>  | Host does not use CHAP authentication. If authentication is enabled, specify <code>chapProhibited</code> to disable it. | Software iSCSI<br>Dependent hardware iSCSI<br>Independent hardware iSCSI |
| <code>chapDiscouraged</code> | Host uses a non-CHAP connection, but allows a CHAP connection as fallback.                                              | Software iSCSI<br>Dependent hardware iSCSI                               |
| <code>chapPreferred</code>   | Host uses CHAP if the CHAP connection succeeds, but uses non-CHAP connections as fallback.                              | Software iSCSI<br>Dependent hardware iSCSI<br>Independent hardware iSCSI |
| <code>chapRequired</code>    | Host requires successful CHAP authentication. The connection fails if CHAP negotiation fails.                           | Software iSCSI<br>Dependent hardware iSCSI                               |

## Returning Authentication to Default Inheritance

The values of iSCSI authentication settings associated with a dynamic discovery address or a static discovery target are inherited from the corresponding settings of the parent. For the dynamic discovery address, the parent is the adapter. For the static target, the parent is the adapter or discovery address.

- If you use the vSphere Web Client to modify authentication settings, you must deselect the **Inherit from Parent** check box before you can make a change to the discovery address or discovery target.
- If you use `vicfg-iscsi`, the value you set overrides the inherited value.
- If you use `esxcli iscsi` commands, the value you set overrides the inherited value. You can set CHAP at these levels:
  - `esxcli iscsi adapter auth chap [get|set]`
  - `esxcli iscsi adapter discovery sendtarget auth chap [get|set]`
  - `esxcli iscsi adapter target portal auth chap [get|set]`

Inheritance is relevant only if you want to return a dynamic discovery address or a static discovery target to its inherited value. In that case, use one of the following commands:

- Dynamic discovery: `esxcli iscsi adapter discovery sendtarget auth chap set --inherit`
- Static discovery: `esxcli iscsi adapter target portal auth chap set --inherit`.

---

**NOTE** You can set target-level CHAP authentication properties to be inherited from the send target level and set send target level CHAP authentication properties to be inherited from the adapter level. Resetting adapter-level properties is not supported.

---

## Command Syntax for `esxcli iscsi` and `vicfg-iscsi`

In vSphere 5.0 and later, you can manage iSCSI storage by using either `esxcli iscsi` commands or `vicfg-iscsi` options. See the *vSphere Command-Line Interface Reference*. “[esxcli iscsi Command Syntax](#)” on page 63 and “[vicfg-iscsi Command Syntax](#)” on page 65 provide an overview.

### esxcli iscsi Command Syntax

The `esxcli iscsi` command includes a number of nested namespaces. The following table illustrates the namespace hierarchy. Commands at each level are included in bold. Many namespaces include both commands and namespaces.

**Table 5-2.** `esxcli iscsi` Command Overview

|                               |                               |                                       |                        |                       |
|-------------------------------|-------------------------------|---------------------------------------|------------------------|-----------------------|
| adapter <b>[get list set]</b> | auth                          | chap <b>[set get]</b>                 |                        |                       |
|                               | discovery <b>[rediscover]</b> | sendtarget <b>[add list remove]</b>   |                        |                       |
|                               |                               |                                       | auth                   | chap <b>[get set]</b> |
|                               |                               |                                       | param <b>[get set]</b> |                       |
|                               |                               | statictarget <b>[add list remove]</b> |                        |                       |
|                               |                               | status <b>get</b>                     |                        |                       |
|                               | target <b>[list]</b>          | portal <b>[list]</b>                  | auth                   | chap <b>[get set]</b> |
|                               |                               |                                       | param <b>[get set]</b> |                       |
|                               | capabilities <b>get</b>       |                                       |                        |                       |
|                               | firmware <b>[get set]</b>     |                                       |                        |                       |
|                               | param <b>[get set]</b>        |                                       |                        |                       |

**Table 5-2.** esxcli iscsi Command Overview

|                                    |                    |
|------------------------------------|--------------------|
| networkportal<br>[add list remove] | ipconfig [get set] |
| physicalnetworkportal<br>[list]    | param [get set]    |
| session [add list remove]          | connection list    |
| ibftboot [get import]              |                    |
| logicalnetworkportal list          |                    |
| plugin list                        |                    |
| software [get set]                 |                    |

## Key to esxcli iscsi Short Options

ESXCLI commands for iSCSI management consistently use the same short options. For several options, the associated full option depends on the command.

**Table 5-3.** Short Options for iSCSI ESXCLI Command Options

| Lower-case Option | Option                    | Upper-case Option | Option                | Number | Option |
|-------------------|---------------------------|-------------------|-----------------------|--------|--------|
| a                 | --address, alias          | A                 | --adapter             | 1      | --dns1 |
| c                 | --cid                     |                   |                       | 2      | --dns2 |
| d                 | --direction               | D                 | --default             |        |        |
| f                 | --file, force             |                   |                       |        |        |
| g                 | --gateway                 |                   |                       |        |        |
| i                 | --ip                      | I                 | --inherit             |        |        |
| k                 | --key                     |                   |                       |        |        |
| l                 | --level                   |                   |                       |        |        |
| m                 | --method                  | M                 | --module              |        |        |
| n                 | --nic                     | N                 | --authname,<br>--name |        |        |
| o                 | --option                  |                   |                       |        |        |
| p                 | --plugin                  |                   |                       |        |        |
| s                 | --isid, subnet,<br>switch | S                 | --state, secret       |        |        |
| v                 | --value                   |                   |                       |        |        |

## vicfg-iscsi Command Syntax

vicfg-iscsi supports a comprehensive set of options, listed in [Table 5-4](#).

**Table 5-4.** Options for vicfg-iscsi

| Option              | Suboptions                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Description                                                                                                |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| -A --authentication | <pre>-c &lt;level&gt; -m &lt;auth_method&gt; -b -v &lt;ma_username&gt; -x &lt;ma_password&gt; [-i &lt;stor_ip_addr stor_hostname&gt; [:&lt;portnum&gt;] [-n &lt;iscsi_name&gt;]] &lt;adapter_name&gt;  --level &lt;level&gt; --method &lt;auth_method&gt; --mutual --mchap_username &lt;ma_username&gt; --mchap_password &lt;ma_password&gt; [--ip &lt;stor_ip_addr stor_hostname&gt; [:&lt;portnum&gt;] [--name &lt;iscsi_name&gt;]] &lt;adapter_name&gt;</pre> | Enables mutual authentication. You must enable authentication before you can enable mutual authentication. |
| -A --authentication | <pre>-c &lt;level&gt; -m &lt;auth_method&gt; -u &lt;auth_u_name&gt; -w &lt;a_password&gt; [-i &lt;stor_ip_addr stor_hostname&gt; [:&lt;portnum&gt;] [-n &lt;iscsi_name&gt;]] &lt;adapter_name&gt;  --level &lt;level&gt; --method &lt;auth_method&gt; --chap_password &lt;auth_u_name&gt; --chap_username &lt;chap_password&gt; [--ip &lt;stor_ip_addr stor_hostname&gt; [:&lt;portnum&gt;] [--name &lt;iscsi_name&gt;]] &lt;adapter_name&gt;</pre>              | Enables authentication using the specified options.                                                        |
| -A --authentication | <pre>-l &lt;adapter_name&gt; --list &lt;adapter_name&gt;</pre>                                                                                                                                                                                                                                                                                                                                                                                                   | Lists supported authentication methods.                                                                    |
| -D --discovery      | <pre>-a -i &lt;stor_ip_addr stor_hostname&gt;[:&lt;portnum&gt;] &lt;adapter_name&gt; --add --ip &lt;stor_ip_addr stor_hostname&gt; [:&lt;portnum&gt;] &lt;adapter_name&gt;</pre>                                                                                                                                                                                                                                                                                 | Adds a dynamic discovery address.                                                                          |
| -D --discovery      | <pre>-l &lt;adapter_name&gt; --list &lt;adapter_name&gt;</pre>                                                                                                                                                                                                                                                                                                                                                                                                   | Lists dynamic discovery addresses.                                                                         |
| -D --discovery      | <pre>-r -i &lt;stor_ip_addr stor_hostname&gt;[:&lt;portnum&gt;] &lt;adapter_name&gt; --remove --ip &lt;stor_ip_addr stor_hostname&gt; [:&lt;portnum&gt;] &lt;adapter_name&gt;</pre>                                                                                                                                                                                                                                                                              | Removes a dynamic discovery address.                                                                       |
| -H                  | <pre>-l [&lt;adapter_name&gt;] --list [&lt;adapter_name&gt;]</pre>                                                                                                                                                                                                                                                                                                                                                                                               | Lists all iSCSI adapters or a specified adapter.                                                           |
| -L --lun            | <pre>-l &lt;adapter_name&gt; --list &lt;adapter_name&gt;</pre>                                                                                                                                                                                                                                                                                                                                                                                                   | Lists LUN information.                                                                                     |

**Table 5-4.** Options for vicfg-iscsi (Continued)

| Option                                         | Suboptions                                                                                                                                                                          | Description                                                           |
|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| -L --lun                                       |                                                                                                                                                                                     |                                                                       |
|                                                | -l -t <target_ID> <adapter_name><br>--list --target_id <target_id> <adapter_name>                                                                                                   | Lists LUN information for a specific target.                          |
| -N --network (Independent hardware iSCSI only) |                                                                                                                                                                                     |                                                                       |
|                                                | -l <adapter_name><br>--list <adapter_name>                                                                                                                                          | Lists network properties.                                             |
| -N --network (Independent hardware iSCSI only) |                                                                                                                                                                                     |                                                                       |
|                                                | -i <ip_addr> <adapter_name><br>--ip <ip_addr> <vmhba>                                                                                                                               | Sets the HBA IPv4 address to ip_addr.                                 |
| -N --network (Independent hardware iSCSI only) |                                                                                                                                                                                     |                                                                       |
|                                                | -s <subnet_mask> <adapter_name><br>--subnetmask <subnet_mask> <adapter_name>                                                                                                        | Sets the HBA network mask to subnet_mask.                             |
| -N --network (Independent hardware iSCSI only) |                                                                                                                                                                                     |                                                                       |
|                                                | -g <default_gateway> <adapter_name><br>--gateway <default_gateway> <adapter_name>                                                                                                   | Sets the HBA gateway to default_gateway.                              |
| -N --network (Independent hardware iSCSI only) |                                                                                                                                                                                     |                                                                       |
|                                                | -i <ip_addr> -s <subnet mask><br>-g <default_gateway> <adapter_name><br>--ip <ip_addr> --subnetmask <subnet_mask><br>--gateway <default_gateway> <adapter_name>                     | Sets the IP address, subnet mask, and default gateway in one command. |
| -p --pnp (Independent hardware iSCSI only)     |                                                                                                                                                                                     |                                                                       |
|                                                | -l <adapter_name><br>--list <adapter_name>                                                                                                                                          | Lists physical network portal options.                                |
| -p --pnp (Independent hardware iSCSI only)     |                                                                                                                                                                                     |                                                                       |
|                                                | -M <mtu_size> <adapter_name><br>--mtu <mtu-size> <adapter_name>                                                                                                                     | Sets physical network portal options.                                 |
| -I --iscsiname                                 |                                                                                                                                                                                     |                                                                       |
|                                                | -a <alias_name> <adapter_name><br>--alias <alias_name> <adapter_name>                                                                                                               | Sets the iSCSI initiator alias.                                       |
| -I --iscsiname                                 |                                                                                                                                                                                     |                                                                       |
|                                                | -n <iscsi_name> <adapter_name><br>--name <iscsi_name> <adapter_name>                                                                                                                | Sets the iSCSI initiator name.                                        |
| -I --iscsiname                                 |                                                                                                                                                                                     |                                                                       |
|                                                | -l <adapter_name><br>--list <adapter_name>                                                                                                                                          | Lists iSCSI initiator options.                                        |
| -M --mtu                                       |                                                                                                                                                                                     |                                                                       |
|                                                | -p -M <mtu_size> <adapter_name><br>--pnp --mtu <mtu-size> <adapter_name>                                                                                                            | Sets MTU size. Used with the --pnp option.                            |
| -S --static                                    |                                                                                                                                                                                     |                                                                       |
|                                                | -l <adapter_name><br>--list <adapter_name>                                                                                                                                          | Lists static discovery addresses.                                     |
| -S --static                                    |                                                                                                                                                                                     |                                                                       |
|                                                | -r -i <stor_ip_addr stor_hostname> [:<portnum>] -n <target_name><br><adapter_name><br>--remove --ip <stor_ip_addr stor_hostname> [:<portnum>] -name<br><target_name> <adapter_name> | Removes a static discovery address.                                   |

**Table 5-4.** Options for vicfg-iscsi (Continued)

| Option          | Suboptions                                                                                                                                                                                                                                                                                                                     | Description                                                                                                              |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| -S --static     | <pre>-a -i &lt;stor_ip_addr stor_hostname&gt; [:&lt;portnum&gt;]   -n &lt;target_name&gt; &lt;adapter_name&gt; --add --ip &lt;stor_ip_addr stor_hostname&gt; [:&lt;portnum&gt;]   -name &lt;target_name&gt; &lt;adapter_name&gt;</pre>                                                                                         | Adds a static discovery address.                                                                                         |
| -P --phba       | <pre>-l &lt;adapter_name&gt; --list &lt;adapter_name&gt;</pre>                                                                                                                                                                                                                                                                 | Lists external, vendor-specific properties of an iSCSI adapter.                                                          |
| -T --target     | <pre>-l &lt;adapter_name&gt; --list &lt;adapter_name&gt;</pre>                                                                                                                                                                                                                                                                 | Lists target information.                                                                                                |
| -W --parameter  | <pre>-l [-i &lt;stor_ip_addr stor_hostname&gt; [:&lt;portnum&gt;]   [-n &lt;iscsi_name&gt;]] &lt;adapter_name&gt;  --list [--ip &lt;stor_ip_addr stor_hostname&gt; [:&lt;portnum&gt;]   [--name &lt;iscsi_name&gt;]] &lt;adapter_name&gt;</pre>                                                                                | Lists iSCSI parameter information.                                                                                       |
| -W --parameter  | <pre>-l -k [-i &lt;stor_ip_addr stor_hostname&gt; [:&lt;portnum&gt;] [-n &lt;iscsi_name&gt;]] &lt;adapter_name&gt;  --list --detail   [--ip &lt;stor_ip_addr stor_hostname&gt; [:&lt;portnum&gt;]   [--name &lt;iscsi_name&gt;]] &lt;adapter_name&gt;</pre>                                                                    | Lists iSCSI parameter details.                                                                                           |
| -W --parameter  | <pre>-W -j &lt;name&gt;=&lt;value&gt; -i &lt;stor_ip_addr stor_hostname&gt; [:&lt;port_num&gt;] [-n &lt;iscsi_name&gt;]] &lt;adapter_name&gt;  --parameter --set &lt;name&gt;=&lt;value&gt; --ip &lt;stor_ip_addr stor_hostname&gt; [:&lt;port_num&gt;]   [--name &lt;iscsi_name&gt;]] &lt;adapter_name&gt;</pre>              | Sets iSCSI parameters.                                                                                                   |
| -W --parameter  | <pre>-W -o &lt;param_name&gt; -i &lt;stor_ip_addr stor_hostname&gt; [:&lt;port_num&gt;] [-n &lt;iscsi_name&gt;]] &lt;adapter_name&gt;  -parameter --reset &lt;param_name&gt; -ip &lt;stor_ip_addr stor_hostname&gt; [:&lt;port_num&gt;]   [-name &lt;iscsi_name&gt;]] &lt;adapter_name&gt;</pre>                               | Returns parameters in discovery target or send target to default inheritance behavior.                                   |
| -z --reset_auth | <pre>-a -z -m &lt;auth_method&gt; -b [-i &lt;stor_ip_addr stor_hostname&gt; [:&lt;portnum&gt;] [-n &lt;iscsi_name&gt;]] &lt;adapter_name&gt;  --authentication --reset_auth --method &lt;auth_method&gt; [--ip &lt;stor_ip_addr stor_hostname&gt; [:&lt;portnum&gt;]   [--name &lt;iscsi_name&gt;]] &lt;adapter_name&gt;</pre> | Resets target level authentication properties to be inherited from adapter level. Used with the --authentication option. |

## iSCSI Storage Setup with ESXCLI

You can set up iSCSI storage using vSphere Web Client, commands in the `esxcli iscsi` namespace, or `vicfg-iscsi` commands (see [“iSCSI Storage Setup with vicfg-iscsi”](#) on page 73).

### Setting Up Software iSCSI with ESXCLI

Software iSCSI setup requires several tasks. For each task, see the discussion of the corresponding command in this chapter or the reference information available from `esxcli iscsi --help` and the VMware Documentation Center. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`

- 1 Enable software iSCSI.

```
esxcli <conn_options> iscsi software set --enabled=true
```

- 2 Check whether a network portal, that is, a bound port, exists for iSCSI traffic.

```
esxcli <conn_options> iscsi adapter list
```

- 3 If no adapter exists, add one. Software iSCSI does not require port binding, but requires that at least one VMkernel NIC is available and can be used as an iSCSI NIC. You can name the adapter as you add it.

```
esxcli <conn_options> iscsi networkportal add -n <portal_name> -A <vmhba>
```

- 4 (Optional) Check the status.

```
esxcli <conn_options> iscsi software get
```

The system prints `true` if software iSCSI is enabled, or `false` if it is not enabled.

- 5 (Optional) Set the iSCSI name and alias.

```
esxcli <conn_options> iscsi adapter set --adapter=<iscsi_adapter> --name=<name>
esxcli <conn_options> iscsi adapter set --adapter=<iscsi_adapter> --alias=<alias>
```

- 6 Add a dynamic discovery address or a static discovery address.

The two types of target differ as follows:

- With dynamic discovery, all storage targets associated with a host name or IP address are discovered. You run the following command.

```
esxcli <conn_options> iscsi adapter discovery sendtarget add --address=<ip/dns[:port]>
--adapter=<adapter_name>
```

- With static discovery, you must specify the host name or IP address and the iSCSI name of the storage target. You run the following command.

```
esxcli <conn_options> iscsi adapter discovery statictarget add --address=<ip/dns[:port]>
--adapter=<adapter_name> --name=<target_name>
```

When you later remove a discovery address, it might still be displayed as the parent of a static target. You can add the discovery address and rescan to display the correct parent for the static targets.

- 7 (Optional) Set the authentication information for CHAP (see [“Setting iSCSI CHAP”](#) on page 62 and [“Enabling iSCSI Authentication”](#) on page 82). You can set per-target CHAP for static targets, per-adapter CHAP, or apply the command to the discovery address.



|                         |                                                                                                                                                                                                                                                                                                           |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Adapter-level<br>CHAP   | <code>esxcli iscsi adapter auth chap set --direction=uni<br/>--chap_username=&lt;name&gt; --chap_password=&lt;pwd&gt;<br/>--level=[prohibited, discouraged, preferred, required]<br/>--secret=&lt;string&gt; --adapter=&lt;vmhba&gt;</code>                                                               |
| Discovery-level<br>CHAP | <code>esxcli iscsi adapter discovery sendtarget auth chap set --direction=uni<br/>--chap_username=&lt;name&gt; --chap_password=&lt;pwd&gt;<br/>--level=[prohibited, discouraged, preferred, required]<br/>--secret=&lt;string&gt; --adapter=&lt;vmhba&gt;<br/>--address=&lt;sendtarget_address&gt;</code> |
| Target-level<br>CHAP    | <code>esxcli iscsi adapter target portal auth chap set --direction=uni<br/>--chap_username=&lt;name&gt; --chap_password=&lt;pwd&gt;<br/>--level=[prohibited, discouraged, preferred, required]<br/>--secret=&lt;string&gt; --adapter=&lt;vmhba&gt;<br/>--name&lt;iscsi_iqn_name&gt;</code>                |

[Table 5-1, “Supported Levels for CHAP,”](#) on page 62 lists what each supported level does.

For example:

```
esxcli <conn_options> iscsi adapter auth chap set --direction=uni --chap_username=<name>
--chap_password=<pwd> --level=preferred --secret=uni_secret --adapter=vmhba33
```

- 8 (Optional) Set the authentication information for mutual CHAP by running `esxcli iscsi adapter auth chap set` again with `--direction` set to `mutual` and a different authentication user name and secret.

|                         |                                                                                                                                                                                                                                                                                           |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Adapter-level<br>CHAP   | <code>esxcli iscsi adapter auth chap set --direction=mutual<br/>--mchap_username=&lt;name2&gt; --mchap_password=&lt;pwd2&gt;<br/>--level=[prohibited required] --secret=&lt;string2&gt;<br/>--adapter=&lt;vmhba&gt;</code>                                                                |
| Discovery-level<br>CHAP | <code>esxcli iscsi adapter discovery sendtarget auth chap set<br/>--direction=mutual --mchap_username=&lt;name2&gt;<br/>--mchap_password=&lt;pwd2&gt; --level=[prohibited, required]<br/>--secret=&lt;string2&gt; --adapter=&lt;vmhba&gt;<br/>--address=&lt;sendtarget_address&gt;</code> |
| Target-level<br>CHAP    | <code>eesxcli iscsi adapter target portal auth chap set --direction=mutual<br/>--mchap_username=&lt;name2&gt; --mchap_password=&lt;pwd2&gt;<br/>--level=[prohibited required] --secret=&lt;string2&gt;<br/>--adapter=&lt;vmhba&gt; --name=&lt;iscsi_iqn_name&gt;</code>                   |

---

**IMPORTANT** You are responsible for making sure that CHAP is set before you set mutual CHAP, and for using compatible levels for CHAP and mutual CHAP.

---

- 9 (Optional) Set iSCSI parameters.

|                               |                                                                                                                                                                                     |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Adapter-level<br>parameters   | <code>esxcli iscsi adapter param set --adapter=&lt;vmhba&gt; --key=&lt;key&gt;<br/>--value=&lt;value&gt;</code>                                                                     |
| Discovery-level<br>parameters | <code>esxcli iscsi adapter discovery sendtarget param set --adapter=&lt;vmhba&gt;<br/>--key=&lt;key&gt; --value=&lt;value&gt;<br/>--address=&lt;sendtarget_address&gt;</code>       |
| Target-level<br>parameters    | <code>esxcli iscsi adapter target portal param set --adapter=&lt;vmhba&gt;<br/>--key=&lt;key&gt; --value=&lt;value&gt; --address=&lt;address&gt;<br/>--name=&lt;iqn.name&gt;</code> |

See [“Listing and Setting iSCSI Parameters”](#) on page 78

- 10 After setup is complete, perform rediscovery and rescan all storage devices. For example:

```
esxcli <conn_options> iscsi adapter discovery rediscover
esxcli <conn_options> storage core adapter rescan --adapter=vmhba36
```

- 11 (Optional) If you want to make additional iSCSI login parameter changes (see [“Listing and Setting iSCSI Parameters”](#) on page 78), you must log out of the corresponding iSCSI session and log back in.
  - a Run `esxcli iscsi session remove` to log out.
  - b Run `esxcli iscsi session add` or rescan the adapter to add the session back.

## Setting Up Dependent Hardware iSCSI with ESXCLI

Dependent hardware iSCSI setup requires several high-level tasks. For each task, see the discussion of the corresponding command in this chapter or the reference information available from `esxcli iscsi --help` and the VMware Documentation Center. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- 1 Determine the iSCSI adapter type and retrieve the iSCSI adapter ID.
 

```
esxcli <conn_options> iscsi adapter list
```
- 2 (Optional) Set the iSCSI name and alias.
 

```
esxcli <conn_options> iscsi adapter set --adapter <adapter_name> --name=<name>
esxcli <conn_options> iscsi adapter set --adapter <adapter_name> --alias=<alias>
```
- 3 Set up port binding by following these steps:
  - a Identify the VMkernel port of the dependent hardware iSCSI adapter.
 

```
esxcli <conn_options> iscsi logicalnetworkportal list --adapter=<adapter_name>
```
  - b Connect the dependent hardware iSCSI initiator to the iSCSI VMkernel ports by running the following command for each port.
 

```
esxcli <conn_options> iscsi networkportal add --nic=<bound_vmknics>
--adapter=<iscsi_adapter>
```
  - c Verify that the ports were added to the dependent hardware iSCSI initiator.
 

```
esxcli <conn_options> iscsi physicalnetworkportal list --adapter=<adapter_name>
```
- 4 Add a dynamic discovery address or a static discovery address.

The two types of target differ as follows:

- With dynamic discovery, all storage targets associated with a host name or IP address are discovered. You run the following command.

```
esxcli <conn_options> iscsi adapter discovery sendtarget add --address=<ip/dns[:port]>
--adapter=<adapter_name>
```

- With static discovery, you must specify the host name or IP address and the iSCSI name of the storage target. You run the following command.

```
esxcli <conn_options> iscsi adapter discovery statictarget add --address=<ip/dns[:port]>
--adapter=<adapter_name> --name=<target_name>
```

When you later remove a discovery address, it might still be displayed as the parent of a static target. You can add the discovery address and rescan to display the correct parent for the static targets.

- 5 (Optional) Set the authentication information for CHAP (see [“Setting iSCSI CHAP”](#) on page 62 and [“Enabling iSCSI Authentication”](#) on page 82). You can set per-target CHAP for static targets, per-adapter CHAP, or apply the command to the discovery address.

|                         |                                                                                                                                                                                                                                                                                                           |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Adapter-level<br>CHAP   | <code>esxcli iscsi adapter auth chap set --direction=uni<br/>--chap_username=&lt;name&gt; --chap_password=&lt;pwd&gt;<br/>--level=[prohibited, discouraged, preferred, required]<br/>--secret=&lt;string&gt; --adapter=&lt;vmhba&gt;</code>                                                               |
| Discovery-level<br>CHAP | <code>esxcli iscsi adapter discovery sendtarget auth chap set --direction=uni<br/>--chap_username=&lt;name&gt; --chap_password=&lt;pwd&gt;<br/>--level=[prohibited, discouraged, preferred, required]<br/>--secret=&lt;string&gt; --adapter=&lt;vmhba&gt;<br/>--address=&lt;sendtarget_address&gt;</code> |
| Target-level<br>CHAP    | <code>esxcli iscsi adapter target portal auth chap set --direction=uni<br/>--chap_username=&lt;name&gt; --chap_password=&lt;pwd&gt;<br/>--level=[prohibited, discouraged, preferred, required]<br/>--secret=&lt;string&gt; --adapter=&lt;vmhba&gt;<br/>--name=&lt;iscsi_iqn_name&gt;</code>               |

[Table 5-1, “Supported Levels for CHAP,”](#) on page 62 lists what each supported level does.

For example:

```
esxcli <conn_options> iscsi adapter auth chap set --direction=uni --chap_username=<name>
--chap_password=<pwd> --level=preferred --secret=uni_secret --adapter=vmhba33
```

- 6 (Optional) Set the authentication information for mutual CHAP by running `esxcli iscsi adapter auth chap set` again with `--direction` set to `mutual` and a different authentication user name and secret.

|                         |                                                                                                                                                                                                                                                                                         |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Adapter-level<br>CHAP   | <code>esxcli iscsi adapter auth chap set --direction=mutual<br/>--mchap_username=&lt;name&gt; --mchap_password=&lt;pwd&gt;<br/>--level=[prohibited required] --secret=&lt;string2&gt;<br/>--adapter=&lt;vmhba&gt;</code>                                                                |
| Discovery-level<br>CHAP | <code>esxcli iscsi adapter discovery sendtarget auth chap set<br/>--direction=mutual --mchap_username=&lt;name&gt;<br/>--mchap_password=&lt;pwd&gt; --level=[prohibited, required]<br/>--secret=&lt;string2&gt; --adapter=&lt;vmhba&gt;<br/>--address=&lt;sendtarget_address&gt;</code> |
| Target-level<br>CHAP    | <code>esxcli iscsi adapter target portal auth chap set --direction=mutual<br/>--mchap_username=&lt;name&gt; --mchap_password=&lt;pwd&gt;<br/>--level=[prohibited required] --secret=&lt;string2&gt;<br/>--adapter=&lt;vmhba&gt; --name=&lt;iscsi_iqn_name&gt;</code>                    |

---

**IMPORTANT** You are responsible for making sure that CHAP is set before you set mutual CHAP, and for using compatible levels for CHAP and mutual CHAP.

---

- 7 (Optional) Set iSCSI parameters.

|                               |                                                                                                                                                                                     |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Adapter-level<br>parameters   | <code>esxcli iscsi adapter param set --adapter=&lt;vmhba&gt; --key=&lt;key&gt;<br/>--value=&lt;value&gt;</code>                                                                     |
| Discovery-level<br>parameters | <code>esxcli iscsi adapter discovery sendtarget param set --adapter=&lt;vmhba&gt;<br/>--key=&lt;key&gt; --value=&lt;value&gt;<br/>--address=&lt;sendtarget_address&gt;</code>       |
| Target-level<br>parameters    | <code>esxcli iscsi adapter target portal param set --adapter=&lt;vmhba&gt;<br/>--key=&lt;key&gt; --value=&lt;value&gt; --address=&lt;address&gt;<br/>--name=&lt;iqn.name&gt;</code> |

See [“Listing and Setting iSCSI Parameters”](#) on page 78

- 8 After setup is complete, perform rediscovery and rescan all storage devices. For example:

```
esxcli <conn_options> iscsi adapter discovery rediscover
esxcli <conn_options> storage core adapter rescan --adapter=vmhba36
```

- 9 (Optional) If you want to make additional iSCSI login parameter changes (see [“Listing and Setting iSCSI Parameters”](#) on page 78), you must log out of the corresponding iSCSI session and log back in.
  - a Run `esxcli iscsi session remove` to log out.
  - b Run `esxcli iscsi session add` or rescan the adapter to add the session back.

## Setting Up Independent Hardware iSCSI with ESXCLI

With independent hardware-based iSCSI storage, you use a specialized third-party adapter capable of accessing iSCSI storage over TCP/IP. This iSCSI initiator handles all iSCSI and network processing and management for your ESXi system.

You must install and configure the independent hardware iSCSI adapter for your host before you can access the iSCSI storage device. For installation information, see vendor documentation.

Hardware iSCSI setup requires a number of high-level tasks. For each task, see the discussion of the corresponding command-line option in this chapter or the reference information. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- 1 Determine the iSCSI adapter type and retrieve the iSCSI adapter ID.
 

```
esxcli <conn_options> iscsi adapter list
```
- 2 Configure the hardware initiator (HBA) by running `esxcli iscsi networkportal ipconfig` with one or more of the following options.

|                                       |                                              |
|---------------------------------------|----------------------------------------------|
| <code>-A --adapter=&lt;str&gt;</code> | iSCSI adapter name. (required).              |
| <code>-1 --dns1=&lt;str&gt;</code>    | iSCSI network portal primary DNS address.    |
| <code>-2 --dns2=&lt;str&gt;</code>    | iSCSI network portal secondary DNS address.  |
| <code>-g --gateway=&lt;str&gt;</code> | iSCSI network portal gateway address.        |
| <code>-i --ip=&lt;str&gt;</code>      | iSCSI network portal IP address (required).  |
| <code>-n --nic=&lt;str&gt;</code>     | iSCSI network portal (vmknic).               |
| <code>-s --subnet=&lt;str&gt;</code>  | iSCSI network portal subnet mask (required). |

- 3 (Optional) Set the iSCSI name and alias.

```
esxcli <conn_options> iscsi adapter set --adapter <adapter_name> --name=<name>
esxcli <conn_options> iscsi adapter set --adapter <adapter_name> --alias=<alias>
```

- 4 Add a dynamic discovery address or a static discovery address.

The two types of target differ as follows:

- With dynamic discovery, all storage targets associated with a host name or IP address are discovered. You run the following command.

```
esxcli <conn_options> iscsi adapter discovery sendtarget add --address=<ip/dns[:port]>
--adapter=<adapter_name>
```

- With static discovery, you must specify the host name or IP address and the iSCSI name of the storage target. You run the following command.

```
esxcli <conn_options> iscsi adapter discovery statictarget add --address=<ip/dns[:port]>
```

- 5 (Optional) Set the authentication information for CHAP (see [“Setting iSCSI CHAP”](#) on page 62 and [“Enabling iSCSI Authentication”](#) on page 82). You can set per-target CHAP for static targets, per-adapter CHAP, or apply the command to the discovery address.

|                         |                                                                                                                                                                                                                                                                                        |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Adapter-level<br>CHAP   | <pre>esxcli iscsi adapter auth chap set --direction=uni --chap_username=&lt;name&gt; --chap_password=&lt;pwd&gt; --level=[prohibited, discouraged, preferred, required] --secret=&lt;string&gt; --adapter=&lt;vmhba&gt;</pre>                                                          |
| Discovery-level<br>CHAP | <pre>esxcli iscsi adapter discovery sendtarget auth chap set --direction=uni --chap_username=&lt;name&gt; --chap_password=&lt;pwd&gt; --level=[prohibited, discouraged, preferred, required] --secret=&lt;string&gt; --adapter=&lt;vmhba&gt; --address&lt;sendtarget_address&gt;</pre> |
| Target-level<br>CHAP    | <pre>esxcli iscsi adapter target portal auth chap set --direction=uni --chap_username=&lt;name&gt; --chap_password=&lt;pwd&gt; --level=[prohibited, discouraged, preferred, required] --secret=&lt;string&gt; --adapter=&lt;vmhba&gt; --name&lt;iscsi_iqn_name&gt;</pre>               |

Table 5-1, “Supported Levels for CHAP,” on page 62 lists what each supported level does.

For example:

```
esxcli <conn_options> iscsi adapter auth chap set --direction=uni --chap_username=<name>
--chap_password=<pwd> --level=preferred --secret=uni_secret --adapter=vmhba33
```

Mutual CHAP is not supported for independent hardware iSCSI storage.

- (Optional) Set iSCSI parameters.

|                               |                                                                                                                                                                           |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Adapter-level<br>parameters   | <pre>esxcli iscsi adapter param set --adapter=&lt;vmhba&gt; --key=&lt;key&gt; --value=&lt;value&gt;</pre>                                                                 |
| Discovery-level<br>parameters | <pre>esxcli iscsi adapter discovery sendtarget param set --adapter=&lt;vmhba&gt; --key=&lt;key&gt; --value=&lt;value&gt; --address=&lt;sendtarget_address&gt;</pre>       |
| Target-level<br>parameters    | <pre>esxcli iscsi adapter target portal param set --adapter=&lt;vmhba&gt; --key=&lt;key&gt; --value=&lt;value&gt; --address=&lt;address&gt; --name=&lt;iqn.name&gt;</pre> |

See “Listing and Setting iSCSI Parameters” on page 78

- After setup is complete, run `esxcli storage core adapter rescan --adapter=<iscsi_adapter>` to rescan all storage devices.
- After setup is complete, perform rediscovery and rescan all storage devices. For example:

```
esxcli <conn_options> iscsi adapter discovery rediscover
esxcli <conn_options> storage core adapter rescan --adapter=vmhba36
```

## iSCSI Storage Setup with vicfg-iscsi

You can set up iSCSI storage using the vSphere Web Client, commands in the `esxcli iscsi` namespace (see “iSCSI Storage Setup with ESXCLI” on page 68) or the `vicfg-iscsi` command.

### Setting Up Software iSCSI with vicfg-iscsi

Software iSCSI setup requires a number of high-level tasks. For each task, see the discussion of the corresponding command-line option in this chapter or the reference information. Specify one of the options listed in “Connection Options for vCLI Host Management Commands” on page 18 in place of `<conn_options>`.

- Determine the HBA type and retrieve the HBA ID.  
`vicfg-iscsi <conn_options> --adapter --list`
- Enable software iSCSI for the HBA.

```
vicfg-iscsi <conn_options> --swiscsi --enable
```

- 3 (Optional) Check the status.

```
vicfg-iscsi <conn_options> --swiscsi --list
```

The system prints `Software iSCSI is enabled` or `Software iSCSI is not enabled`.

- 4 (Optional) Set the iSCSI name and alias.

```
vicfg-iscsi <conn_options> -I -n <iscsi_name> <adapter_name>
vicfg-iscsi <conn_options> --iscsiname --name <iscsi_name> <adapter_name>
vicfg-iscsi <conn_options> -I -a <alias_name> <adapter_name>
vicfg-iscsi <conn_options> --iscsiname --alias <alias_name> <adapter_name>
```

- 5 Add a dynamic discovery address or a static discovery address.

The two types of target differ as follows:

- With dynamic discovery, all storage targets associated with a host name or IP address are discovered. You run the following command:

```
vicfg-iscsi <conn_options> --discovery --add --ip <ip_addr | domain_name> <adapter_name>
```

- With static discovery, you must specify the host name or IP address and the iSCSI name of the storage target. You run the following command:

```
vicfg-iscsi <conn_options> --static --add --ip <ip_addr | domain_name>
--name <iscsi_name> <adapter_name>
```

When you later remove a discovery address, it might still be displayed as the parent of a static target. You can add the discovery address and rescan to display the correct parent for the static targets.

- 6 (Optional) Set the authentication information for CHAP (see [“Setting iSCSI CHAP”](#) on page 62 and [“Enabling iSCSI Authentication”](#) on page 82).

```
vicfg-iscsi <conn_options> -A -c <level> -m <auth_method> -u <auth_u_name> -w <chap_password>
[-i <stor_ip_addr|stor_hostname> [:<portnum>] [-n <iscsi_name>]] <adapter_name>
vicfg-iscsi <conn_options> --authentication --level <level> --method <auth_method>
--chap_username <auth_u_name> --chap_password <chap_password>
[--ip <stor_ip_addr|stor_hostname> [:<portnum>] [-name <iscsi_name>]]
<adapter_name>
```

The target (-i) and name (-n) option determine what the command applies to.

| Option            | Result                                                 |
|-------------------|--------------------------------------------------------|
| -i and -n         | Command applies to per-target CHAP for static targets. |
| Only -i           | Command applies to the discovery address.              |
| Neither -i nor -n | Command applies to per-adapter CHAP.                   |

- 7 (Optional) Set the authentication information for mutual CHAP by running `vicfg-iscsi -A` again with the `-b` option and a different authentication user name and password.

For <level>, specify `chapProhibited` or `chapRequired`.

- `chapProhibited` – The host does not use CHAP authentication. If authentication is enabled, specify `chapProhibited` to disable it.
- `chapRequired` – The host requires successful CHAP authentication. The connection fails if CHAP negotiation fails. You can set this value for mutual CHAP only if CHAP is set to `chapRequired`.

For <auth\_method>, CHAP is the only valid value.

---

**IMPORTANT** You are responsible for making sure that CHAP is set before you set mutual CHAP, and for using compatible levels for CHAP and mutual CHAP.

---

- 8 (Optional) Set iSCSI parameters by running `vicfg-iscsi -W`.

- 9 After setup is complete, run `vicfg-rescan` to rescan all storage devices.

## Setting Up Dependent Hardware iSCSI with `vicfg-iscsi`

Dependent hardware iSCSI setup requires a number of high-level tasks. For each task, see the discussion of the corresponding command-line option in this chapter, or the reference information. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- 1 Determine the HBA type and retrieve the HBA ID.

```
vicfg-iscsi <conn_options> --adapter --list
```

- 2 (Optional) Set the iSCSI name and alias.

```
vicfg-iscsi <conn_options> -I -n <iscsi_name> <adapter_name>
vicfg-iscsi <conn_options> --iscsiname --name <iscsi_name> <adapter_name>
vicfg-iscsi <conn_options> -I -a <alias_name> <adapter_name>
vicfg-iscsi <conn_options> --iscsiname --alias <alias_name> <adapter_name>
```

- 3 Set up port binding by following these steps:

- a Identify the VMkernel port of the dependent hardware iSCSI adapter.

```
esxcli <conn_options> swiscsi vmknics list -d <vmhba>
```

- b Connect the dependent hardware iSCSI initiator to the iSCSI VMkernel ports by running the following command for each port.

```
esxcli <conn_options> swiscsi nic add -n <port_name> -d <vmhba>
```

- c Verify that the ports were added to the dependent hardware iSCSI initiator.

```
esxcli <conn_options> swiscsi nic list -d <vmhba>
```

- d Rescan the dependent hardware SCSI initiator.

```
vicfg-rescan <conn_options> <vmhba>
```

- 4 Add a dynamic discovery address or a static discovery address.

The two types of target differ as follows:

- With dynamic discovery, all storage targets associated with a host name or IP address are discovered. You run the following command:

```
vicfg-iscsi <conn_options> --discovery --add --ip <ip_addr | domain_name> <adapter_name>
```

- With static discovery, you must specify the host name or IP address and the iSCSI name of the storage target. You run the following command:

```
vicfg-iscsi <conn_options> --static --add --ip <ip_addr | domain_name>
--name <iscsi_name> <adapter_name>
```

When you later remove a discovery address, it might still be displayed as the parent of a static target. You can add the discovery address and rescan to display the correct parent for the static targets.

- 5 (Optional) Set the authentication information for CHAP (see [“Setting iSCSI CHAP”](#) on page 62 and [“Enabling iSCSI Authentication”](#) on page 82).

```
vicfg-iscsi <conn_options> -A -c <level> -m <auth_method> -u <auth_u_name> -w <chap_password>
[-i <stor_ip_addr|stor_hostname> [[:portnum]] [-n <iscsi_name>]] <adapter_name>
vicfg-iscsi <conn_options> --authentication --level <level> --method <auth_method>
--chap_username <auth_u_name> --chap_password <chap_password>
[--ip <stor_ip_addr|stor_hostname> [[:portnum]] [-name <iscsi_name>]]
<adapter_name>
```

The target (`-i`) and name (`-n`) option determine what the command applies to.

| Option            | Result                                                 |
|-------------------|--------------------------------------------------------|
| -i and -n         | Command applies to per-target CHAP for static targets. |
| Only -i           | Command applies to the discovery address.              |
| Neither -i nor -n | Command applies to per-adapter CHAP.                   |

- 6 (Optional) Set the authentication information for mutual CHAP by running `vicfg-iscsi -A` again with the `-b` option and a different authentication user name and password.

For <level>, specify `chapProhibited` or `chapRequired`.

- `chapProhibited` – The host does not use CHAP authentication. If authentication is enabled, specify `chapProhibited` to disable it.
- `chapRequired` – The host requires successful CHAP authentication. The connection fails if CHAP negotiation fails. You can set this value for mutual CHAP only if CHAP is set to `chapRequired`.

For <auth\_method>, CHAP is the only valid value.

---

**IMPORTANT** You are responsible for making sure that CHAP is set before you set mutual CHAP, and for using compatible levels for CHAP and mutual CHAP.

---

- 7 (Optional) Set iSCSI parameters by running `vicfg-iscsi -W`.
- 8 After setup is complete, run `vicfg-rescan` to rescan all storage devices.

## Setting Up Independent Hardware iSCSI with `vicfg-iscsi`

With independent hardware-based iSCSI storage, you use a specialized third-party adapter capable of accessing iSCSI storage over TCP/IP. This iSCSI initiator handles all iSCSI and network processing and management for your ESXi system.

You must install and configure the independent hardware iSCSI adapter for your host before you can access the iSCSI storage device. For installation information, see vendor documentation.

Hardware iSCSI setup requires a number of high-level tasks. For each task, see the discussion of the corresponding command-line option in this chapter, the manpage (Linux), or the reference information. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of <conn\_options>.

- 1 Determine the HBA type and retrieve the HBA ID.
- ```
vicfg-iscsi <conn_options> --adapter --list
```
- 2 Configure the hardware initiator (HBA) by running `vicfg-iscsi -N` with one or more of the following options.

- `--list` – List network properties.
- `--ip <ip_addr>` – Set HBA IPv4 address.
- `--subnetmask <subnet_mask>` – Set HBA network mask.
- `--gateway <default_gateway>` – Set HBA gateway.
- `--set ARP=true|false` – Enable or disable ARP redirect.

You can also set the HBA IPv4 address and network mask and gateway in one command.

```
vicfg-iscsi <conn_options> --ip <ip_addr> --subnetmask <subnet_mask> --gateway <default_gateway>
```

- 3 (Optional) Set the iSCSI name and alias.

```
vicfg-iscsi <conn_options> -I -n <iscsi_name> <adapter_name>
vicfg-iscsi <conn_options> --iscsiname --name <iscsi_name> <adapter_name>
vicfg-iscsi <conn_options> -I -a <alias_name> <adapter_name>
```



```
vicfg-iscsi <conn_options> --iscsiname --alias <alias_name> <adapter_name>
```

- 4 Add a dynamic discovery address or a static discovery address.

The two types of target differ as follows:

- With dynamic discovery, all storage targets associated with an IP address are discovered. Run the following command:

```
vicfg-iscsi <conn_options> --discovery --add --ip <ip_addr> <adapter_name>
```

- With static discovery, you must specify the IP address and the iSCSI name of the storage target to be added. Run the following command:

```
vicfg-iscsi <conn_options> --static --add --ip <ip_addr>
--name <iscsi_name> <adapter_name>
```

When you later remove a discovery address, it might still be displayed as the parent of a static target. You can later add the discovery address and rescan to display the correct parent for the static targets.

- 5 (Optional) Set the authentication information for CHAP by running `vicfg-iscsi -A`.

You can set the information for per adapter, per discovery, and per target CHAP. See [“Setting iSCSI CHAP”](#) on page 62 and [“Enabling iSCSI Authentication”](#) on page 82.

```
vicfg-iscsi <conn_options> --authentication --level <level> --method <auth_method>
--chap_username <auth_u_name> --chap_password <chap_password>
[--ip <stor_ip_addr|stor_hostname> [:<portnum>] [--name <iscsi_name>]
<adapter_name>
```

The target (-i) and name (-n) option determine what the command applies to.

Option	Result
-i and -n	Command applies to per-target CHAP for static targets.
Only -i	Command applies to the discovery address.
Neither -i nor -n	Command applies to per-adapter CHAP.

Mutual CHAP is not supported for independent hardware iSCSI storage.

- 6 (Optional) Set additional iSCSI parameters by running `vicfg-iscsi -W`.
- 7 After setup is complete, call `vicfg-rescan` to rescan all storage devices.

## Listing and Setting iSCSI Options

You can list and set iSCSI options with ESXCLI or with `vicfg-iscsi`. You can also manage parameters. See [“Listing and Setting iSCSI Parameters”](#) on page 78.

### Listing iSCSI Options with ESXCLI

Use `esxcli iscsi` information retrieval commands to list external HBA properties, information about targets, and LUNs. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- Run `esxcli iscsi adapter firmware` to list or upload the firmware for the iSCSI adapter.

```
esxcli <conn_options> iscsi adapter firmware get --adapter=<adapter_name>
esxcli <conn_options> iscsi adapter firmware set --file=<firmware_file_path>
```

The system returns information about the vendor, model, description, and serial number of the HBA.

- Run commands in the `esxcli iscsi adapter target name` space.
  - `esxcli iscsi adapter target portal` lists and sets authentication and portal parameters.
  - `esxcli iscsi adapter target list` lists LUN information.

## Setting MTU with ESXCLI

If you want to change the MTU used for your iSCSI storage, you must make the change in two places.

- Run `esxcli network vswitch standard set` to change the MTU of the virtual switch.
- Run `esxcli network ip interface set` to change the MTU of the network interface.

## Listing and Setting iSCSI Options with vicfg-iscsi

Use `vicfg-iscsi` information retrieval options to list external HBA properties, information about targets, and LUNs. You can use the following `vicfg-iscsi` options to list iSCSI parameters. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- Run `vicfg-iscsi -P|--phba` to list external (vendor-specific) properties of an iSCSI adapter.

```
vicfg-iscsi <conn_options> -P -l <adapter_name>
vicfg-iscsi <conn_options> --phba --list <adapter_name>
```

The system returns information about the vendor, model, description, and serial number of the HBA.

- Run `vicfg-iscsi -T | --target` to list target information.

```
vicfg-iscsi <conn_options> -T -l <adapter_name>
vicfg-iscsi <conn_options> --target --list <adapter_name>
```

The system returns information about targets for the specified adapter, including the iSCSI name (IQN or EUI format) and alias. See [“Discovery Target Names”](#) on page 61.

- Run `vicfg-iscsi -L|--lun` to list LUN information.

```
vicfg-iscsi <conn_options> -L -l <adapter_name>
vicfg-iscsi <conn_options> --lun --list <adapter_name>
```

The command returns the operating system device name, bus number, target ID, LUN ID, and LUN size for the LUN.

- Run `vicfg-iscsi -L` with `-t` to list only LUNs on a specified target.

```
vicfg-iscsi <conn_options> -L -l -t <target_ID> <adapter_name>
vicfg-iscsi <conn_options> --lun --list --target_id <target_id> <adapter_name>
```

The system returns the LUNs on the specified target and the corresponding device name, device number, LUN ID, and LUN size.

- Run `vicfg-iscsi -p|--pnp` to list physical network portal information for independent hardware iSCSI devices. You also use this option with `--mtu`.

```
vicfg-iscsi <conn_options> -p -l <adapter_name>
vicfg-iscsi <conn_options> --pnp --list <adapter_name>
```

The system returns information about the MAC address, MTU, and current transfer rate.

- Run `vicfg-iscsi -I -l` to list information about the iSCSI initiator. ESXi systems use a software-based iSCSI initiator in the VMkernel to connect to storage. The command returns the iSCSI name, alias name, and alias settable bit for the initiator.

```
vicfg-iscsi <conn_options> -I -l vmhba42
```

- Run `vicfg-iscsi -p -M` to set the MTU for the adapter. You specify the size and adapter name.

```
vicfg-iscsi <conn_options> -p -M <mtu_size> <adapter_name>
vicfg-iscsi <conn_options> --pnp --mtu <mtu-size> <adapter_name>
```

## Listing and Setting iSCSI Parameters

You can list and set iSCSI parameters for software iSCSI and for dependent hardware iSCSI with ESXCLI or with `vicfg-iscsi`.

## Listing and Setting iSCSI Parameters with ESXCLI

You can retrieve and set iSCSI parameters by running one of the following commands.

```

Adapter-level parameters  esxcli iscsi adapter param set --adapter=<vmhba> --key=<key> --value=<value>
Target-level parameters   esxcli iscsi adapter target portal param set --adapter=<vmhba> --key=<key>
                             --value=<value> --address=<address> --name=<iqn.name>
Discovery-level parameters esxcli iscsi adapter discovery sendtarget param set --adapter=<vmhba>
                             --key=<key> --value=<value> --address=<address>

```

[Table 5-6](#) lists all settable parameters. These parameters are also described in the IETF rfc 3720. You can run `esxcli iscsi adapter param get` to determine whether a parameter is settable or not.

The parameters in [Table 5-6](#) apply to software iSCSI and dependent hardware iSCSI.

**Table 5-5.** Settable iSCSI Parameters

Parameter	Description
DataDigestType	Increases data integrity. When data digest is enabled, the system performs a checksum over each PDUs data part and verifies using the CRC32C algorithm. <b>Note:</b> Systems that use Intel Nehalem processors offload the iSCSI digest calculations for software iSCSI, thus reducing the impact on performance. Valid values are <code>digestProhibited</code> , <code>digestDiscouraged</code> , <code>digestPreferred</code> , or <code>digestRequired</code> .
HeaderDigest	Increases data integrity. When header digest is enabled, the system performs a checksum over the header part of each iSCSI Protocol Data Unit (PDU) and verifies using the CRC32C algorithm.
MaxOutstandingR2T	Max Outstanding R2T defines the Ready to Transfer (R2T) PDUs that can be in transition before an acknowledgement PDU is received.
FirstBurstLength	Maximum amount of unsolicited data an iSCSI initiator can send to the target during the execution of a single SCSI command, in bytes.
MaxBurstLength	Maximum SCSI data payload in a Data-In or a solicited Data-Out iSCSI sequence, in bytes.
MaxRecvDataSegLen	Maximum data segment length, in bytes, that can be received in an iSCSI PDU.
NoopOutInterval	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. Supported only at the initiator level.
NoopOutTimeout	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 <code>NoopTimeout</code> limit is exceeded, the initiator terminates the current session and starts a new one. Supported only at the initiator level.
RecoveryTimeout	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.
DelayedAck	Allows systems to delay acknowledgment of received data packets.

You can use the following ESXCLI commands to list parameter options.

- Run `esxcli iscsi adapter param get` to list parameter options for the iSCSI adapter.
- Run `esxcli iscsi adapter discovery sendtarget param get` or `esxcli iscsi adapter target portal param set` to retrieve information about iSCSI parameters and whether they are settable.
- Run `esxcli iscsi adapter discovery sendtarget param get` or `esxcli iscsi adapter target portal param set` to set iSCSI parameter options.

If special characters are in the `<name>=<value>` sequence, for example, if you add a space, you must surround the sequence with double quotes ("`<name> = <value>`").

## Returning Parameters to Default Inheritance

The values of iSCSI parameters associated with a dynamic discovery address or a static discovery target are inherited from the corresponding settings of the parent. For the dynamic discovery address, the parent is the adapter. For the static target, the parent is the adapter or discovery address.

- If you use the vSphere Web Client to modify authentication settings, you deselect the **Inherit from Parent** check box before you can make a change to the discovery address or discovery target.
- If you use `esxcli iscsi`, the value you set overrides the inherited value.

Inheritance is relevant only if you want to return a dynamic discovery address or a static discovery target to its inherited value. In that case, use the following command, which requires the `--name` option for static discovery addresses, but not for dynamic discovery targets.

- Dynamic target: `esxcli iscsi adapter discovery sendtarget param set`
- Static target: `esxcli iscsi adapter target portal param set`

## Listing and Setting iSCSI Parameters with `vicfg-iscsi`

You can list and set iSCSI parameters by running `vicfg-iscsi -W`. [Table 5-6](#) lists all settable parameters. These parameters are also described in the IETF rfc 3720. You can also run `vicfg-iscsi --parameter --list --details` to determine whether a parameter is settable or not.

The parameters in [Table 5-6](#) apply to software iSCSI and dependent hardware iSCSI.

**Table 5-6.** Settable iSCSI Parameters

Parameter	Description
DataDigestType	Increases data integrity. When data digest is enabled, the system performs a checksum over each PDUs data part and verifies using the CRC32C algorithm. <b>Note:</b> Systems that use Intel Nehalem processors offload the iSCSI digest calculations for software iSCSI, thus reducing the impact on performance. Valid values are <code>digestProhibited</code> , <code>digestDiscouraged</code> , <code>digestPreferred</code> , or <code>digestRequired</code> .
HeaderDigest	Increases data integrity. When header digest is enabled, the system performs a checksum over the header part of each iSCSI Protocol Data Unit (PDU) and verifies using the CRC32C algorithm.
MaxOutstandingR2T	Max Outstanding R2T defines the Ready to Transfer (R2T) PDUs that can be in transition before an acknowledgement PDU is received.
FirstBurstLength	Maximum amount of unsolicited data an iSCSI initiator can send to the target during the execution of a single SCSI command, in bytes.
MaxBurstLength	Maximum SCSI data payload in a Data-In or a solicited Data-Out iSCSI sequence, in bytes.
MaxRecvDataSegLen	Maximum data segment length, in bytes, that can be received in an iSCSI PDU.
NoopOutInterval	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. Supported only at the initiator level.
NoopOutTimeout	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 <code>NoopTimeout</code> limit is exceeded, the initiator terminates the current session and starts a new one. Supported only at the initiator level.
RecoveryTimeout	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.
DelayedAck	Allows systems to delay acknowledgment of received data packets.

You can use the following `vicfg-iscsi` options to list parameter options. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- Run `vicfg-iscsi -W -l` to list parameter options for the HBA.

```
vicfg-iscsi <conn_options> -W -l
[-i <stor_ip_addr|stor_hostname> [:<portnum>] [-n <iscsi_name>]] <adapter_name>

vicfg-iscsi <conn_options> --parameter --list
[--ip <stor_ip_addr|stor_hostname> [:<portnum>] [--name <iscsi_name>]] <adapter_name>
```

The target (`-i`) and name (`-n`) option determine what the command applies to.

Option	Result
<code>-i</code> and <code>-n</code>	Command applies to static targets.
Only <code>-i</code>	Command applies to the discovery address.
Neither <code>-i</code> nor <code>-n</code>	Command applies to per-adapter parameters.

- Run `vicfg-iscsi -W -l -k` to list iSCSI parameters and whether they are settable.

```
vicfg-iscsi <conn_options> -W -l -k
[-i <stor_ip_addr|stor_hostname>[:<port_num>] [-n <iscsi_name>]] <adapter_name>

vicfg-iscsi <conn_options> --parameter --list --detail
[--ip <stor_ip_addr|stor_hostname>[:<port_num>][--name <iscsi_name>]] <adapter_name>
```

- Run `vicfg-iscsi -W -j` to set iSCSI parameter options.

```
vicfg-iscsi <conn_options> -W -j <name>=<value>
-i <stor_ip_addr|stor_hostname>[:port_num][<iscsi_name>]] <adapter_name>

vicfg-iscsi <conn_options> --parameter --set <name>=<value>
--ip <stor_ip_addr|stor_hostname>[:port_num][--name <iscsi_name>]] <adapter_name>
```

The target (`-i`) and name (`-n`) option determine what the command applies to.

Option	Result
<code>-i</code> and <code>-n</code>	Command applies to per-target CHAP for static targets.
Only <code>-i</code>	Command applies to the discovery address.
Neither <code>-i</code> nor <code>-n</code>	Command applies to per-adapter CHAP.

If special characters are in the `<name>=<value>` sequence, for example, if you add a space, you must surround the sequence with double quotes (“`<name> = <value>`”).

## Returning Parameters to Default Inheritance

The values of iSCSI parameters associated with a dynamic discovery address or a static discovery target are inherited from the corresponding settings of the parent. For the dynamic discovery address, the parent is the adapter. For the static target, the parent is the adapter or discovery address.

- If you use the vSphere Web Client to modify authentication settings, you deselect the **Inherit from Parent** check box before you can make a change to the discovery address or discovery target.
- If you use `vicfg-iscsi`, the value you set overrides the inherited value.

Inheritance is relevant only if you want to return a dynamic discovery address or a static discovery target to its inherited value. In that case, use the `--reset <param_name>` option, which requires the `--name` option for static discovery addresses, but not for dynamic discovery targets.

```
vicfg-iscsi <conn_options> --parameter --reset <param_name>
--ip <stor_ip_addr | stor_hostname>[:port_num] <adapter_name>
vicfg-iscsi <conn_options> -W -o <param_name>
-i <stor_ip_addr|stor_hostname>[:port_num] <adapter_name>
```

## Enabling iSCSI Authentication

You can enable iSCSI authentication with ESXCLI or with `vicfg-iscsi`.

### Enabling iSCSI Authentication with ESXCLI

The `esxcli iscsi adapter auth` commands enable iSCSI authentication. Mutual authentication is supported for software iSCSI and dependent hardware iSCSI, but not for independent hardware iSCSI (see [“Setting iSCSI CHAP”](#) on page 62).

- 1 (Optional) Set the authentication information for CHAP.

```
esxcli <conn_options> iscsi adapter auth chap set --direction=uni --chap_username=<name>
--chap_password=<pwd> --level=[prohibited, discouraged, preferred, required]
--secret=<string> --adapter=<adapter_name>
```

You can set per-target CHAP for static targets, per-adapter CHAP, or apply the command to the discovery address.

- per-adapter CHAP: `esxcli iscsi adapter auth chap set`
- per-discovery CHAP: `esxcli iscsi adapter discovery sendtarget auth chap set`
- per-target CHAP: `esxcli iscsi adapter target portal auth chap set`

For example:

```
esxcli <conn_options> iscsi adapter auth chap set --direction=uni --chap_username=User1
--chap_password=MySpecialPwd --level=preferred --secret=uni_secret
--adapter=vmhba33
```

- 2 (Optional) Set the authentication information for mutual CHAP by running `esxcli iscsi adapter auth chap set` again with the `-d` option set to `mutual` option and a different authentication user name and secret.

```
esxcli <conn_options> iscsi adapter auth chap set --direction=mutual
--mchap_username=<m_name> --mchap_password=<m_pwd> --level=[prohibited,
required] --secret=<string> --adapter=<adapter_name>
```

For `<level>`, specify `prohibited` or `required`.

- `prohibited` – The host does not use CHAP authentication. If authentication is enabled, specify `chapProhibited` to disable it.
- `required` – The host requires successful CHAP authentication. The connection fails if CHAP negotiation fails. You can set this value for mutual CHAP only if CHAP is set to `chapRequired`.

For direction, specify `mutual`.

---

**IMPORTANT** You are responsible for making sure that CHAP is set before you set mutual CHAP, and for using compatible levels for CHAP and mutual CHAP. Use a different secret in CHAP and mutual CHAP.

---

#### To enable mutual authentication

- 1 Enable authentication.

```
esxcli <conn_options> iscsi adapter auth chap set --direction=uni --chap_username=<name>
--chap_password=<pw> --level=[prohibited, discouraged, preferred, required]
--secret=<string> --adapter=<adapter_name>
```

The specified `chap_username` and `secret` must be supported on the storage side.

- 2 List possible VMkernel NICs to bind.

```
esxcli <conn_options> iscsi logicalnetworkportal list
```

- 3 Enable mutual authentication.

```
esxcli <conn_options> iscsi adapter auth chap set --direction=mutual
--mchap_username=<m_name> --mchap_password=<m_pwd> --level=[prohibited,
required] --secret=<string> --adapter=<adapter_name>
```

The specified `mchap_username` and `secret` must be supported on the storage side.

Make sure the following requirements are met.

- CHAP authentication is already set up when you start setting up mutual CHAP.
- CHAP and mutual CHAP use different user names and passwords. The second user name and password are supported for mutual authentication on the storage side.
- CHAP and mutual CHAP use compatible CHAP levels.

- 4 After setup is complete, perform rediscovery and rescan all storage devices. For example:

```
esxcli <conn_options> iscsi adapter discovery rediscover
esxcli <conn_options> storage core adapter rescan --adapter=vmhba36
```

## Enabling iSCSI Authentication with vicfg-iscsi

The `vicfg-iscsi -A -c` options enable iSCSI authentication. Mutual authentication is supported for software iSCSI and dependent hardware iSCSI, but not for independent hardware iSCSI. See [“Setting iSCSI CHAP”](#) on page 62.

### To enable mutual authentication

- 1 Enable authentication on the ESXi host.

```
vicfg-iscsi <conn_options> -A -c <level> -m <auth_method> -u <auth_u_name> -w <chap_password>
[-i <stor_ip_addr|stor_hostname> [:<portnum>]] [-n <iscsi_name>]] <adapter_name>
```

The specified user name and password must be supported on the storage side.

- 2 Enable mutual authentication on the ESXi host.

```
vicfg-iscsi <conn_options> -A -c <level> -m <auth_method> -b -u <ma_username>
-w <ma_password> [-i <stor_ip_addr|stor_hostname> [:<portnum>]]
[-n <iscsi_name>]] <adapter_name>
```

Make sure the following requirements are met.

- CHAP authentication is already set up when you start setting up mutual CHAP.
- CHAP and mutual CHAP use different user names and passwords. The second user name and password are supported for mutual authentication on the storage side.
- CHAP and mutual CHAP use compatible CHAP levels.

- 3 After setup is complete, perform rediscovery and rescan all storage devices.

## Setting Up Ports for iSCSI Multipathing

With port binding, you create a separate VMkernel port for each physical NIC using 1:1 mapping. You can add all network adapter and VMkernel port pairs to a single vSwitch. The *vSphere Storage* documentation explains in detail how to specify port binding.

You cannot set up ports for multipathing by using `vicfg-iscsi`.

In the examples below, specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

---

**IMPORTANT** The ESX/ESXi 4.x ESXCLI commands for setting up iSCSI are no longer supported.

---

**To specify port binding**

- 1 Find out which uplinks are available for use with iSCSI adapters.  

```
esxcli <conn_options> iscsi physicalnetworkportal list --adapter=<adapter_name>
```
- 2 Connect the software iSCSI or dependent hardware iSCSI initiator to the iSCSI VMkernel ports by running the following command for each port.  

```
esxcli <conn_options> iscsi networkportal nic add --adapter=<adapter_name> --nic=<bound_nic>
```
- 3 Verify that the ports were added to the iSCSI initiator by running the following command:  

```
esxcli <conn_options> iscsi networkportal list --adapter=<adapter_name>
```
- 4 (Optional) If there are active iSCSI sessions between your host and targets, discontinue them. See [“Removing iSCSI Sessions”](#) on page 85.
- 5 Rescan the iSCSI initiator.  

```
esxcli <conn_options> storage core adapter rescan --adapter <iscsi adapter>
```
- 6 To disconnect the iSCSI initiator from the ports, run the following command.  

```
esxcli <conn_options> iscsi networkportal remove --adapter=<adapter_name> --nic=<bound_nic>
```

## Managing iSCSI Sessions

To communicate with each other, iSCSI initiators and targets establish iSCSI sessions. You can use `esxcli iscsi session` to list and manage iSCSI sessions for software iSCSI and dependent hardware iSCSI.

### Introduction to iSCSI Session Management

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 has more than one port, your host can establish multiple sessions. The default number of sessions for each target equals the number of ports on the iSCSI adapter times the number of target ports. You can display all current sessions to analyze and debug them. You might add sessions to the default for several reasons.

- **Cloning sessions.** Some iSCSI arrays support multiple sessions between the iSCSI adapter and target ports. If you clone an existing session on one of these arrays, the array presents more data paths for your adapter. Duplicate sessions do not persist across reboot. Additional sessions to the target might have performance benefits, but the result of cloning depends entirely on the array. You must log out from an iSCSI session if you want to clone a session. You can use the `esxcli iscsi session add` command to clone a session.
- **Enabling Header and Data Digest.** If you are logged in to a session and want to enable the Header and Data Digest parameters, you must set the parameter, remove the session, and add the session back for the parameter change to take effect. You must log out from an iSCSI session if you want to clone a session.
- **Establishing target-specific sessions.** You can 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.



**CAUTION** 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.

The following example scenario uses the available commands. Run `esxcli iscsi session --help` and each command with `--help` for reference information. The example uses a configuration file to log in to the host. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.



---

**IMPORTANT** The ESX/ESXi 4.x ESXCLI commands for managing iSCSI sessions are not supported against ESXi 5.0 hosts.

---

## Listing iSCSI Sessions

- List a software iSCSI session at the adapter level.  

```
esxcli <conn_options> iscsi session list --adapter=<iscsi_adapter>
```
- List a software iSCSI session at the target level.  

```
esxcli <conn_options> iscsi session list --name=<target> --adapter=<iscsi_adapter>
```

## Logging in to iSCSI Sessions

You can use `esxcli iscsi session` to log in to a session. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- Log in to a session on the current software iSCSI or dependent hardware iSCSI configuration at the adapter level.  

```
esxcli <conn_options> iscsi session add --adapter=<adapter_name>
```

For example:

```
esxcli --config /host-config-file iscsi session add --adapter=vmhba36
```
- Log in to a session on the current software iSCSI or dependent hardware iSCSI configuration at the target level.  

```
esxcli <conn_options> iscsi session add --name=<target> --adapter=<adapter_name>
```

For example:

```
esxcli --config /host-config-file iscsi session add -name=iqn.xxx --adapter=vmhba36
```
- Add duplicate sessions with target and session IDs in current software iSCSI or dependent hardware iSCSI configuration.  

```
esxcli <conn_options> iscsi session add --name=<iqn.xxxx> --isid=<session_id>
--adapter=<iscsi_adapter>
```

`iqn.xxxx` is the target IQN, which you can determine by listing all sessions. `session_id` is the session’s iSCSI ID. For example:

```
esxcli --config /host-config-file iscsi session add -name=iqn.xxx --isid='00:02:3d:00:00:01'
--adapter=vmhba36
```

## Removing iSCSI Sessions

You can use `esxcli iscsi session` to remove iSCSI sessions. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- Remove sessions from the current software iSCSI or dependent hardware iSCSI configuration at the adapter level.  

```
esxcli <conn_options> iscsi session remove --adapter=<iscsi_adapter>
```

For example:

```
esxcli iscsi session remove --adapter=vmhba36
```
- Remove sessions from the current software iSCSI or dependent hardware iSCSI configuration at the target level.  

```
esxcli <conn_options> iscsi session remove --name=<iqn> --adapter=<iscsi_adapter>
```

For example:

```
esxcli <conn_options> iscsi session remove --name=iqn.xxx --adapter=vmhba38
```

- Remove sessions from the current software iSCSI or dependent hardware iSCSI configuration with target and session ID.

```
esxcli <conn_options> iscsi session remove --name=<iqn.xxxx> --isid=<session id>  
      --adapter=<iscsi_adapter>
```

`iqn.xxxx` is the target IQN, which you can determine by listing all sessions. `session_id` is the session's iSCSI ID.

For example:

```
esxcli --config /host-config-file iscsi session remove --name=iqn.xxx  
      --session='00:02:3d:01:00:01' --adapter=vmhba36
```

## Managing Third-Party Storage Arrays

VMware partners and customers can optimize performance of their storage arrays in conjunction with VMware vSphere using VMware PSA (pluggable storage architecture). The `esxcli storage core` namespace manages VMware PSA and the `esxcli storage nmp` namespace manages the VMware NMP plug-in.

The *vSphere Storage* documentation discusses PSA functionality in detail and explains how to use the vSphere Client or the vSphere Web Client to manage the PSA, the associated native multipathing plug-in (NMP) and third-party plug-ins.

This chapter uses the following acronyms.

Acronym	Meaning
PSA	Pluggable Storage Architecture.
NMP	Native Multipathing Plugin. Generic VMware multipathing module.
PSP	Path Selection Plugin. Handles path selection for a given device.
SATP	Storage Array Type Plugin. Handles path failover for a given storage array.

This chapter includes these topics:

- [“Managing NMP with `esxcli storage nmp`”](#) on page 87
- [“Path Claiming with `esxcli storage core claiming`”](#) on page 93
- [“Managing Claim Rules”](#) on page 95

### Managing NMP with `esxcli storage nmp`

The NMP (Native Multipathing Plugin) is an extensible multipathing module that ESXi supports by default. You can use `esxcli storage nmp` to manage devices associated with NMP and to set path policies.

The NMP supports all storage arrays listed on the VMware storage Hardware Compatibility List (HCL) and provides a path selection algorithm based on the array type. The NMP associates a set of physical paths with a storage device (LUN). A Storage Array Type Plugin (SATP) determines how path failover is handled for a specific storage array. A Path Selection Plugin (PSP) determines which physical path is used to issue an I/O request to a storage device. SATPs and PSPs are plugins within the NMP plugin.

### Device Management with `esxcli storage nmp device`

The `device` option performs operations on devices currently claimed by the VMware NMP plugin.

### esxcli storage nmp device list

The `list` command lists the devices controlled by VMware NMP and shows the SATP and PSP information associated with each device. To show the paths claimed by NMP, run `esxcli storage nmp path list` to list information for all devices, or for just one device with the `--device` option.

Options	Description
<code>--device &lt;device&gt;</code> <code>-d &lt;device&gt;</code>	Filters the output of the command to show information about a single device. Default is all devices.

### esxcli storage nmp device set

The `set` command sets the Path Selection Policy (PSP) for a device to one of the policies loaded on the system.

Any device can use the PSP assigned to the SATP handling that device, or you can run `esxcli storage nmp device set --device naa.xxx --psp <psp>` to specifically override the PSP assigned to the device.

- If a device does not have a specific PSP set, it always uses the PSP assigned to the SATP. If the default PSP for the SATP changes, the PSP assigned to the device changes only after reboot or after a device is reclaimed. A device is reclaimed when you unclaim all paths for the device and reclaim the paths.
- If you use `esxcli storage nmp device set` to override the SATP's default PSP with a specific PSP, the PSP changes immediately and remains the user-defined PSP across reboots. A change in the SATP's PSP has no effect.

Use the `--default` option to return the device to using the SATP's PSP.

Options	Description
<code>--default</code> <code>-E</code>	Sets the PSP back to the default for the SATP assigned to this device.
<code>--device &lt;device&gt;</code> <code>-d &lt;device&gt;</code>	Device to set the PSP for.
<code>--psp &lt;PSP&gt;</code> <code>-P &lt;PSP&gt;</code>	PSP to assign to the specified device. Call <code>esxcli storage nmp psp list</code> to display all currently available PSPs. See <a href="#">Table 4-1, "Supported Path Policies,"</a> on page 48. See <i>vSphere Storage</i> for a discussion of path policies.

To set the path policy for the specified device to `VMW_PSP_FIXED`, run the following command:

```
esxcli <conn_options> storage nmp device set --device naa.xxx --psp VMW_PSP_FIXED
```

### Listing Paths with esxcli storage nmp path

Use the `path` option to list paths claimed by NMP. By default, the command displays information about all paths on all devices. You can filter in the following ways:

- Only show paths to a single device (`esxcli storage nmp path list --device <device>`).
- Only show information for a single path (`esxcli storage nmp path list --path=<path>`).

To list devices, call `esxcli storage nmp device list`.

### Managing Path Selection Policy Plugins with esxcli storage nmp psp

Use `esxcli storage nmp psp` to manage VMware path selection policy plugins included with the VMware NMP plugin and to manage third-party PSPs.

---

**IMPORTANT** When used with third-party PSPs, the syntax depends on the third-party PSP implementation.

---

## Retrieving PSP Information

The `esxcli storage nmp psp generic deviceconfig get` and `esxcli storage nmp psp generic pathconfig get` command retrieves PSP configuration parameters. The type of PSP determines which command to use.

- Use `nmp psp generic deviceconfig get` for PSPs that are set to `VMW_PSP_RR`, `VMW_PSP_FIXED` or `VMW_PSP_MRU`.
- Use `nmp psp generic pathconfig get` for PSPs that are set to `VMW_PSP_FIXED` or `VMW_PSP_MRU`. No path configuration information is available for `VMW_PSP_RR`.

To retrieve PSP configuration parameters, use the appropriate command for the PSP.

- Device configuration information.

```
esxcli <conn_options> storage nmp psp generic deviceconfig get --device=<device>
esxcli <conn_options> storage nmp psp fixed deviceconfig get --device=<device>
esxcli <conn_options> storage nmp psp roundrobin deviceconfig get --device=<device>
```

- Path configuration information.

```
esxcli <conn_options> storage nmp psp generic pathconfig get --path=<path>
```

- Retrieve the PSP configuration for the specified path.

```
esxcli <conn_options> nmp psp pathconfig generic get --path vmhba4:C1:T2:L23
```

The `esxcli storage nmp psp list` command shows the list of Path Selection Plugins on the system and a brief description of each plugin.

## Setting Configuration Parameters for Third-Party Extensions

The `esxcli storage nmp psp generic deviceconfig set` and `esxcli storage nmp psp generic pathconfig set` commands support future third-party PSA expansion. The `setconfig` command sets PSP configuration parameters for those third-party extensions.

---

**NOTE** The precise results of these commands depend on the third-party extension. See the extension documentation for information.

---

Use `esxcli storage nmp roundrobin setconfig` for other path policy configuration. See [“Customizing Round Robin Setup”](#) on page 90.

You can run `esxcli storage nmp psp generic deviceconfig set --device=<device>` to specify PSP information for a device, and `esxcli storage nmp psp generic pathconfig set --path=<path>` to specify PSP information for a path. For each command, use `--config` to set the specified configuration string.

Options	Description
<code>--config &lt;config_string&gt;</code> <code>-c &lt;config_string&gt;</code>	Configuration string to set for the device or path specified by <code>--device</code> or <code>--path</code> . See <a href="#">Table 4-1, “Supported Path Policies,”</a> on page 48.
<code>--device &lt;device&gt;</code> <code>-d &lt;device&gt;</code>	Device for which you want to customize the path policy.
<code>--path &lt;path&gt;</code> <code>-p &lt;path&gt;</code>	Path for which you want to customize the path policy.

## Fixed Path Selection Policy Operations

The `fixed` option gets and sets the preferred path policy for NMP devices configured to use `VMW_PSP_FIXED`.

### Retrieving the Preferred Path

The `esxcli storage nmp fixed deviceconfig get` command retrieves the preferred path on a specified device that is using NMP and the `VMW_PSP_FIXED` PSP.

Options	Description
<code>--device &lt;device&gt;</code> <code>-d &lt;device&gt;</code>	Device for which you want to get the preferred path. This device must be controlled by the VMW_PSP_FIXED PSP.

To return the path configured as the preferred path for the specified device, run the following command. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

```
esxcli <conn_options> storage nmp fixed deviceconfig get --device naa.xxx
```

### Setting the Preferred Path

The `esxcli storage nmp fixed deviceconfig set` command sets the preferred path on a specified device that is using NMP and the VMW\_PSP\_FIXED path policy.

Options	Description
<code>--device &lt;device&gt;</code> <code>-d &lt;device&gt;</code>	Device for which you want to set the preferred path. This device must be controlled by the VMW_PSP_FIXED PSP. Use <code>esxcli storage nmp device --list</code> to list the policies for all devices.
<code>--path &lt;path&gt;</code> <code>-p &lt;path&gt;</code>	Path to set as the preferred path for the specified device.

To set the preferred path for the specified device to `vmhba3:C0:T5:L3`, run the following command. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

```
esxcli <conn_options> storage nmp fixed deviceconfig set --device naa.xxx --path vmhba3:C0:T5:L3
```

## Customizing Round Robin Setup

The `esxcli storage nmp psp roundrobin` commands sets round robin path options on a device controlled by the VMW\_PSP\_RR PSP. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

### To specify and customize round robin path policies

- 1 Set the path policy to round robin.

```
esxcli <conn_options> storage nmp device set --device naa.xxx --psp VMW_PSP_RR
```

- 2 Specify when to switch paths.

You can choose the number of I/O operations, number of bytes, and so on. For example:

```
esxcli <conn_options> storage nmp psp roundrobin deviceconfig set --type "bytes" -B 12345 --device naa.xxx
```

Sets the device specified by `--device` to switch to the next path each time 12345 bytes have been sent along the current path.

```
esxcli <conn_options> storage nmp psp roundrobin deviceconfig set --type=iops --iops 4200 --device naa.xxx
```

Sets the device specified by `--device` to switch after 4200 I/O operations have been performed on a path.

### Retrieving Path Selection Settings

The `esxcli storage nmp psp roundrobin deviceconfig get` command retrieves path selection settings for a device that is using the roundrobin PSP. You can specify the device to retrieve the information for.

Options	Description
<code>-d &lt;device&gt;</code> <code>--device &lt;device&gt;</code>	Device to get round robin properties for.

## Specifying Conditions for Path Changes

The `esxcli storage nmp psp roundrobin deviceconfig set` command specifies under which conditions a device that is using the `VMW_PSP_RR` PSP changes to a different path. You can use `--bytes` or `--iops` to specify when the path should change.

Options	Description
<code>--bytes</code> <code>-B</code>	Number of bytes to send along one path for this device before the PSP switches to the next path. You can use this option only when <code>--type</code> is set to <code>bytes</code> .
<code>--device</code> <code>-d</code>	Device to set round robin properties for. This device must be controlled by the round robin ( <code>VMW_PSP_RR</code> ) PSP.
<code>--iops</code> <code>-I</code>	Number of I/O operations to send along one path for this device before the PSP switches to the next path. You can use this option only when <code>--type</code> is set to <code>iops</code> .
<code>--type</code> <code>-t</code>	Type of round robin path switching to enable for this device. The following values for <code>type</code> are supported: <ul style="list-style-type: none"> <li>■ <code>bytes</code>: Set the trigger for path switching based on the number of bytes sent down a path.</li> <li>■ <code>default</code>: Set the trigger for path switching back to default values.</li> <li>■ <code>iops</code>: Set the trigger for path switching based on the number of I/O operations on a path.</li> </ul> An equal sign (=) before the type or double quotes around the type are optional.
<code>--useANO</code> <code>-U</code>	If set to 1, the round robin PSP includes paths in the active, unoptimized state in the round robin set. If set to 0, the PSP uses active, unoptimized paths only if no active optimized paths are available. Otherwise, the PSP includes only active optimized paths in the round robin path set.

## Managing SATPs

The `esxcli storage nmp satp` commands manage SATPs. You can use these commands to perform the following tasks:

- Retrieve and set configuration parameters
- Add and remove rules from the list of claim rules for a specified SATP
- Set the default PSP for a specified SATP
- List SATPs that are currently loaded into NMP and the associated claim rules

The default SATP for an active-active FC array with a vendor and model not listed in the SATP rules is `VMW_SATP_DEFAULT_AA`.

### Retrieving Information About SATPs

The `esxcli storage nmp satp list` command lists the SATPs that are currently available to the NMP system and displays information about those SATPs. This command supports no options and displays information about these SATPs.

```
esxcli <conn_options> storage nmp satp list
```

The `rule list` command lists the claim rules for SATPs.

```
esxcli <conn_options> storage nmp satp rule list
```

### Adding SATP Rules

Claim rules specify that a storage device that uses a certain driver or transport or has a certain vendor or model should use a certain SATP. The `esxcli storage nmp satp rule add` command adds a rule that performs such a mapping to the list of claim rules. The options you specify define the rule. For example, the following command specifies that if a path has vendor `VMWARE` and model `Virtual`, the PSA assigns it to the `VMW_SATP_LOCAL` SATP.

```
esxcli <conn_options> storage nmp satp rule add --satp="VMW_SATP_LOCAL" --vendor="VMWARE"
--model="Virtual" --description="VMware virtual disk"
```

Option	Description
--driver -D	Driver string to set when adding the SATP claim rule.
--device -d	Device to set when adding SATP claim rules. Device rules are mutually exclusive with vendor/model and driver rules.
--force -f	Force claim rules to ignore validity checks and install the rule even if checks fail.
--model -M	Model string to set when adding the SATP claim rule. Can be the model name or a pattern <code>^mod*</code> , which matches all devices that start with <code>mod</code> . That is, the pattern successfully matches <code>mod1</code> and <code>modz</code> , but not <code>mymod1</code> .  The command supports the start/end (^) and wildcard (*) functionality but no other regular expressions.
--transport -R	Transport string to set when adding the SATP claim rule. Describes the type of storage HBA, for example, <code>iscsi</code> or <code>fc</code> .
--vendor -V	Vendor string to set when adding the SATP claim rule.
--satp -s	SATP for which the rule is added.
--claim-option -c	Claim option string to set when adding the SATP claim rule.
--description -e	Description string to set when adding the SATP claim rule.
--option -o	Option string to set when adding the SATP claim rule. Surround the option string in double quotes, and use a space, not a comma, when specifying more than one option. "enable_local enable_ssd".
--psp -P	Default PSP for the SATP claim rule.
--psp-option -O	PSP options for the SATP claim rule.
--type -t	Set the claim type when adding a SATP claim rule.

The following examples illustrate adding SATP rules. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- Add a SATP rule that specifies that disks with vendor string `VMWARE` and model string `Virtual` should be added to `VMW_SATP_LOCAL`.

```
esxcli <conn_options> storage nmp satp rule add --satp="VMW_SATP_LOCAL" --vendor="VMWARE"
--model="Virtual" --description="VMware virtual disk"
```

- Add a SATP rule that specifies that disks with the driver string `somedriver` should be added to `VMW_SATP_LOCAL`.

```
esxcli <conn_options> storage nmp satp rule add --satp="VMW_SATP_LOCAL"
--driver="somedriver"
```

- Add a rule that specifies that all storage devices with vendor string `ABC` and a model name that starts with `120` should use `VMW_SATP_DEFAULT_AA` (`VMW_SATP_DEFAULT_AA` is an example).

```
esxcli <conn_options> storage nmp satp rule add --satp VMW_SATP_DEFAULT_AA --vendor="ABC"
--model="^120"
```

## Removing SATP Rules

The `esxcli storage nmp satp rule remove` command removes an existing SATP rule. The options you specify define the rule to remove. The options listed for [“Adding SATP Rules”](#) on page 91 are supported.



The following example removes the rule that assigns devices with vendor string `VMWARE` and model string `Virtual` to `VMW_SATP_LOCAL`.

```
esxcli <conn_options> storage nmp satp rule remove
    --satp="VMW_SATP_LOCAL" --vendor="VMWARE" --model="Virtual"
```

## Retrieving and Setting SATP Configuration Parameters

The `esxcli storage nmp satp generic deviceconfig get` and `esxcli storage nmp satp generic pathconfig get` commands retrieve per-device or per-path SATP configuration parameters. You cannot retrieve paths or devices for all SATPs, you must retrieve the information one path or one device at a time.

Use this command to retrieve per device or per path SATP configuration parameters, and to see whether you can set certain configuration parameters for a device or path.

For example `esxcli storage nmp satp generic deviceconfig get --device naa.xxx` might return `SATP VMW_SATP_LSI does not support device configuration`.

`esxcli storage nmp satp generic pathconfig get -path vmhba1:C0:T0:L8` might return `INIT,AVT OFF,v5.4,DUAL ACTIVE,ESX FAILOVER`

The `esxcli storage nmp satp generic deviceconfig set` and `esxcli storage nmp satp generic pathconfig set` commands set configuration parameters for SATPs that are loaded into the system, if they support device configuration. You can set per-path or per-device SATP configuration parameters.

---

**IMPORTANT** The command passes the configuration string to the SATP associated with that device or path.

---

The configuration strings might vary by SATP. VMware supports a fixed set of configuration strings for a subset of its SATPs. The strings might change in future releases.

Options	Description
<code>--config</code> <code>-c</code>	Configuration string to set for the path specified by <code>--path</code> or the device specified by <code>--device</code> . You can set the configuration for the following SATPs: <ul style="list-style-type: none"> <li>■ <code>VMW_SATP_ALUA_CX</code></li> <li>■ <code>VMW_SATP_ALUA</code></li> <li>■ <code>VMW_SATP_CX</code></li> <li>■ <code>VMW_SATP_INV</code></li> </ul> You can specify one of the following device configuration strings: <ul style="list-style-type: none"> <li>■ <code>navireg_on</code> – starts automatic registration of the device with Navisphere.</li> <li>■ <code>navireg_off</code> – stops the automatic registration of the device.</li> <li>■ <code>ipfilter_on</code> – stops the sending of the host name for Navisphere registration. Used if host is known as <code>localhost</code>.</li> <li>■ <code>ipfilter_off</code> – enables the sending of the host name during Navisphere registration.</li> </ul>
<code>--device</code> <code>-d</code>	Device to set SATP configuration for. Not all SATPs support the <code>setconfig</code> option on devices.
<code>--path</code> <code>-p</code>	Path to set SATP configuration for. Not all SATPs support the <code>setconfig</code> option on paths.

Run `esxcli storage nmp device set --default --device=<device>` to set the PSP for the specified device back to the default for the assigned SATP for this device.

## Path Claiming with `esxcli storage core claiming`

The `esxcli storage core claiming` namespace includes a number of troubleshooting commands. These commands are not persistent and are useful only to developers who are writing PSA plugins or troubleshooting a system. If I/O is active on the path, `unclaim` and `reclaim` actions fail.

---

**IMPORTANT** The help for `esxcli storage core claiming` includes the `autoclaim` command. Do not use this command unless instructed to do so by VMware support staff.

---

## Using the Reclaim Troubleshooting Command

The `esxcli storage core claiming reclaim` troubleshooting command is intended for PSA plugin developers or administrators who troubleshoot PSA plugins. The command proceeds as follows.

- Attempts to unclaim all paths to a device.
- Runs the loaded claim rules on each of the unclaimed paths to reclaim those paths.

It is normal for this command to fail if a device is in use.

---

**IMPORTANT** The `reclaim` command unclaims paths associated with a device.

You cannot use the command to reclaim paths currently associated with the `MASK_PATH` plugin because `--device` is the only option for `reclaim` and `MASK_PATH` paths are not associated with a device.

You can use the command to unclaim paths for a device and have those paths reclaimed by the `MASK_PATH` plugin.

---

Options	Description
<code>--device &lt;device&gt;</code> <code>-d &lt;device&gt;</code>	Name of the device on which all paths are reclaimed.
<code>--help</code>	Displays the help message.

---

## Unclaiming Paths or Sets of Paths

The `esxcli storage core claiming unclaim` command unclaims a path or set of paths, disassociating those paths from a PSA plugin. The command fails if the device is in use.

You can unclaim only active paths with no outstanding requests. You cannot unclaim the ESXi USB partition or devices with VMFS volumes on them. It is therefore normal for this command to fail, especially when you specify a plugin or adapter to unclaim.

Unclaiming does not persist. Periodic path claiming reclaims unclaimed paths unless claim rules are configured to mask a path. See the *vSphere Storage* documentation for details.

---

**IMPORTANT** The `unclaim` command unclaims paths associated with a device. You can use this command to unclaim paths associated with the `MASK_PATH` plugin but cannot use the `--device` option to unclaim those paths.

---

Options	Description
<code>--adapter &lt;adapter&gt;</code> <code>-A &lt;adapter&gt;</code>	If <code>--type</code> is set to <code>location</code> , specifies the name of the HBA for the paths that you want to unclaim. If you do not specify this option, unclaiming runs on paths from all adapters.
<code>--channel &lt;channel&gt;</code> <code>-C &lt;channel&gt;</code>	If <code>--type</code> is set to <code>location</code> , specifies the SCSI channel number for the paths that you want to unclaim. If you do not specify this option, unclaiming runs on paths from all channels.
<code>--claimrule-class &lt;cl&gt;</code> <code>-c &lt;cl&gt;</code>	Claim rule class to use in this operation. You can specify <code>MP</code> (Multipathing), <code>Filter</code> , or <code>VAAI</code> . Multipathing is the default. <code>Filter</code> is used only for <code>VAAI</code> . Specify claim rules for both <code>VAAI_FILTER</code> and <code>VAAI</code> plugin to use it.
<code>--device &lt;device&gt;</code> <code>-d &lt;device&gt;</code>	If <code>--type</code> is set to <code>device</code> , attempts to unclaim all paths to the specified device. If there are active I/O operations on the specified device, at least one path cannot be unclaimed.
<code>--driver &lt;driver&gt;</code> <code>-D &lt;driver&gt;</code>	If <code>--type</code> is <code>driver</code> , unclaims all paths specified by this HBA driver.
<code>--lun &lt;lun_number&gt;</code> <code>-L &lt;lun_number&gt;</code>	If <code>--type</code> is <code>location</code> , specifies the SCSI LUN for the paths to unclaim. If you do not specify <code>--lun</code> , unclaiming runs on paths with any LUN number.
<code>--model &lt;model&gt;</code> <code>-m &lt;model&gt;</code>	If <code>--type</code> is <code>vendor</code> , attempts to unclaim all paths to devices with specific model information (for multipathing plugins) or unclaim the device itself (for filter plugins). If there are active I/O operations on this device, at least one path fails to unclaim.

---

Options	Description
<code>--path &lt;path&gt;</code> <code>-p &lt;path&gt;</code>	If <code>--type</code> is <code>path</code> , unclaims a path specified by its path UID or runtime name.
<code>--plugin &lt;plugin&gt;</code> <code>-P</code>	If <code>--type</code> is <code>plugin</code> , unclaims all paths for a specified multipath plugin. <code>&lt;plugin&gt;</code> can be any valid PSA plugin on the system. By default only NMP and MASK_PATH are available, but additional plugins might be installed.
<code>--target &lt;target&gt;</code> <code>-T &lt;target&gt;</code>	If <code>--type</code> is <code>location</code> , unclaims the paths with the SCSI target number specified by <code>target</code> . If you do not specify <code>--target</code> , unclaiming runs on paths from all targets.
<code>--type &lt;type&gt;</code> <code>-t &lt;type&gt;</code>	Type of unclaim operation to perform. Valid values are <code>location</code> , <code>path</code> , <code>driver</code> , <code>device</code> , <code>plugin</code> , and <code>vendor</code> .
<code>--vendor &lt;vendor&gt;</code> <code>-v &lt;vendor&gt;</code>	If <code>--type</code> is <code>vendor</code> , attempts to unclaim all paths to devices with specific vendor info for multipathing plugins or unclaim the device itself for filter plugins. If there are any active I/O operations on this device, at least one path fails to unclaim.

The following troubleshooting command tries to unclaim all paths on `vmhba1`.

```
esxcli <conn_options> storage core claiming unclaim --type location -A vmhba1
```

Run `vicfg-mpath <conn_options> -l` to verify that the command succeeded.

If a path is the last path to a device that was in use, or a if a path was very recently in use, the unclaim operation might fail. An error is logged that not all paths could be unclaimed. Stop processes that might use the device and wait 15 seconds to let the device be quiesced. Retry the command.

## Managing Claim Rules

The PSA uses claim rules to determine which multipathing module should claim the paths to a particular device and to manage the device. `esxcli storage core claimrule` manages claim rules.

Claim rule modification commands do not operate on the VMkernel directly. Instead they operate on the configuration file by adding and removing rules. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

### To change the current claim rules in the VMkernel

- 1 Run one or more of the `esxcli storage core claimrule` modification commands (`add`, `remove`, or `move`).
- 2 Run `esxcli storage core claimrule load` to replace the current rules in the VMkernel with the modified rules from the configuration file.

You can also run `esxcli storage core plugin list` to list all loaded plugins.

## Adding Claim Rules

The `esxcli storage core claimrule add` command adds a claim rule to the set of claim rules on the system. You can use this command to add new claim rules or to mask a path using the MASK\_PATH claim rule. You must load the rules after you add them.

Options	Description
<code>--adapter &lt;adapter&gt;</code> <code>-A &lt;adapter&gt;</code>	Adapter of the paths to use. Valid only if <code>--type</code> is <code>location</code> .
<code>--autoassign</code> <code>-u</code>	Adds a claim rule based on its characteristics. The rule number is not required.
<code>--channel &lt;channel&gt;</code> <code>-C &lt;channel&gt;</code>	Channel of the paths to use. Valid only if <code>--type</code> is <code>location</code> .

Options	Description
<code>--claimrule-class &lt;cl&gt;</code> <code>-c &lt;cl&gt;</code>	Claim rule class to use in this operation. You can specify MP (default), Filter, or VAAI. To configure hardware acceleration for a new array, add two claim rules, one for the VAAI filter and another for the VAAI plugin. See <i>vSphere Storage</i> for detailed instructions.
<code>--driver &lt;driver&gt;</code> <code>-D &lt;driver&gt;</code>	Driver for the HBA of the paths to use. Valid only if <code>--type</code> is <code>driver</code> .
<code>--force</code> <code>-f</code>	Force claim rules to ignore validity checks and install the rule.
<code>--lun &lt;lun_number&gt;</code> <code>-L &lt;lun_number&gt;</code>	LUN of the paths to use. Valid only if <code>--type</code> is <code>location</code> .
<code>--model &lt;model&gt;</code> <code>-M &lt;model&gt;</code>	Model of the paths to use. Valid only if <code>--type</code> is <code>vendor</code> . Valid values are values of the <code>Model</code> string from the SCSI inquiry string. Run <code>vicfg-scsidevs &lt;conn_options&gt; -l</code> on each device to see model string values.
<code>--plugin</code> <code>-p</code>	PSA plugin to use. Currently, the values are <code>NMP</code> or <code>MASK_PATH</code> , but third parties can ship their own PSA plugins in the future. <code>MASK_PATH</code> refers to the plugin <code>MASK_PATH_PLUGIN</code> . The command adds claimrules for this plugin if the user wants to mask the path. ESX 3.5 includes the <code>MaskLUNs</code> advanced configuration option. This option is not available in ESX/ESXi 4.x and ESXi 5.0. It has been replaced by the <code>MASK_PATH_PLUGIN</code> . You can add a claim rule that causes the <code>MASK_PATH_PLUGIN</code> to claim the path to mask a path or LUN from the host. See the <i>vSphere Storage</i> documentation for details.
<code>--rule &lt;rule_ID&gt;</code> <code>-r &lt;rule_ID&gt;</code>	Rule ID to use. Run <code>esxcli storage core claimrule list</code> to see the rule ID. The rule ID indicates the order in which the claim rule is to be evaluated. User-defined claim rules are evaluated in numeric order starting with 101.
<code>--target &lt;target&gt;</code> <code>-T &lt;target&gt;</code>	Target of the paths to use. Valid only if <code>--type</code> is <code>location</code> .
<code>--transport &lt;transport&gt;</code> <code>-R &lt;transport&gt;</code>	Transport of the paths to use. Valid only if <code>--type</code> is <code>transport</code> . The following values are supported: <ul style="list-style-type: none"> <li>■ <code>block</code> – block storage</li> <li>■ <code>fc</code> – FibreChannel</li> <li>■ <code>iscsivendor</code> – iSCSI</li> <li>■ <code>iscsi</code> – not currently used</li> <li>■ <code>ide</code> – IDE storage</li> <li>■ <code>sas</code> – SAS storage</li> <li>■ <code>sata</code> – SATA storage</li> <li>■ <code>usb</code> – USB storage</li> <li>■ <code>parallel</code> – parallel</li> <li>■ <code>unknown</code></li> </ul>
<code>--type &lt;type&gt;</code> <code>-t &lt;type&gt;</code>	Type of matching to use for the operation. Valid values are <code>vendor</code> , <code>location</code> , <code>driver</code> , and <code>transport</code> .
<code>--vendor</code> <code>-V</code>	Vendor of the paths to use. Valid only if <code>--type</code> is <code>vendor</code> . Valid values are values of the <code>vendor</code> string from the SCSI inquiry string. Run <code>vicfg-scsidevs &lt;conn_options&gt; -l</code> on each device to see vendor string values.
<code>--wwnn</code>	World-Wide Node Number for the target to use in this operation.
<code>--wwpn</code>	World-Wide Port Number for the target to use in this operation.

Claim rules are numbered as follows.

- Rules 0–100 are reserved for internal use by VMware.
- Rules 101–65435 are available for general use. Any third party multipathing plugins installed on your system use claim rules in this range. By default, the PSA claim rule 101 masks Dell array pseudo devices. Do not remove this rule, unless you want to unmask these devices.

- Rules 65436–65535 are reserved for internal use by VMware.

When claiming a path, the PSA runs through the rules starting from the lowest number and determines if a path matches the claim rule specification. If the PSA finds a match, it gives the path to the corresponding plugin. This is worth noticing because a given path might match several claim rules.

The following examples illustrate adding claim rules. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- Add rule 321, which claims the path on adapter `vmhba0`, channel `0`, target `0`, LUN `0` for the NMP plugin.
 

```
esxcli <conn_options> storage core claimrule add -r 321 -t location -A vmhba0 -C 0 -T 0 -L 0 -P NMP
```
- Add rule 429, which claims all paths provided by an adapter with the `mptscsi` driver for the `MASK_PATH` plugin.
 

```
esxcli <conn_options> storage core claimrule add -r 429 -t driver -D mptscsi -P MASK_PATH
```
- Add rule 914, which claims all paths with vendor string `VMWARE` and model string `Virtual` for the NMP plugin.
 

```
esxcli <conn_options> storage core claimrule add -r 914 -t vendor -V VMWARE -M Virtual -P NMP
```
- Add rule 1015, which claims all paths provided by FC adapters for the NMP plugin.
 

```
esxcli <conn_options> storage core claimrule add -r 1015 -t transport -R fc -P NMP
```

## Converting ESX 3.5 LUN Masks to Claim Rule Format

The `esxcli storage core claimrule convert` command converts LUN masks in ESX 3.5 format (`/adv/Disk/MaskLUNs`) to claim rule format. The command writes the converted list and erases the old LUN mask data. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

### To convert ESX 3.5 format LUN masks to claim rule format

- 1 Run `esxcli storage core claimrule convert` without options.

That call returns `No /adv/Disk/MaskLUNs config entry to convert` or displays the list of claim rules that would result from the conversion. For example:

Rule	Plugin	HbaName	Controller	Target	LUN
120	MASK_PATH	vmhba11	0	0	11
121	MASK_PATH	vmhba11	0	0	10
122	MASK_PATH	vmhba4	0	2	1

- 2 Run `esxcli storage core claimrule convert --commit` to actually commit the change.

When you convert LUN masking to the claim rule format after an upgrade from ESX/ESXi 3.5 to ESX/ESXi 4.x, this command converts the `/adv/Disk/MaskLUNs` advanced configuration entry in the `esx.conf` file to claim rules with `MASK_PATH` as the plug-in.

---

**IMPORTANT** This conversion does not work for all input Mask LUN variations. For example, role conversion for software iSCSI LUNs is not supported.

Inspect the list of generated claim rules carefully before you commit them by using `--commit`.

---

**Table 6-1.** `esxcli storage core claimrule convert` Options

Options	Description
<code>--commit</code>	Forces LUN mask configuration changes to be saved. If you call the command without this parameter, changes are not saved, and you can first inspect the generated claim rules.
<code>-C</code>	

## Removing Claim Rules

The `esxcli storage core claimrule remove` command removes a claim rule from the set of claim rules on the system.

---

**IMPORTANT** By default, the PSA claim rule 101 masks Dell array pseudo devices. Do not remove this rule, unless you want to unmask these devices.

---

Option	Description
<code>--rule &lt;rule_ID&gt;</code> <code>-r &lt;rule_ID&gt;</code>	ID of the rule to be removed. Run <code>esxcli storage core claimrule list</code> to see the rule ID.

The following example removes rule 1015.

```
esxcli <conn_options> storage core claimrule remove -r 1015
```

## Listing Claim Rules

The `list` command lists all claim rules on the system. You can specify the claim rule class as an argument.

Option	Description
<code>--claimrule-class &lt;cl&gt;</code> <code>-c &lt;cl&gt;</code>	Claim rule class to use in this operation. You can specify MP (Multipathing), Filter, or VAAI. Multipathing is the default. Filter is used only for VAAI. Specify claim rules for both VAAI_FILTER and VAAI plugin to use it. See <i>vSphere Storage</i> for information about VAAI.

You can run the command as follows. The equal sign is optional, so both forms of the command have the same result. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

```
esxcli <conn_options> storage core claimrule list -c Filter
esxcli <conn_options> storage core claimrule list --claimrule-class=Filter
```

## Loading Claim Rules

The `esxcli storage core claimrule load` command loads claim rules from the `esx.conf` configuration file into the VMkernel. Developers and experienced storage administrators might use this command for boot time configuration.

This command has no options; it always loads all claim rules from `esx.conf`.

## Moving Claim Rules

The `esxcli storage core claimrule move` command moves a claim rule from one rule ID to another.

Options	Description
<code>--claimrule-class &lt;cl&gt;</code> <code>-c &lt;cl&gt;</code>	Claim rule class to use in this operation.
<code>--new-rule &lt;rule_ID&gt;</code> <code>-n &lt;rule_ID&gt;</code>	New rule ID you want to give to the rule specified by the <code>--rule</code> option.
<code>--rule &lt;rule_ID&gt;</code> <code>-r &lt;rule_ID&gt;</code>	ID of the rule to be removed. Run <code>esxcli storage core claimrule list</code> to display the rule ID.

The following example renames rule 1016 to rule 1015 and removes rule 1016. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

```
esxcli <conn_options> storage core claimrule move -r 1015 -n 1016
```

## Running Path Claiming Rules

The `esxcli storage core claimrule run` command runs path claiming rules. Run this command apply claim rules that are loaded. If you do not call `run`, the system checks for claim rule updates every five minutes and applies them. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

### To load and apply claim rules

- 1 Modify rules and load them.

```
esxcli <conn_options> storage core claimrule load
```

- 2 Quiesce the devices that use paths for which you want to change the rule and unclaim those paths.

```
esxcli <conn_options> storage core claiming unclaim --device=<device>
```

- 3 Run path claiming rules.

```
esxcli <conn_options> storage core claimrule run
```

This command is also used for troubleshooting and boot time configuration.

Options	Description
<code>--adapter &lt;adapter&gt;</code> <code>-A &lt;adapter&gt;</code>	If <code>--type</code> is <code>location</code> , name of the HBA for the paths to run the claim rules on. To run claim rules on paths from all adapters, omit this option.
<code>--channel &lt;channel&gt;</code> <code>-C &lt;channel&gt;</code>	If <code>--type</code> is <code>location</code> , value of the SCSI channel number for the paths to run the claim rules on. To run claim rules on paths with any channel number, omit this option.
<code>--claimrule-class</code> <code>-c</code>	Claim rule class to use in this operation.
<code>--lun &lt;lun&gt;</code> <code>-L &lt;lun&gt;</code>	If <code>--type</code> is <code>location</code> , value of the SCSI LUN for the paths to run claim rules on. To run claim rules on paths with any LUN, omit this option.
<code>--path &lt;path_UID&gt;</code> <code>-p &lt;path_UID&gt;</code>	If <code>--type</code> is <code>path</code> , this option indicates the unique path identifier (UID) or the runtime name of a path to run claim rules on.
<code>--target &lt;target&gt;</code> <code>-T &lt;target&gt;</code>	If <code>--type</code> is <code>location</code> , value of the SCSI target number for the paths to run claim rules on. To run claim rules on paths with any target number, omit this option.
<code>--type</code> <code>&lt;location path all&gt;</code> <code>-t &lt;location path all&gt;</code>	Type of claim to perform. By default, uses <code>all</code> , which means claim rules run without restriction to specific paths or SCSI addresses. Valid values are <code>location</code> , <code>path</code> , and <code>all</code> .
<code>--wait</code> <code>-w</code>	You can use this option only if you also use <code>--type all</code> . If the option is included, the claim waits for paths to settle before running the claim operation. In that case, the system does not start the claiming process until it is likely that all paths on the system have appeared before starting the claim process. After the claiming process has started, the command does not return until device registration has completed. If you add or remove paths during the claiming or the discovery process, this option might not work correctly.





# Managing Users

---

An ESXi system grants access to its resources when a known user with appropriate permissions logs on to the system with a password that matches the one stored for that user. You can use the vSphere Client or the vSphere SDK for all user management tasks. You cannot create ESXi users with the vSphere Web Client.

You can use the `vicfg-user` command to create, modify, delete, and list local direct access users on an ESXi host. You cannot run this command against a vCenter Server system.

---

**IMPORTANT** Starting with vSphere 5.1, you can no longer manage groups with `vicfg-user`.

---

This chapter includes the following topics:

- [“Users in the vSphere Environment”](#) on page 101
- [“vicfg-user Command Syntax”](#) on page 101
- [“Managing Users with vicfg-user”](#) on page 102
- [“Assigning Permissions with ESXCLI”](#) on page 104

## Users in the vSphere Environment

Users, and roles control who has access to vSphere components and what actions each user can perform. User management is discussed in detail in the *vSphere Security* documentation.

---

**IMPORTANT** You cannot use `vicfg-user` to create roles. You can manage system-defined roles.

---

vCenter Server and ESXi systems authenticate a user with a combination of user name, password, and permissions. Servers and hosts maintain lists of authorized users and the permissions assigned to each user.

Privileges define basic individual rights that are required to perform actions and retrieve information. ESXi and vCenter Server use sets of privileges, or roles, to control which users can access particular vSphere objects. ESXi and vCenter Server provide a set of pre-established roles.

The privileges and roles assigned on an ESXi host are separate from the privileges and roles assigned on a vCenter Server system. When you manage a host by using vCenter Server system, only the privileges and roles assigned through the vCenter Server system are available. If you connect directly to the host by using the vSphere Client, only the privileges and roles assigned directly on the host are available. You cannot create ESXi users with the vSphere Web Client.

## vicfg-user Command Syntax

The `vicfg-user` syntax differs from other vCLI commands. You specify operations as follows:

```
vicfg-user <conn_options> -e <user> -o <add|modify|delete|list>
```

If you create a user without specifying the role (`--role`), the user has no permissions. You cannot change the user's role, you can only change the user's permission.

---

**IMPORTANT** If you create a user with the vSphere Client, you cannot make changes to that user with the `vicfg-user` command.

---

### Options

The `vicfg-user` command-specific options manipulate users. You must also specify connection options. See [“Connection Options for vCLI Host Management Commands”](#) on page 18.

Option	Description
<code>--adduser &lt;user_list&gt;</code> <code>-u &lt;user_list&gt;</code>	Adds the specified users. Takes a comma-separated list of users.
<code>--entity &lt;user&gt;</code> <code>-e &lt;user&gt;</code>	Entity to perform the operation on. Starting with vSphere 5.1, <code>entity</code> is always <code>user</code> .
<code>--login &lt;login_id&gt;</code> <code>-l &lt;login_id&gt;</code>	Login ID of the user.
<code>--newpassword &lt;p_wd&gt;</code> <code>-p &lt;p_wd&gt;</code>	Password for the target user.
<code>--newuserid &lt;UUID&gt;</code> <code>-i &lt;UUID&gt;</code>	New UUID for the target user.
<code>--newusername &lt;name&gt;</code> <code>-n &lt;name&gt;</code>	New user name for the target user.
<code>--operation</code> <code>-o</code>	Operation to perform. Specify <code>add</code> , <code>modify</code> , <code>delete</code> , or <code>list</code> .
<code>--role &lt;admin read-only no-access&gt;</code> <code>-r &lt;admin read-only no-access&gt;</code>	Role for the target user. Specify one of <code>admin</code> , <code>read-only</code> , or <code>no-access</code> . Users that you create without assigning permissions have no permissions.
<code>--shell</code> <code>-s</code>	Grant shell access to the target user. Default is no shell access. Use this command to change the default or to revoke shell access rights after they have been granted. Valid values are <code>yes</code> and <code>no</code> . This option is not supported against vSphere 5.0 systems. The option is supported only against ESX. The option is not supported against ESXi.

---

## Managing Users with `vicfg-user`

A user is an individual authorized to log in to an ESXi or vCenter Server system.

vSphere does not explicitly restrict users with the same authentication credentials from accessing and taking action within the vSphere environment simultaneously.

You manage users defined on the vCenter Server system and users defined on individual hosts separately.

- Manage users defined on ESXi with the vSphere Client, the vSphere Web Services SDK, or `vicfg-user`.
- Manage vCenter Server users with the vSphere Client, the vSphere Web Client, or the vSphere Web Services SDK.

---

**IMPORTANT** You cannot modify users created with the vSphere Client with the `vicfg-user` command.

---

Even if the user lists of a host and a vCenter Server system appear to have common users (for instance, a user called `devuser`), these users are separate users with the same name. The attributes of `devuser` in vCenter Server, including permissions, passwords, and so forth, are separate from the attributes of `devuser` on the ESXi host. If you log in to vCenter Server as `devuser`, you might have permission to view and delete files from a datastore. If you log in to an ESXi host as `devuser`, you might not have these permissions.

Users authorized to work directly on an ESXi host are added to the internal user list when ESXi is installed or can be added by a system administrator after installation. You can use `vicfg-user` to add users, remove users, change passwords, and configure permissions.



**CAUTION** See the Authentication and User Management chapter of *vSphere Security* for information about root users before you make any changes to the default users. Mistakes regarding root users can have serious access consequences.

Each ESXi host has several default users:

- The root user has full administrative privileges. Administrators use this login and its associated password to log in to a host through the vSphere Client. Root users can control all aspects of the host that they are logged on to. Root users can manipulate permissions, creating users (on ESXi hosts only), working with events, and so on.
- The vpxuser user is a vCenter Server entity with root rights on the ESXi host, allowing it to manage activities for that host. The system creates vpxuser when an ESXi host is attached to vCenter Server. vpxuser is not present on the ESXi host unless the host is being managed through vCenter Server.
- Other users might be defined by the system, depending on the networking setup and other factors.

The following example scenario illustrates some of the tasks that you can perform. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of <conn\_options>.

### To create, modify, and delete users

- 1 List the existing users.

```
vicfg-user <conn_options> -e user -o list
```

The list displays all users that are predefined by the system and all users that were added later.

---

**IMPORTANT** The command lists a maximum of 100 users.

---

- 2 Add a new user, specifying a login ID and password.

```
vicfg-user <conn_options> -e user -o add -l user27 -p 27_password
```

The command creates the user. By default, the command autogenerates a UID for the user.

- 3 List the users again to verify that the new user was added and a UID was generated.

```
vicfg-user <conn_options> -e user -o list
USERS
```

```
-----
Principal -: root
Full Name -: root
UID -: 0
Shell Access -> 1
-----
```

```
...
```

```
-----
Principal -: user27
Full Name -:
UID -: 501
Shell Access -> 0
-----
```

- 4 Modify the password for user user27.

```
vicfg-user <conn_options> -e user -o modify -l user27 -p 27_password2
```

The system might return `Updated user user27 successfully.`

- 5 Assign read-only privileges to the user (who currently has no access).

```
vicfg-user <conn_options> -e user -o modify -l user27 --role read-only
```

The system prompts whether you want to change the password, which might be advisable if the user does not currently have a password. Answer y or n. The system then updates the user.

```
Updated user user27 successfully.
Assigned the role read-only
```

- 6 Remove the user with login ID user27.

```
vicfg-user <conn_options> -e user -o delete -l user27
```

The system removes the user and prints a message.

```
Removed the user user27 successfully.
```

## Assigning Permissions with ESXCLI

Starting with vSphere 6.0, a set of ESXCLI commands allows you to:

- Give permissions to local users and groups by assigning them one of the predefined roles.
- Give permissions to Active Directory user and groups if your ESXi host has been joined to an Active Directory domain by assigning them one of the predefined roles.

---

**IMPORTANT** When you manage local users on your ESXi host, you are not affecting the vCenter users.

---

You can list, remove, and set permissions for a user or group, as shown in the following example.

- 1 List permissions.

```
esxcli system permission list
```

The system displays permission information. The second column indicates whether the information is for a user or group.

Principal	Is Group	Role
ABCDEFGH\esx^admins	true	Admin
dcui	false	Admin
root	false	Admin
vpxuser	false	Admin
test1	false	ReadOnly

- 2 Set permissions for a user or group. Specify the ID of the user or group, and set the `--group` option to `true` to indicate a group. Specify one of three roles, `Admin`, `ReadOnly` and `NoAccess`.

```
esxcli system permission set --id test1 -r ReadOnly
```

- 3 Remove permissions for a user or group.

```
esxcli permission remove --id test1
```

You can manage accounts with the following commands:

```
esxcli system account add
esxcli system account set
esxcli system account list
esxcli system account remove
```

# Managing Virtual Machines

---

You can manage virtual machines with the vSphere Web Client or the `vmware-cmd` vCLI command. Using `vmware-cmd` you can register and unregister virtual machines, retrieve virtual machine information, manage snapshots, turn the virtual machine on and off, add and remove virtual devices, and prompt for user input.

The chapter includes these topics:

- [“vmware-cmd Overview”](#) on page 105
- [“Listing and Registering Virtual Machines”](#) on page 106
- [“Retrieving Virtual Machine Attributes”](#) on page 107
- [“Managing Virtual Machine Snapshots with vmware-cmd”](#) on page 108
- [“Powering Virtual Machines On and Off”](#) on page 109
- [“Connecting and Disconnecting Virtual Devices”](#) on page 110
- [“Working with the AnswerVM API”](#) on page 111
- [“Forcibly Stopping Virtual Machines with EXCLI”](#) on page 111

Some virtual machine management utility applications are included in the vSphere SDK for Perl.

The vSphere PowerCLI cmdlets, which you can install for use with Microsoft PowerShell, manage many aspects of virtual machines.

## vmware-cmd Overview

`vmware-cmd` was included in earlier version of the ESX Service Console. A `vmware-cmd` command has been available in the vCLI package since ESXi version 3.0.

---

**IMPORTANT** `vmware-cmd` is not available in the ESXi Shell. Run the `vmware-cmd` vCLI command instead.

---

Older versions of `vmware-cmd` support a set of connection options and general options that differ from the options in other vCLI commands. The `vmware-cmd` vCLI command supports these options. The vCLI command also supports the standard vCLI `--server`, `--username`, `--password`, and `--vhost` options. `vmware-cmd` does not support other connection options.

---

**IMPORTANT** `vmware-cmd` is a legacy tool and supports the usage of VMFS paths for virtual machine configuration files. As a rule, use datastore paths to access virtual machine configuration files.

---

## Connection Options for vmware-cmd

The `vmware-cmd` vCLI command supports only the following connection options. Other vCLI connection options are not supported, for example, you cannot use variables because the corresponding option is not supported.

Option	Description
<code>--server &lt;host&gt;</code> <code>-H &lt;host&gt;</code>	Target ESXi or vCenter Server system.
<code>--vihost &lt;target&gt;</code> <code>-h &lt;target&gt;</code>	When you run <code>vmware-cmd</code> with the <code>-H</code> option pointing to a vCenter Server system, use <code>--vihost</code> to specify the ESXi host to run the command against.
<code>-O &lt;port&gt;</code>	Alternative connection port. The default port number is 902.
<code>--username &lt;username&gt;</code> <code>-U &lt;username&gt;</code>	User who is authorized to log in to the host specified by <code>--server</code> or <code>--vihost</code> .
<code>--password &lt;password&gt;</code> <code>-P &lt;password&gt;</code>	Password for the user specified by <code>-U</code> .
<code>-Q &lt;protocol&gt;</code>	Protocol to use, either <code>http</code> or <code>https</code> . Default is <code>https</code> .

## General Options for vmware-cmd

The `vmware-cmd` vCLI command supports the following general options.

Option	Description
<code>--help</code>	Prints a help message that lists the options for this command.
<code>-q</code>	Runs in quiet mode with minimal output. The output does not display the specified operation and arguments.
<code>-v</code>	Runs in verbose mode.

## Format for Specifying Virtual Machines

When you run `vmware-cmd`, the virtual machine path is usually required. You can specify the virtual machine using one of the following formats:

- Datastore prefix style: `'[ds_name] relative_path'`, for example:
  - `'[myStorage1] testvms/VM1/VM1.vmx'` (Linux)
  - `"[myStorage1] testvms/VM1/VM1.vmx"` (Windows)
- UUID-based path: `folder/subfolder/file`, for example:
  - `'/vmfs/volumes/mystorage/testvms/VM1/VM1.vmx'` (Linux)
  - `"/vmfs/volumes/mystorage/testvms/VM1/VM1.vmx"` (Windows)

## Listing and Registering Virtual Machines

Registering or unregistering a virtual machine means adding the virtual machine to the vCenter Server or ESXi inventory or removing the virtual machine.

---

**IMPORTANT** If you register a virtual machine with a vCenter Server system, and then remove it from the ESXi host, an orphaned virtual machine results. Call `vmware-cmd -s unregister` with the vCenter Server system as the target to resolve the issue.

---

The following example scenario lists all registered virtual machines on a vCenter Server, unregisters a virtual machine, and reregisters the virtual machine.

### To list, unregister, and register virtual machines

- 1 Run `vmware-cmd -l` to list all registered virtual machines on a server.

```
vmware-cmd -H <vc_server> -U <login_user> -P <login_password> --vihost <esx_host> -l
```

The command lists the VMX file for each virtual machine.

```
/vmfs/volumes/<storage>/winxpPro-sp2/winxpPro-sp2.vmx
/vmfs/volumes/<storage>/RHEL-lsi/RHEL-lsi.vmx
/vmfs/volumes/<storage>/VIMA0809/VIMA0809.vmx
.....
```

- 2 Run `vmware-cmd -s unregister` to remove a virtual machine from the inventory.

```
vmware-cmd -H <vc_server> -U <login_user> -P <login_password> --vihost <esx_host>
-s unregister /vmfs/volumes/Storage2/testvm/testvm.vmx
```

The system returns 0 to indicate success, 1 to indicate failure.

---

**NOTE** When you run against a vCenter Server system, you must specify the data center and the resource pool to register the virtual machine in. The default data center is `ha-datacenter` and the default resource pool is `Resources`.

When you run against an ESXi host, you usually do not specify the resource pool and data center. However, if two virtual machines with the same name exist in two resource pools, you must specify the resource pool.

---

- 3 Run `vmware-cmd -l` again to verify that the virtual machine was removed from the inventory.
- 4 Run `vmware-cmd -s register` to add the virtual machine back to the inventory.

```
vmware-cmd -H <vc_server> -U <login_user> -P <login_password> --vihost <esx_host> -s register
/vmfs/volumes/Storage2/testvm/testvm.vmx
```

The system returns 0 to indicate success, 1 to indicate failure.

## Retrieving Virtual Machine Attributes

`vmware-cmd` includes options for retrieving information about a virtual machine. Each option requires that you specify the virtual machine path (see [“Format for Specifying Virtual Machines”](#) on page 106). You must also specify connection options, which differ from other vCLI commands (see [“Connection Options for vmware-cmd”](#) on page 106).

You can use `vmware-cmd` options to retrieve a number of different virtual machine attributes. For a complete list of options, see the *vSphere CLI Reference*.

- The `guestinfo` option allows you to retrieve information about the guest operating system. For example, you can retrieve the number of remote consoles allowed by a virtual machine by using `guestinfo` with the `RemoteDisplay.maxConnections` variable.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
/vmfs/volumes/Storage2/testvm/testvm.vmx getguestinfo RemoteDisplay.maxConnections
```

The *Hardening Guide* includes additional information about variables you can use in conjunction with `guestinfo`. A complete list is not available.

- The `getuptime` option retrieves the uptime of the guest operating system on the virtual machine, in seconds.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
/vmfs/volumes/Storage2/testvm/testvm.vmx getuptime
```

```
getuptime() = 17921
```

- The `getproductinfo product` option lists the VMware product that the virtual machine runs on.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
/vmfs/volumes/Storage2/testvm/testvm.vmx getproductinfo product
```

The return value is `esx` (VMware ESX), `embeddedESX` (VMware ESXi), or `unknown`.

- The `getproductinfo platform` option lists the platform that the virtual machine runs on.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
/vmfs/volumes/Storage2/testvm/testvm.vmx getproductinfo platform
```

The return value is `win32-x86` (x86-based Windows system), `linux-x86` (x86-based Linux system), or `vmnix-x86` (x86-based ESXi microkernel).

- The `getproductinfo build`, `getproductinfo majorversion`, or `getproductinfo minorversion` options retrieve version information.

- The `getstate` option retrieves the execution state of the virtual machine, which can be `on`, `off`, `suspended`, or `unknown`.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
/vmfs/volumes/Storage2/testvm/testvm.vmx getstate
getstate() = on
```

- The `gettoolslastactive` option indicates whether VMware Tools is installed and whether the guest operating system is responding normally.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
/vmfs/volumes/Storage2/testvm/testvm.vmx gettoolslastactive
```

The command returns an integer indicating how much time has passed, in seconds, since the last heartbeat was detected from the VMware Tools service. This value is initialized to zero when a virtual machine powers on. The value stays at zero until the first heartbeat is detected. After the first heartbeat, the value is always greater than zero until the virtual machine is power cycled again. The command returns one of the following values:

- 0 – VMware Tools is not installed or not running.
- 1 – Guest operating system is responding normally.
- 5 – Intermittent heartbeat. There might be a problem with the guest operating system.
- 100 – No heartbeat. Guest operating system might have stopped responding.

---

**NOTE** You usually use the `vmware-cmd guestinfo` option only when VMware Support instructs you to do so. The command is therefore not discussed in this document.

---

## Managing Virtual Machine Snapshots with `vmware-cmd`

A snapshot captures the entire state of the virtual machine at the time you take the snapshot.

Virtual machine state includes the following aspects of the virtual machine.

- **Memory state.** Contents of the virtual machine's memory.
- **Settings state.** Virtual machine settings.
- **Disk state.** State of all the virtual machine's virtual disks.

When you revert to a snapshot, you return these items to the state they were in at the time that you took the snapshot. If you want the virtual machine to be running or to be shut down when you start it, make sure that it is in that state when you take the snapshot.

You can use snapshots as restoration points when you install update packages, or during a branching process, such as installing different versions of a program. Taking snapshots ensures that each installation begins from an identical baseline. The *vSphere Virtual Machine Administration* documentation discusses snapshots in detail.

---

**IMPORTANT** Use the vSphere Web Client to revert to a named snapshot. `vmware-cmd` only supports reverting to the current snapshot.

---



## Taking Virtual Machine Snapshots

You can take a snapshot while a virtual machine is running, shut down, or suspended. If you are in the process of suspending a virtual machine, wait until the suspend operation has finished before taking a snapshot.

If a virtual machine has multiple disks in different disk modes, you must shut down the virtual machine before taking a snapshot. For example, if you have a special-purpose configuration that requires you to use an independent disk, you must shut down the virtual machine before taking a snapshot.

### To take a snapshot

- 1 (Optional) If the virtual machine has multiple disks in different disk modes, shut down the virtual machine.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
/vmfs/volumes/Storage2/testvm/testvm.vmx stop soft
```

- 2 (Optional) Check that the shut down operation has been completed.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
/vmfs/volumes/Storage2/testvm/testvm.vmx getstate
```

- 3 Run `vmware-cmd` with the `createsnapshot` option.

You must specify the description, quiesce flag (0 or 1) and memory flag (0 or 1).

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
/vmfs/volumes/Storage2/testvm/testvm.vmx createsnapshot VM1Aug09
'test snapshot August 09' 0 0
```

- 4 Check that the virtual machine has a snapshot by using the `hassnapshot` option.

The call returns 1 if the virtual machine has a snapshot and returns 0 otherwise.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
/vmfs/volumes/Storage2/testvm/testvm.vmx hassnapshot
```

```
hassnapshot () = 1
```

## Reverting and Removing Snapshots

You can use `vmware-cmd` to revert to the current snapshot or to remove a snapshot.

---

**IMPORTANT** You cannot use `vmware-cmd` to revert to a named snapshot. Use the vSphere Web Client to revert to a named snapshot.

---

Run `vmware-cmd` with the `revertsnapshot` option to revert to the current snapshot. If no snapshot exists, the command does nothing and leaves the virtual machine state unchanged.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
/vmfs/volumes/Storage2/testvm/testvm.vmx revertsnapshot
```

Run `vmware-cmd` with the `removesnapshots` option to remove all snapshots associated with a virtual machine. If no snapshot exists, the command does nothing.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
/vmfs/volumes/Storage2/testvm/testvm.vmx removesnapshots
```

## Powering Virtual Machines On and Off

You can start, reboot, stop, and suspend virtual machines by using `vmware-cmd`. You must supply a value for the `powerop_mode` flag, which can be `soft` or `hard`.

---

**IMPORTANT** You must have the current version of VMware Tools installed and running in the guest operating system to use a `soft` power operation.

---

- **Soft power operations.** When you specify `soft` as the `powerop_mode` value, the result of the call depends on the operation.

Operation	Result
Stop	<code>vmware-cmd</code> attempts to shut down the guest operating system and powers off the virtual machine.
Reset	<code>vmware-cmd</code> attempts to shut down the guest operating system and reboots the virtual machine.
Suspend	<code>vmware-cmd</code> attempts to run a script in the guest operating system before suspending the virtual machine.

- **Hard power operations.** `vmware-cmd` immediately and unconditionally shuts down, resets, or suspends the virtual machine.

The following examples illustrate how to use `vmware-cmd`.

- **Start.** Use the `start` option to power on a virtual machine or to resume a suspended virtual machine. The `powerop_mode`, either `hard` or `soft`, is required.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
/vmfs/volumes/Storage2/testvm/testvm.vmx start soft
```

- **Reset.** When you reset the virtual machine with the `soft` `power_op` mode (the default), the guest operating system is shut down before the reset.

If VMware Tools is not currently installed on the virtual machine, you can perform only a hard reset operation.

- Check that VMware tools is installed so that you can reset the virtual machine with the default `power_op` mode, which is `soft`.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
/vmfs/volumes/Storage2/testvm/testvm.vmx gettoolslastactive
```

See [“Retrieving Virtual Machine Attributes”](#) on page 107.

- Use the `reset` option to shut down and restart the virtual machine.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
/vmfs/volumes/Storage2/testvm/testvm.vmx reset soft
```

- **Suspend.** You have two options for suspending a virtual machine.

- The `suspend` option with the `hard` `powerop` mode unconditionally shuts down a virtual machine.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
/vmfs/volumes/Storage2/testvm/testvm.vmx suspend hard
```

- The `suspend` option with the `soft` `powerop` mode runs scripts that result in a graceful shut-down of the guest operating system and shuts down the virtual machine. VMware Tools must be installed for `soft` `powerop_mode`.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
/vmfs/volumes/Storage2/testvm/testvm.vmx suspend soft
```

## Connecting and Disconnecting Virtual Devices

The `vmware-command` options `connectdevice` and `disconnectdevice` allow you to connect the following types of devices to your virtual machines and to disconnect them:

- Network adapters
- CD/DVD drives
- Floppy drives

These devices must already be defined in the virtual machine virtual hardware list.

The command options connect and disconnect a defined ISO or FLP file or a physical device on the host. After you connect a device, its content can be accessed from the Guest OS. For network adapters, the options connect the virtual NIC to its defined port group or disconnect the NIC. This is equivalent to selecting or deselecting the **Connected** check box in the vSphere Web Client.

---

**NOTE** The terms **CD/DVD drive**, **Floppy drive**, and **Network adapter** are case-sensitive.

---

You can connect or disconnect devices if the following conditions are met:

- The virtual machine has a guest operating system that supports hot-plug functionality. See the *Operating System Installation* documentation.
- The virtual machine is using hardware version 7 or later.
- The virtual machine is powered on

The following examples illustrate connecting and disconnecting a virtual device. Device names are case sensitive.

- The `connectdevice` option connects the virtual IDE device CD/DVD Drive 2 to the specified virtual machine.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
/vmfs/volumes/Storage2/testvm/testvm.vmx connectdevice "CD/DVD drive 2"
```

- The `disconnectdevice` option disconnects the virtual device.

```
vmware-cmd -H <vc_system> -U <user> -P <password> --vihost <esx_host>
/vmfs/volumes/Storage2/testvm/testvm.vmx disconnectdevice "CD/DVD drive 2"
```

## Working with the AnswerVM API

The AnswerVM API allows users to provide input to questions, thereby allowing blocked virtual machine operations to complete. The `vmware-cmd --answer` option allows you to access the input. You might use this option when you want to configure a virtual machine based on a user's input. For example:

- 1 The user clones a virtual machine and provides the default virtual disk type.
- 2 When the user powers on the virtual machine, it prompts for the desired virtual disk type.

## Forcibly Stopping Virtual Machines with EXCLI

In some cases, virtual machines do not respond to the normal shutdown or stop commands. In these cases, it might be necessary to forcibly shut down the virtual machines. Forcibly shutting down a virtual machine might result in guest operating system data loss and is similar to pulling the power cable on a physical machine.

You can forcibly stop virtual machines that are not responding to normal stop operation with the `esxcli vm process kill` command. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

### To forcibly stop a virtual machine

- 1 List all running virtual machines on the system to see the World ID of the virtual machine that you want to stop.

```
esxcli <conn_options> vm process list
```

- 2 Stop the virtual machine by running the following command.

```
esxcli <conn_options> vm process kill --type <kill_type> --world-id <ID>
```

The command supports three `--type` options. Try the types sequentially (soft before hard, hard before force). The following types are supported through the `--type` option:

- `soft`. Gives the VMX process a chance to shut down cleanly (like `kill` or `kill -SIGTERM`)

- `hard`. Stops the VMX process immediately (like `kill -9` or `kill -SIGKILL`)
- `force`. Stops the VMX process when other options do not work.

If all three options do not work, reboot your ESXi host to resolve the issue.

# Managing vSphere Networking

---

The vSphere CLI networking commands allow you to manage the vSphere network services. You can connect virtual machines to the physical network and to each other and configure vSphere standard switches. Limited configuration of vSphere distributed switches is also supported. You can also set up your vSphere environment to work with external networks such as SNMP or NTP.

This chapter includes the following topics:

- [“Introduction to vSphere Networking”](#) on page 113
- [“Retrieving Basic Networking Information”](#) on page 115
- [“Network Troubleshooting”](#) on page 116
- [“Setting Up vSphere Networking with vSphere Standard Switches”](#) on page 117
- [“Setting Up vSphere Networking with vSphere Distributed Switch”](#) on page 128
- [“Managing Standard Networking Services in the vSphere Environment”](#) on page 128
- [“Setting the DNS Configuration”](#) on page 128
- [“Adding and Starting an NTP Server”](#) on page 131
- [“Managing the IP Gateway”](#) on page 131
- [“Setting Up IPsec”](#) on page 132
- [“Managing the ESXi Firewall”](#) on page 135
- [“Monitoring VXLAN”](#) on page 136

## Introduction to vSphere Networking

At the core of vSphere Networking are virtual switches. vSphere supports standard switches (VSS) and distributed switches (VDS). Each virtual switch has a preset number of ports and one or more port groups.

Virtual switches allow your virtual machines to connect to each other and to connect to the outside world.

- When two or more virtual machines are connected to the same virtual switch, and those virtual machines are also on the same portgroup or VLAN, network traffic between them is routed locally.
- When virtual machines are connected to a virtual switch that is connected to an uplink adapter, each virtual machine can access the external network through that uplink. The adapter can be an uplink connected to a standard switch or a distributed uplink port connected to a distributed switch.

Virtual switches allow your ESXi host to migrate virtual machines with VMware vMotion and to use IP storage through VMkernel network interfaces.

- Using vMotion, you can migrate running virtual machines with no downtime. You can enable vMotion with `vicfg-vmknic --enable-vmotion`. You cannot enable vMotion with ESXCLI.

- IP storage refers to any form of storage that uses TCP/IP network communication as its foundation and includes iSCSI and NFS for ESXi. Because these storage types are network based, they can use the same VMkernel interface and port group.

The network services that the VMkernel provides (iSCSI, NFS, and vMotion) use a TCP/IP stack in the VMkernel. The VMkernel TCP/IP stack is also separate from the guest operating system's network stack. Each of these stacks accesses various networks by attaching to one or more port groups on one or more virtual switches.

## Networking Using vSphere Standard Switches

vSphere standard switches allow you to connect virtual machines to the outside world.

**Figure 9-1.** Networking with vSphere Standard Switches

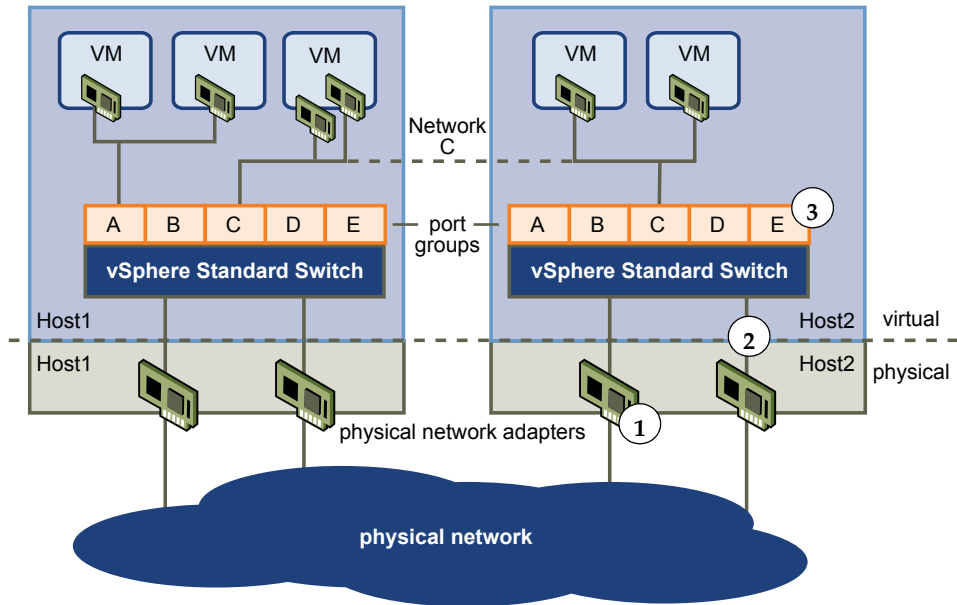


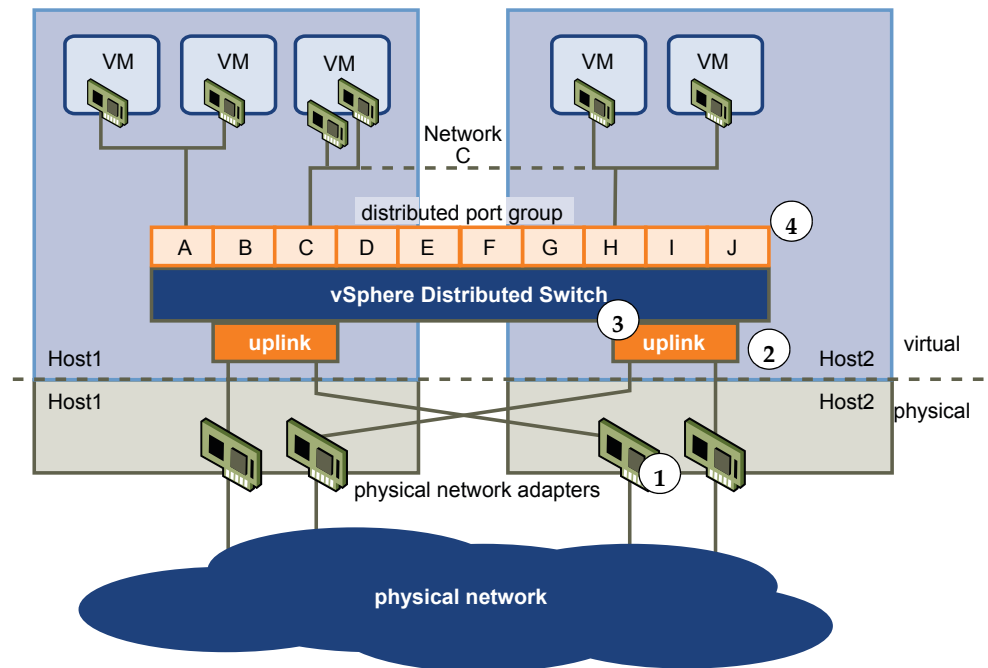
Figure 9-1 shows the relationship between the physical and virtual network elements. The numbers match those in the figure.

- Associated with each ESXi host are one or more uplink adapters (1). Uplink adapters represent the physical switches the ESXi host uses to connect to the network. You can manage uplink adapters using the `esxcli network nic` or `vicfg-nics` vCLI command. See [“Managing Uplink Adapters”](#) on page 122.
- Each uplink adapter is connected to a standard switch (2). You can manage a standard switch and associate it with uplink adapters by using the `esxcli network vswitch` or `vicfg-vswitch` vCLI command. See [“Setting Up Virtual Switches and Associating a Switch with a Network Interface”](#) on page 117.
- Associated with the standard switch are port groups (3). Port group is a unique concept in the virtual environment. You can configure port groups to enforce policies that provide enhanced networking security, network segmentation, better performance, high availability, and traffic management. You can use the `esxcli network vswitch standard portgroup` or `vicfg-vswitch` command to associate a standard switch with a port group, and the `esxcli network ip interface` or `vicfg-vmknic` command to associate a port group with a VMkernel network interface.
- The VMkernel TCP/IP networking stack supports iSCSI, NFS, and vMotion and has an associated VMkernel network interface. You configure VMkernel network interfaces with `esxcli network ip interface` or `vicfg-vmknic`. See [“Adding and Modifying VMkernel Network Interfaces”](#) on page 125. Separate VMkernel network interfaces are often used for separate tasks, for example, you might devote one VMkernel Network interface card to vMotion only. Virtual machines run their own systems' TCP/IP stacks and connect to the VMkernel at the Ethernet level through virtual switches.

## Networking Using vSphere Distributed Switches

When you want to connect a virtual machine to the outside world, you can use a standard switch or a distributed switch. With a distributed switch, the virtual machine can maintain its network settings even if the virtual machine is migrated to a different host.

**Figure 9-2.** Networking with vSphere Distributed Switches



- Each physical network adapter (1) on the host is paired with a distributed uplink port (2), which represents the uplink to the virtual machine. With distributed switches, the virtual machine no longer depends on the host's physical uplink but on the (virtual) uplink port. You manage a uplink ports primarily using the vSphere Web Client, or vSphere APIs.
- The distributed switch itself (3) functions as a single virtual switch across all associated hosts. Because the switch is not associated with a single host, virtual machines can maintain consistent network configuration as they migrate from one host to another.

Like a standard switch, each distributed switch is a network hub that virtual machines can use. A distributed switch can route traffic internally between virtual machines or link to an external network by connecting to physical network adapters. You create a distributed switch using the the vSphere Web Client UI, but can manage some aspects of a distributed switch with `vicfg-vswitch`. You can list distributed virtual switches with the `esxcli network vswitch` command. See [“Setting Up Virtual Switches and Associating a Switch with a Network Interface”](#) on page 117.

## Retrieving Basic Networking Information

Service console commands for retrieving networking information are not included in the ESXi Shell. You can instead use ESXCLI commands directly in the shell or use vCLI commands.

On ESXi 5.0, `ifconfig` information should be the information of the VMkernel NIC that attaches to the Management Network port group. You can retrieve information by using ESXCLI commands.

```
esxcli <conn_options> network ip interface list
esxcli <conn_options> network ip interface ipv4 get -n vmk<X>
esxcli <conn_options> network ip interface ipv6 get -n vmk<X>
esxcli <conn_options> network ip interface ipv6 address list
```

For information corresponding to the Linux `netstat` command, use the following ESXCLI command.

```
esxcli <conn_options> network ip connection list
```

You can also ping individual hosts with the `esxcli network diag ping` command. The command includes options for using ICMPv4 or ICMPv6 packet requests, specifying an interface to use, specifying the interval, and so on.

## Network Troubleshooting

You can use vCLI network commands to view network statistics and troubleshoot your networking setup. The nested hierarchy of commands allows you to drill down to potential trouble spots.

- 1 List all virtual machine networks on a host.

```
esxcli network vm list
```

The command returns for each virtual machine the World ID, name, number of ports, and networks, as in the following example.

World ID	Name	Num Ports	Networks
10374	ubuntu-server-11.04-1	2	VM Network, dvportgroup-19
10375	ubuntu-server-11.04-2	2	VM Network, dvportgroup-19
10376	ubuntu-server-11.04-3	2	VM Network, dvportgroup-19
10408	ubuntu-server-11.04-4	3	VM Network, VM Network 10Gbps, dvportgroup-19

- 2 List the ports for one of the VMs by specifying its World ID.

```
esxcli network vm port list -w 10408
```

The command returns port information, as in the following example.

```
Port:
  Port ID: XXXXXXXX
  vSwitch: vSwitch0
  Portgroup: VM Network
  DVPort ID:
  MAC Address: 00:XX:XX:aa:XX:XX
  IP Address: 10.XXX.XXX.XXX
  Team Uplink: vmnic0
  Uplink Port ID: 12345678
  Active Filters:
```

- 3 Retrieve the switch statistics for a port.

```
esxcli network port stats get -p 12345678
```

The command returns detailed statistics, as in the following example.

```
Packet statistics for port 12345678:
  Packets received: 517631
  Packets sent: 18937
  Bytes received: 100471874
  Bytes sent: 1527233
  Broadcast packets received: 474160
  Broadcast packets sent: 107
  Multicast packets received: 8020
  Multicast packets sent: 8
  Unicast packets received: 35451
  Unicast packets sent: 18822
  Receive packets dropped: 45
  Transmit packets dropped: 0
```

- 4 Retrieve the filter information for the port.

```
esxcli network port filter stats get -p 12345678
```

The command returns detailed statistics, as in the following example.

```
Filter statistics for dfilter-test:
  Filter direction: Receive
  Packets in: 202080
  Packets out: 202080
  Packets dropped: 0
```



```

Packets filtered: 0
Packets faulted: 0
Packets queued: 0
Packets injected: 0
Packet errors: 0

```

- 5 Retrieve complete statistics for a NIC.

```
esxcli network nic stats get -n vmnic0
```

- 6 Get a per-VLAN packed breakdown on a NIC.

```
esxcli network nic vlan stats get -n vmnic0
```

The command returns the number of packets sent and received for the VLAN you specified.

## Setting Up vSphere Networking with vSphere Standard Switches

You can set up your virtual network by performing these tasks.

- 7 Create or manipulate virtual switches using `esxcli network vswitch` or `vicfg-vswitch`. By default, each ESXi host has one virtual switch, `vSwitch0`. You can create additional virtual switches or manage existing switches. See [“Setting Up Virtual Switches and Associating a Switch with a Network Interface”](#) on page 117.
- 8 (Optional) Make changes to the uplink adapter using `esxcli network vswitch standard uplink` or `vicfg-nics`. See [“Managing Uplink Adapters”](#) on page 122.
- 9 (Optional) Use `esxcli network vswitch standard portgroup` or `vicfg-vswitch` to add port groups to the virtual switch. See [“Managing Port Groups with vicfg-vswitch”](#) on page 120.
- 10 (Optional) Use `esxcli network vswitch standard portgroup set` or `vicfg-vswitch` to establish VLANs by associating port groups with VLAN IDs. See [“Setting the Port Group VLAN ID with vicfg-vswitch”](#) on page 122.
- 11 Use `esxcli network ip interface` or `vicfg-vmknics` to configure the VMkernel network interfaces. See [“Adding and Modifying VMkernel Network Interfaces”](#) on page 125.

### Setting Up Virtual Switches and Associating a Switch with a Network Interface

A virtual switch models a physical Ethernet switch. You can manage virtual switches and port groups by using the vSphere Web Client (see the *vSphere Networking* documentation) or by using vSphere CLI commands.

You can create a maximum of 127 virtual switches on a single ESXi host. By default, each ESXi host has a single virtual switch called `vSwitch0`. By default, a virtual switch has 56 logical ports. See the *Configuration Maximums* document on the vSphere documentation main page for details. Ports connect to the virtual machines and the ESXi physical network adapters.

- You can connect one virtual machine network adapter to each port by using the vSphere Web Client UI.
- You can connect the uplink adapter to the virtual switches by using `vicfg-vswitch` or `esxcli network vswitch standard uplink`. See [“Linking and Unlinking Uplink Adapters with vicfg-vswitch”](#) on page 124.

When two or more virtual machines are connected to the same virtual switch, network traffic between them is routed locally. If an uplink adapter is attached to the virtual switch, each virtual machine can access the external network that the adapter is connected to.

This section discusses working in a standard switch environment. See [“Networking Using vSphere Distributed Switches”](#) on page 115 for information about distributed switch environments.

When working with virtual switches and port groups, perform the following tasks:

- 1 Find out which virtual switches are available and (optionally) what the associated MTU and CDP (Cisco Discovery Protocol) settings are. See [“Retrieving Information about Virtual Switches with ESXCLI”](#) on page 118 and [“Retrieving Information about Virtual Switches with vicfg-vswitch”](#) on page 118.
- 2 Add a virtual switch. See [“Adding and Deleting Virtual Switches with ESXCLI”](#) on page 119 and [“Adding and Deleting Virtual Switches with vicfg-vswitch”](#) on page 119.
- 3 For a newly added switch, perform these tasks:
  - a Add a port group. See [“Managing Port Groups with ESXCLI”](#) on page 120 and [“Managing Port Groups with vicfg-vswitch”](#) on page 120.
  - b (Optional) Set the port group VLAN ID. See [“Setting the Port Group VLAN ID with ESXCLI”](#) on page 121 and [“Setting the Port Group VLAN ID with vicfg-vswitch”](#) on page 122.
  - c Add an uplink adapter. See [“Linking and Unlinking Uplink Adapters with ESXCLI”](#) on page 124 and [“Linking and Unlinking Uplink Adapters with vicfg-vswitch”](#) on page 124.
  - d (Optional) Change the MTU or CDP settings. See [“Setting Switch Attributes with esxcli network vswitch standard”](#) on page 119 and [“Setting Switch Attributes with vicfg-vswitch”](#) on page 120.

## Retrieving Information About Virtual Switches

You can retrieve information about virtual switches by using ESXCLI or vicfg-vswitch. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of <conn\_options>.

### Retrieving Information about Virtual Switches with ESXCLI

You can retrieve information about virtual switches by using `esxcli network vswitch` commands.

- List all virtual switches and associated port groups.

```
esxcli <conn_options> network vswitch standard list
```

The command prints information about the virtual switch, which might include its name, number of ports, MTU, port groups, and other information. The output includes information about CDP settings for the virtual switch. The precise information depends on the target system. The default port groups are Management Network and VM Network.

- List the network policy settings (security policy, traffic shaping policy, and failover policy) for the virtual switch. The following commands are supported.

```
esxcli <conn_options> network vswitch standard policy failover get
esxcli <conn_options> network vswitch standard policy security get
esxcli <conn_options> network vswitch standard policy shaping get
```

### Retrieving Information about Virtual Switches with vicfg-vswitch

You can retrieve information about virtual switches by using the `vicfg-vswitch` command. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of <conn\_options>.

- Check whether vSwitch1 exists.

```
vicfg-vswitch <conn_options> -c vSwitch1
```

- List all virtual switches and associated port groups.

```
vicfg-vswitch <conn_options> -l
```

The command prints information about the virtual switch, which might include its name, number of ports, MTU, port groups, and other information. The default port groups are Management Network and VM Network.

- Retrieve the current CDP (Cisco Discovery Protocol) setting for this virtual switch.

If CDP is enabled on a virtual switch, ESXi administrators can find out which Cisco switch port is connected to which virtual switch uplink. CDP is a link-level protocol that supports discovery of CDP-aware network hardware at either end of a direct connection. CDP is bit forwarded through switches. CDP is a simple advertisement protocol which beacons information about the switch or host and some port information.

```
vicfg-vswitch <conn_options> --get-cdp vSwitch1
```

## Adding and Deleting Virtual Switches

You can add and delete virtual switches with ESXCLI and with `vicfg-vswitch`.

### Adding and Deleting Virtual Switches with ESXCLI

You can add and delete virtual switches using the `esxcli network vswitch standard` namespace. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- Add a virtual switch.

```
esxcli <conn_options> network vswitch standard add --vswitch-name=vSwitch42
```

You can specify the number of port groups while adding the virtual switch. If you do not specify a value, the default value is used. The system-wide port count cannot be greater than 4096.

```
esxcli <conn_options> network vswitch standard add --vswitch-name=vSwitch42 --ports=8
```

After you have added a virtual switch, you can set switch attributes ([“Setting Switch Attributes with esxcli network vswitch standard”](#) on page 119) and add one or more uplink adapters ([“Linking and Unlinking Uplink Adapters with ESXCLI”](#) on page 124).

- Delete a virtual switch.

```
esxcli <conn_options> network vswitch standard remove --vswitch-name=vSwitch42
```

You cannot delete a virtual switch if any ports on the switch are still in use by VMkernel networks or virtual machines. Run `esxcli network vswitch standard list` to determine whether a virtual switch is in use.

### Adding and Deleting Virtual Switches with `vicfg-vswitch`

You can add and delete virtual switches using the `--add|-a` and `--delete|-d` options. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- Add a virtual switch.

```
vicfg-vswitch <conn_options> --add vSwitch2
```

After you have added a virtual switch, you can set switch attributes ([“Setting Switch Attributes with vicfg-vswitch”](#) on page 120) and add one or more uplink adapters ([“Linking and Unlinking Uplink Adapters with vicfg-vswitch”](#) on page 124).

- Delete a virtual switch.

```
vicfg-vswitch <conn_options> --delete vSwitch1
```

You cannot delete a virtual switch if any ports on the switch are still in use by VMkernel networks, virtual machines, or `vswifs`. Run `vicfg-vswitch --list` to determine whether a virtual switch is in use.

### Setting Switch Attributes with `esxcli network vswitch standard`

You can set the maximum transmission unit (MTU) and CDP status for a virtual switch. The CDP status shows which Cisco switch port is connected to which uplink. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- Set the MTU for a vSwitch.

```
esxcli <conn_options> network vswitch standard set --mtu=9000 --vswitch-name=vSwitch1
```

The MTU is the size, in bytes, of the largest protocol data unit the switch can process. When you set this option, it affects all uplinks assigned to the virtual switch.

- Set the CDP value for a vSwitch. You can set status to down, listen, advertise, or both.

```
esxcli <conn_options> network vswitch standard set --cdp-status=listen
--vswitch-name=vSwitch1
```

### Setting Switch Attributes with vicfg-vswitch

You can set the maximum transmission unit (MTU) and CDP status for a virtual switch. The CDP status shows which Cisco switch port is connected to which uplink. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of <conn\_options>.

- Set the MTU for a vSwitch.

```
vicfg-vswitch <conn_options> -m 9000 vSwitch1
```

The MTU is the size (in bytes) of the largest protocol data unit the switch can process. When you set this option, it affects all uplinks assigned to the virtual switch.

- Set the CDP value for a vSwitch. You can set status to down, listen, advertise, or both.

```
vicfg-vswitch <conn_options> --set-cdp 'listen'
```

## Checking, Adding, and Removing Port Groups

You can check, add, and remove port groups with ESXCLI and with vicfg-vswitch.

### Managing Port Groups with ESXCLI

Network services connect to vSwitches through port groups. A port group allows you to group traffic and specify configuration options such as bandwidth limitations and VLAN tagging policies for each port in the port group. A virtual switch must have one port group assigned to it. You can assign additional port groups.

You can use `esxcli network vswitch standard portgroup` to check, add, and remove port groups. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of <conn\_options>.

- List port groups currently associated with a virtual switch.

```
esxcli <conn_options> network vswitch standard portgroup list
```

Lists the port group name, associated virtual switch, active clients, and VLAN ID.

- Add a port group.

```
esxcli <conn_options> network vswitch standard portgroup add --portgroup-name=<name>
--vswitch-name=vSwitch1
```

- Delete one of the existing port groups.

```
esxcli <conn_options> network vswitch standard portgroup remove --portgroup-name=<name>
--vswitch-name=vSwitch1
```

### Managing Port Groups with vicfg-vswitch

Network services connect to virtual switches through port groups. A port group allows you to group traffic and specify configuration options such as bandwidth limitations and VLAN tagging policies for each port in the port group. A virtual switch must have one port group assigned to it. You can assign additional port groups. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of <conn\_options>.

You can use `vicfg-vswitch` to check, add, and remove port groups.

- Check whether port groups are currently associated with a virtual switch.

```
vicfg-vswitch <conn_options> --check-pg <port_group> vSwitch1
```

The command returns 0 if the specified port group is associated with the virtual switch, and returns 1 otherwise. Use `vicfg-vswitch --list` to list all port groups.

- Add a port group.  
`vicfg-vswitch <conn_options> --add-pg <port_group_name> vSwitch1`
- Delete one of the existing port groups.  
`vicfg-vswitch <conn_options> --del-pg <port_group_name> vSwitch1`

## Managing Uplinks and Port Groups

You can manage uplinks and port groups with ESXCLI and with `vicfg-vswitch`.

### Connecting and Disconnecting Uplink Adapters and Port Groups with ESXCLI

If your setup includes one or more port groups, you can associate each port group with one or more uplink adapters (and remove the association). This functionality allows you to filter traffic from a port group to a specific uplink, even if the virtual switch is connected with multiple uplinks. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- Connect a port group with an uplink adapter.  
`esxcli <conn_options> network vswitch standard portgroup policy failover set --active-uplinks=vmnic1,vmnic6,vmnic7`

This command fails silently if the uplink adapter does not exist.

- Make some of the adapters standby instead of active.  
`esxcli <conn_options> network vswitch standard portgroup policy failover set --standby-uplinks=vmnic1,vmnic6,vmnic7`

### Connecting and Disconnecting Uplinks and Port Groups with `vicfg-vswitch`

If your setup includes one or more port groups, you can associate each port group with one or more uplink adapters (and remove the association). This functionality allows you to filter traffic from a port group to a specific uplink, even if the virtual switch is connected with multiple uplinks. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- Connect a port group with an uplink adapter.  
`vicfg-vswitch <conn_options> --add-pg-uplink <adapter_name> --pg <port_group> <vswitch_name>`  
This command fails silently if the uplink adapter does not exist.
- Remove a port group from an uplink adapter.  
`vicfg-vswitch <conn_options> --del-pg-uplink <adapter_name> --pg <port_group> <vswitch_name>`

## Setting the Port Group VLAN ID

You can set the port group VLAN ID with ESXCLI and with `vicfg-vswitch`.

### Setting the Port Group VLAN ID with ESXCLI

VLANs allow you to further segment a single physical LAN segment so that groups of ports are isolated as if they were on physically different segments. The standard is IEEE 802.1Q.

A VLAN ID restricts port group traffic to a logical Ethernet segment within the physical network.

- Set the VLAN ID to 4095 to allow a port group to reach port groups located on other VLAN.
- Set the VLAN ID to 0 to disable the VLAN for this port group.

If you use VLAN IDs, you must change the port group labels and VLAN IDs together so that the labels properly represent connectivity. VLAN IDs are optional.

You can use the following commands for VLAN management:

- Allow port groups to reach port groups located on other VLANs.  

```
esxcli <conn_options> network vswitch standard portgroup set -p <pg_name> --vlan-id 4095
```

Call the command multiple times to allow all ports to reach port groups located on other VLANs.

- Disable VLAN for port group g42  

```
esxcli <conn_options> network vswitch standard portgroup set --vlan-id 0 -p g42
```

### Setting the Port Group VLAN ID with vicfg-vswitch

VLANs allow you to further segment a single physical LAN segment so that groups of ports are isolated as if they were on physically different segments. The standard is IEEE 802.1Q.

A VLAN ID restricts port group traffic to a logical Ethernet segment within the physical network.

- Set the VLAN ID to 4095 to allow a port group to reach port groups located on other VLAN.
- Set the VLAN ID to 0 to disable the VLAN for this port group.

If you use VLAN IDs, you must change the port group labels and VLAN IDs together so that the labels properly represent connectivity. VLAN IDs are optional.

You can use the following commands for VLAN management:

- Allow all port groups to reach port groups located on other VLANs.  

```
vicfg-vswitch <conn_options> --vlan 4095 --pg "ALL" vSwitch2
```
- Disable VLAN for port group g42.  

```
vicfg-vswitch <conn_options> --vlan 0 --pg g42 vSwitch2
```

Run `vicfg-vswitch -l` to retrieve information about VLAN IDs currently associated with the virtual switches in the network.

Run `esxcli network vswitch standard portgroup list` to list all port groups and associated VLAN IDs.

## Managing Uplink Adapters

You can manage uplink adapters, which represent the physical NICs that connect the ESXi host to the network by using the `esxcli network nics` or the `vicfg-nics` command. You can also use `esxcli network vswitch` and `esxcfg-vswitch` to link and unlink the uplink.

You can use `vicfg-nics` to list information and to specify speed and duplex setting for the uplink.

You can use `esxcli network nic` to list all uplinks, to list information, to set attributes, and to bring a specified uplink down or up.

### Managing Uplink Adapters with esxcli network nic

The following example workflow lists all uplink adapters, lists properties for one uplink adapter, changes the uplink's speed and duplex settings, and brings the uplink down and back up. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

#### To manipulate uplink adapter setup

- 1 List all uplinks and information about each device.

```
esxcli <conn_options> network nic list
```

You can narrow down the information displayed by using `esxcli network nic get --nic-name=<nic>`.

- 2 (Optional) Bring down one of the uplink adapters.

```
esxcli <conn_options> network nic down --nic-name=vmnic0
```

## 3 Change uplink adapter settings.

```
esxcli <conn_options> network nic set <option>
```

Specify one of the following options.

<code>-a --auto</code>	Set the speed and duplex settings to autonegotiate.
<code>-D --duplex=&lt;str&gt;</code>	Duplex to set this NIC to. Acceptable values are <code>full</code> and <code>half</code> .
<code>-P   --phy-address</code>	Set the MAC address of the device
<code>-l --message-level=&lt;long&gt;</code>	Set the driver message level. Message levels and what they imply differ per driver.
<code>-n --nic-name=&lt;str&gt;</code>	Name of the NIC to configured. Must be one of the cards listed in the <code>nic list</code> command (required).
<code>-p --port=&lt;str&gt;</code>	Selects the device port. The following device ports are available. <ul style="list-style-type: none"> <li>■ <code>au i</code> – Select <code>au i</code> as the device port</li> <li>■ <code>bnc</code> – Select <code>bnc</code> as the device port</li> <li>■ <code>fibre</code> – Select <code>mi i</code> as the device port</li> <li>■ <code>mi i</code> – Select <code>mi i</code> as the device port</li> <li>■ <code>tp</code> – Select <code>tp</code> as the device port</li> </ul>
<code>-S --speed=&lt;long&gt;</code>	Speed to set this NIC to. Acceptable values are 10, 100, 1000, and 10000.
<code>-t --transceiver-type=&lt;str&gt;</code>	Select transceiver type. The following transceiver types are available. <ul style="list-style-type: none"> <li>■ <code>external</code> – Set the transceiver type to external</li> <li>■ <code>internal</code> – Set the transceiver type to internal</li> </ul>
<code>-w --wake-on-lan=&lt;str&gt;</code>	Set Wake-on-LAN options. Not all devices support this option. The option value is a string of characters specifying which options to enable. <ul style="list-style-type: none"> <li>■ <code>p</code> – Wake on phy activity</li> <li>■ <code>u</code> – Wake on unicast messages</li> <li>■ <code>m</code> – Wake on multicast messages</li> <li>■ <code>b</code> – Wake on broadcast messages</li> <li>■ <code>a</code> – Wake on ARP</li> <li>■ <code>g</code> – Wake on MagicPacket</li> <li>■ <code>s</code> – Enable SecureOn password for MagicPacket</li> </ul>

## 4 (Optional) Bring the uplink adapter back up.

```
esxcli <conn_options> network nic up --nic-name=vmnic0
```

**Specifying Multiple Uplinks with ESXCLI**

At any time, one port group NIC array and a corresponding set of active uplinks exist. When you change the active uplinks, you also change the standby uplinks and the number of active uplinks.

The following example illustrates how active and standby uplinks are set.

- 1 The portgroup `nic` array is `[vmnic1, vmnic0, vmnic3, vmnic5, vmnic6, vmnic7]` and `active-uplinks` is set to three uplinks (`vmnic1, vmnic0, vmnic3`). The other uplinks are standby uplinks.
- 2 You set the active uplinks to a new set `[vmnic3, vmnic5]`.
- 3 The new uplinks override the old set. The NIC array changes to `[vmnic3, vmnic5, vmnic6, vmnic7]`. `vmnic0` and `vmnic1` are removed from the NIC array and `max-active` becomes 2.

If you want to keep `vmnic0` and `vmnic1` in the array, you can make those NICs standby uplinks in the command that changes the active uplinks.

```
esxcli network vswitch standard portgroup policy failover set -p testPortgroup --active-uplinks
vmnic3,vmnic5 --standby-uplinks vmnic1,vmnic0,vmnic6,vmnic7
```

## Managing Uplink Adapters with vicfg-nics

The following example workflow lists an uplink adapter's properties, changes the duplex and speed, and sets the uplink to autonegotiate its speed and duplex settings. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of <conn\_options>.

### To manipulate uplink adapter setup

- 1 List settings.

```
vicfg-nics <conn_options> -l
```

Lists the uplinks in the system, their current and configured speed, and their duplex setting.

- 2 Set the settings for vmnic0 to full and the speed to 100.

```
vicfg-nics <conn_options> -d full -s 100 vmnic0
```

- 3 Set vmnic2 to autonegotiate its speed and duplex settings.

```
vicfg-nics <conn_options> -a vmnic2
```

## Linking and Unlinking Uplink Adapters with ESXCLI

When you create a virtual switch using `esxcli network vswitch standard add`, all traffic on that virtual switch is initially confined to that virtual switch. All virtual machines connected to the virtual switch can talk to each other, but the virtual machines cannot connect to the network or to virtual machines on other hosts. A virtual machine also cannot connect to virtual machines connected to a different virtual switch on the same host.

Having a virtual switch that is not connected to the network might make sense if you want a group of virtual machines to be able to communicate with each other, but not with other hosts or with virtual machines on other hosts. In most cases, you set up the virtual switch to transfer data to external networks by attaching one or more uplink adapters to the virtual switch.

You can use the following commands to list, add, and remove uplink adapters. When you link using ESXCLI, the physical NIC is added as a standby adapter by default. You can then modify the teaming policy to make the physical NIC active by running the command `esxcli network vswitch standard policy failover set`.

- List uplink adapters.

```
esxcli <conn_options> network vswitch standard list
```

The uplink adapters are returned in the Uplink item.

- Add a new uplink adapter to a virtual switch.

```
esxcli <conn_options> network vswitch standard uplink add --uplink-name=vmnic15
vswitch-name=vSwitch0
```

- Remove an uplink adapter from a virtual switch.

```
esxcli <conn_options> network vswitch standard uplink remove --uplink-name=vmnic15
vswitch-name=vSwitch0
```

## Linking and Unlinking Uplink Adapters with vicfg-vswitch

When you create a virtual switch using `vicfg-vswitch --add`, all traffic on that virtual switch is initially confined to that virtual switch. All virtual machines connected to the virtual switch can talk to each other, but the virtual machines cannot connect to the network or to virtual machines on other hosts. A virtual machine also cannot connect to virtual machines connected to a different virtual switch on the same host.

Having a virtual switch that is not connected to the network might make sense if you want a group of virtual machines to be able to communicate with each other, but not with other hosts or with virtual machines on other hosts. In most cases, you set up the virtual switch to transfer data to external networks by attaching one or more uplink adapters to the virtual switch.

You can use the following commands to add and remove uplink adapters:



- Add a new uplink adapter to a virtual switch.  

```
vicfg-vswitch <conn_options> --link vmnic15 vSwitch0
```
- Remove an uplink adapter from a virtual switch.  

```
vicfg-vswitch <conn_options> --unlink vmnic15 vSwitch0
```

## Adding and Modifying VMkernel Network Interfaces

VMkernel network interfaces are used primarily for management traffic, which can include vMotion, IP Storage, and other management traffic on the ESXi system. You can also bind a newly created VMkernel network interface for use by software and dependent hardware iSCSI by using the `esxcli iscsi` commands.

The VMkernel network interface is separate from the virtual machine network. The guest operating system and application programs communicate with a VMkernel network interface through a commonly available device driver or a VMware device driver optimized for the virtual environment. In either case, communication in the guest operating system occurs as it would with a physical device. Virtual machines can also communicate with a VMkernel network interface if both use the same virtual switch.

Each VMkernel network interface has its own MAC address and one or more IP addresses, and responds to the standard Ethernet protocol as would a physical NIC. The VMkernel network interface is created with TCP Segmentation Offload (TSO) enabled.

You can manage VMkernel NICs with ESXCLI (see [“Managing VMkernel Network Interfaces with ESXCLI”](#) on page 125) and with `vicfg-vmknic` (see [“Managing VMkernel Network Interfaces with vicfg-vmknic”](#) on page 126).

### Managing VMkernel Network Interfaces with ESXCLI

You can configure the VMkernel network interface for IPv4 (see [“To add and configure an IPv4 VMkernel Network Interface for IPv4”](#) on page 125) or for IPv6 (see [“To add and configure a VMkernel Network Interface for IPv6”](#) on page 126) with ESXCLI. In contrast to `vicfg-vmknic`, ESXCLI does not support enabling vMotion.

You can add and configure an IPv4 VMkernel NIC with ESXCLI. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

#### To add and configure an IPv4 VMkernel Network Interface for IPv4

- 1 Add a new VMkernel network interface.

```
esxcli <conn_options> network ip interface add --interface-name=vmk<x>
--portgroup-name=<my_portgroup>
```

You can specify the MTU setting after you have added the network interface by using `esxcli network ip interface set --mtu`.

- 2 Configure the interface as an IPv4 interface. You must specify the IP address using `--ip`, the netmask, and the name. For the following examples, assume that VMSF-VMK-363 is a port group to which you want to add a VMkernel network interface.

```
esxcli <conn_options> network ip interface ipv4 set --ipv4=<ip_address>
--netmask=255.255.255.0 --interface-name=vmk<X>
```

You can set the address as follows.

- `<X.X.X.X>`– Static IPv4 address.
- DHCP – Use IPv4 DHCP.

The VMkernel supports DHCP only for ESXi 4.0 and later.

When the command finishes successfully, the newly added VMkernel network interface is enabled.

- 3 List information about all VMkernel network interfaces on the system.

```
esxcli <conn_options> network ip interface list
```

The command displays the network information, port group, MTU, and current state for each virtual network adapter in the system.

You can add and configure an IPv6 VMkernel NIC with ESXCLI.

### To add and configure a VMkernel Network Interface for IPv6

- 1 Run `esxcli network ip interface add` to add a new VMkernel network interface.

```
esxcli <conn_options> network ip interface add --interface-name=vmk<x>
--portgroup-name=<my_portgroup>
```

You can specify the MTU setting after you have added the network interface by using `esxcli network ip interface set --mtu`.

When the command finishes successfully, the newly added VMkernel network interface is enabled.

- 2 Run `esxcli network ip interface ipv6 address add` to configure the interface as an IPv6 interface. You must specify the IP address using `--ip` and the name. For the following examples, assume that VMSF-VMK-363 is a port group to which you want to add a VMkernel network interface.

```
esxcli <conn_options> network ip interface ipv6 address add --ip=<X:X:X::/X>
--interface-name=vmk<X>
```

You can set the address as follows.

- `<X:X:X::/X>`: Static IPv6 address
- `--enable-dhcpv6`: Enables DHCPv6 on this interface and attempts to acquire an IPv6 address from the network.
- `--enable-router-adv`: Use the IPv6 address advertised by the router. The address is added when the router sends the next router advert.

The VMkernel supports DHCP only for ESXi 4.0 and later.

When the command completes successfully, the newly added VMkernel network interface is enabled.

- 3 List information about all VMkernel network interfaces on the system.

```
esxcli <conn_options> network ip interface list
```

The list contains the network information, port group, MTU, and current state for each VMkernel Network Interface on the system.

- 4 You can later remove the IPv6 address and disable IPv6.

```
esxcli <conn_options> network ip interface ipv6 address remove --interface-name=<VMK_NIC>
--ipv6=<ipv6_addr>
esxcli <conn_options> network ip set --ipv6-enabled=false
```

### Managing VMkernel Network Interfaces with vicfg-vmknic

You can configure the VMkernel network interface for IPv4 (see [“To add and configure an IPv4 VMkernel Network Interface with vicfg-vmknic”](#) on page 126) or for IPv6 (see [“To add and configure an IPv6 VMkernel Network Interface with vicfg-vmknic”](#) on page 127). Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

#### To add and configure an IPv4 VMkernel Network Interface with vicfg-vmknic

- 1 Run `vicfg-vmknic --add` to add a VMkernel network interface.

You must specify the IP address by using `--ip`, the netmask, and the name. For the following examples, assume that VMSF-VMK-363 is a port group to which you want to add a VMkernel network interface.

```
vicfg-vmknic <conn_options> --add --ip <ip_address> -n 255.255.255.0 VMSF-VMK-363
```

You can specify the MTU setting when adding a VMkernel network interface. You cannot change that setting at a later time.

When the command completes successfully, the newly added VMkernel network interface is enabled.

- 2 Change the IP address as needed.

```
vicfg-vmknic <conn_options> --ip <address> VMSF-VMK-363
```

For IPv4, choose one of the following formats:

- <X.X.X.X> – Static IPv4 address.
- DHCP – Use IPv4 DHCP.

The VMkernel supports DHCP only for ESXi 4.0 and later.

- 3 (Optional) Enable vMotion.

By default, vMotion is disabled.

```
vicfg-vmknic <conn_options> --enable-vmotion VMSF-VMK-363
```

You can later use `--disable-vmotion` to disable vMotion for this VMkernel network interface.

- 4 List information about all VMkernel network interfaces on the system.

```
vicfg-vmknic <conn_options> --list
```

The command displays the network information, port group, MTU, and current state for each virtual network adapter in the system.

### To add and configure an IPv6 VMkernel Network Interface with `vicfg-vmknic`

- 1 Run `vicfg-vmknic --add` to add a VMkernel network interface.

You must specify the IP address by using `--ip`, the netmask, and the port group name. For the following examples, assume that VMSF-VMK-363 is a port group to which you want to add a VMkernel network interface.

You can specify the MTU setting when you add a VMkernel network interface. You cannot change that setting at a later time.

When the command completes successfully, the newly added VMkernel network interface is enabled.

- 2 Enable IPv6.

```
vicfg-vmknic <conn_options> --enable-ipv6 true VMSF-VMK-363
```

- 3 Supply an IPv6 address.

```
vicfg-vmknic <conn_options> --ip <ip_address> VMSF-VMK-363
```

For IPv6, the IP address can have one of the following formats:

- <X:X:X::/X> – Static IPv6 address
- DHCPV6 – Use DHCP IPv6 address. The VMkernel supports DHCP only for ESX/ESXi 4.0 and later.
- AUTOCONF – Use the IPv6 address advertised by the router. If you create a VMkernel network interface with AUTOCONF, an address is assigned immediately. If you add AUTOCONF to an existing vmknic, the address is added when the router sends the next router advert.

- 4 (Optional) Enable vMotion.

By default, vMotion is disabled.

```
vicfg-vmknic <conn_options> --enable-vmotion VMSF-VMK-363
```

You can later use `--disable-vmotion` to disable vMotion again.

- 5 List information about all VMkernel network interfaces on the system.

```
vicfg-vmknic <conn_options> --list
```

The list contains the network information, port group, MTU, and current state for each virtual network adapter in the system.

- 6 You can later remove the IPv6 address and disable IPv6.

```
vicfg-vmknic <conn_options> --unset-ip <X:X:X::X> VMSF-VMK-363
vicfg-vmknic <conn_options> --enable-ipv6 false VMSF-VMK-363
```

## Setting Up vSphere Networking with vSphere Distributed Switch

A distributed switch functions as a single virtual switch across all associated hosts. A distributed switch allows virtual machines to maintain a consistent network configuration as they migrate across multiple hosts. See “Networking Using vSphere Distributed Switches” on page 115.

Like a vSphere standard switch, each distributed switch is a network hub that virtual machines can use. A distributed switch can forward traffic internally between virtual machines or link to an external network by connecting to uplink adapters.

Each distributed switch can have one or more distributed port groups assigned to it. Distributed port groups group multiple ports under a common configuration and provide a stable anchor point for virtual machines that are connecting to labeled networks. Each distributed port group is identified by a network label, which is unique to the current data center. A VLAN ID, which restricts port group traffic to a logical Ethernet segment within the physical network, is optional.

You can create distributed switches by using the vSphere Web Client. After you have created a distributed switch, you can add hosts by using the vSphere Web Client, create distributed port groups, and edit distributed switch properties and policies with the vSphere Web Client. You can add and remove uplink ports by using `vicfg-vswitch`.

---

**IMPORTANT** In vSphere 5.0, you cannot create distributed virtual switches with ESXCLI.

---

See the *vSphere Networking* documentation and the white paper available through the Resources link at <http://www.vmware.com/go/networking> for information about distributed switches and how to configure them using the vSphere Web Client.

You can add and remove distributed switch uplink ports with `vicfg-vswitch`.

---

**IMPORTANT** You cannot add and remove uplink ports with ESXCLI.

---

After the distributed switch has been set up, you can use `vicfg-vswitch` to add or remove uplink ports. Specify one of the options listed in “Connection Options for vCLI Host Management Commands” on page 18 in place of `<conn_options>`.

- Add an uplink port.

```
vicfg-vswitch <conn_options> --add-dvp-uplink <adapter_name> --dvp <DVPort_id>
<dswitch_name>
```

- Remove an uplink port.

```
vicfg-vswitch <conn_options> --del-dvp-uplink <adapter> --dvp <DVPort_id> <dswitch_name>
```

## Managing Standard Networking Services in the vSphere Environment

You can use vCLI commands to set up DNS, NTP, SNMP, and the default gateway for your vSphere environment.

### Setting the DNS Configuration

You can set the DNS configuration with ESXCLI or with `vicfg-dns`.

#### Setting the DNS Configuration with ESXCLI

The `esxcli network ip dns` command lists and specifies the DNS configuration of your ESXi host.

---

**IMPORTANT** If you try to change the host or domain name or the DNS server on hosts that use DHCP, an error results.

---

In network environments where a DHCP server and a DNS server are available, ESXi hosts are automatically assigned DNS names.

In network environments where automatic DNS is not available or you do not want to use automatic DNS, you can configure static DNS information, including a host name, primary name server, secondary name server, and DNS suffixes.

The `esxcli network ip dns` namespace includes two namespaces.

- `esxcli network ip dns search` includes commands for DNS search domain configuration.
- `esxcli network ip dns server` includes commands for DNS server configuration.

The following example illustrates setting up a DNS server. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

### To set up a DNS Server

- 1 Print a list of DNS servers configured on the system in the order in which they will be used.

```
esxcli <conn_options> network ip dns server list
```

If DNS is not set up for the target server, the command returns an empty string.

- 2 Add a server by running `esxcli network ip dns server add` and specifying the server IPv4 address or IPv6 address.

```
esxcli <conn_options> network ip dns server add --server=<str>
```

- 3 Change the settings with `esxcli network ip dns`.

- Specify the DNS server using the `--dns` option and the DNS host.

```
esxcli <conn_options> network ip dns server add --server=<server>
```

Run the command multiple times to specify multiple DNS hosts.

- Configure the DNS host name for the server specified by `--server` (or `--vihost`).

```
esxcli <conn_options> system hostname set --host=<new_host_name>
```

- Configure the DNS domain name for the server specified by `--server` (or `--vihost`).

```
esxcli <conn_options> system hostname --domain=mydomain.biz
```

- 4 To turn on DHCP, enable DHCP and set the VMkernel NIC.

- Turn on DHCP for IPv4

```
esxcli <conn_options> network ip interface ipv4 set --type dhcp/none/static
```

```
esxcli <conn_options> network ip interface ipv4 set --peer-dns=<str>
```

- Turn on DHCP for IPv6

```
esxcli <conn_options> network ip interface ipv6 set --enable-dhcpv6=true/false
```

```
esxcli <conn_options> network ip interface ipv6 set --peer-dns=<str>
```

### To modify DNS setup for a preconfigured server

- 1 Display DNS properties for the specified server as follows:

- List the host and domain name.

```
esxcli <conn_options> system hostname get
```

- List available DNS servers

```
esxcli <conn_options> network ip dns server list
```

- List the DHCP settings for individual VMkernel NICs.

```
esxcli <conn_options> network ip interface ipv4 get
```

```
esxcli <conn_options> network ip interface ipv6 get
```

- 2 If the DNS properties are set, and you want to change the DHCP settings, you must specify the virtual network adapter to use when overriding the system DNS. Override the existing DHCP setting as follows:

```
esxcli <conn_options> network ip interface ipv4 set --type dhcp/none/static
esxcli <conn_options> network ip interface ipv6 set --enable-dhcpv6=true/false
```

## Setting the DNS Configuration with vicfg-dns

The `vicfg-dns` command lists and specifies the DNS configuration of your ESXi host. Call the command without command-specific options to list the existing DNS configuration. You can also use `esxcli network ip dns` for DNS management.

---

**IMPORTANT** If you try to change the host or domain name or the DNS server on hosts that use DHCP, an error results.

---

In network environments where a DHCP server and a DNS server are available, ESXi hosts are automatically assigned DNS names.

In network environments where automatic DNS is not available or not desirable, you can configure static DNS information, including a host name, primary name server, secondary name server, and DNS suffixes.

The following example illustrates setting up a DNS server. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

### To set up DNS

- 1 Run `vicfg-dns` without command-specific options to display DNS properties for the specified server.

```
vicfg-dns <conn_options>
```

If DNS is not set up for the target server, the command returns an error.

- 2 To change the settings, use `vicfg-dns` with `--dns`, `--domain`, or `--hostname`.

- Specify the DNS server by using the `--dns` option and a comma-separated list of hosts, in order of preference.

```
vicfg-dns <conn_options --dns <dns1,dns2>
```

- Configure the DNS host name for the server specified by `--server` (or `--vihost`).

```
vicfg-dns <conn_options> -n dns_host_name
```

- Configure the DNS domain name for the server specified by `--server` (or `--vihost`).

```
vicfg-dns <conn_options> -d mydomain.biz
```

- 3 To turn on DHCP, use the `--DHCP` option.

```
vicfg-dns <conn_options> --dhcp yes
```

### To modify DNS setup for a preconfigured server

- 1 Run `vicfg-dns` without command-specific options to display DNS properties for the specified server.

```
vicfg-dns <conn_options>
```

The information includes the host name, domain name, DHCP setting (true or false), and DNS servers on the ESXi host.

- 2 If the DNS properties are set, and you want to change the DHCP settings, you must specify the virtual network adapter to use when overriding the system DNS. `v_nic` must be one of the VMkernel network adapters.

Override the existing DHCP setting as follows:

```
vicfg-dns <conn_options> --dhcp yes --v_nic <vnic>
```

## Adding and Starting an NTP Server

Some protocols, such as Kerberos, must have accurate information about the current time. In those cases, you can add an NTP (Network Time Protocol) server to your ESXi host.

---

**IMPORTANT** No ESXCLI command exists for adding and starting an NTP server.

---

The following example illustrates setting up an NTP server. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of <conn\_options>.

### To manage an NTP Server

- 1 Run `vicfg-ntp --add` to add an NTP server to the host specified in <conn\_options> and use a host name or IP address to specify an already running NTP server.
 

```
vicfg-ntp <conn_options> -a 192.XXX.XXX.XX
```
- 2 Run `vicfg-ntp --start` to start the service.
 

```
vicfg-ntp <conn_options> --start
```
- 3 Run `vicfg-ntp --list` to list the service.
 

```
vicfg-ntp <conn_options> --list
```
- 4 Run `vicfg-ntp --stop` to stop the service.
 

```
vicfg-ntp <conn_options> --stop
```
- 5 Run `vicfg-ntp --delete` to remove the specified NTP server from the host specified in <conn\_options>.
 

```
vicfg-ntp <conn_options> --delete 192.XXX.XXX.XX
```

## Managing the IP Gateway

If you move your ESXi host to a new physical location, you might have to change the default IP gateway. You can use the `vicfg-route` command to manage the default gateway for the VMkernel IP stack. `vicfg-route` supports a subset of the Linux `route` command's options.

---

**IMPORTANT** No ESXCLI command exists to manage the default gateway.

---

If you run `vicfg-route` with no options, the command displays the default gateway. Use `--family` to print the default IPv4 or the default IPv6 gateway. By default, the command displays the default IPv4 gateway. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of <conn\_options>.

### To add, view, and delete a route entry

- 1 Add a route entry to the VMkernel and make it the default.
  - For IPv4 networks, no additional options are required.
 

```
vicfg-route <conn_options> --add <network_ip> <netmask_IP> <gateway_ip>
```

 For example, to add a route to 192.XXX.100.0 through 192.XXX.0.1:
 

```
vicfg-route <conn_options> -a 192.XXX.100.0/24 192.XXX.0.1
```

 or
 

```
vicfg-route <conn_options> -a 192.XXX.100.0 255.255.255.0 192.XXX.0.1
```
  - For IPv6 networks, use `--family v6`

```
vicfg-route <conn_options> -f V6 --add <network_ip_and_mask> <gateway_ip>
```

 For example:
 

```
vicfg-route <conn_options> -f V6 --add 2001:10:20:253::/64 2001:10:20:253::1
```

- List route entries to check that your route was added by running the command without options.

```
vicfg-route <conn_options>
```

The output lists all networks and corresponding netmasks and gateways.

- Set the default gateway.

- For IPv4, use this syntax:

```
vicfg-route <conn_options> 192.XXX.0.1
or
vicfg-route <conn_options> -a default 192.XXX.0.1
```

- For IPv6, use this syntax:

```
vicfg-route <conn_options> -f V6 -a default 2001:10:20:253::1
```

- Run `vicfg-route --delete` to delete the route. Specify first the gateway, and then the network.

```
vicfg-route <conn_options> -d 192.XXX.100.0/24 192.XXX.0.1
```

## Setting Up IPsec

You can set Internet Protocol Security with `esxcli network ip ipsec` commands or with the `vicfg-ipsec` command, which secures IP communications coming from and arriving at ESXi hosts. Administrators who perform IPsec setup must have a solid understanding of both IPv6 and IPsec.

ESXi hosts support IPsec only for IPv6 traffic, but not for IPv4 traffic.

---

**IMPORTANT** In ESX/ESXi 4.1, ESXi 5.0, and ESXi 5.1, IPv6 is by default disabled. You can turn on IPv6 by running one of the following vCLI commands:

```
esxcli <conn_options> network ip interface ipv6 set --enable-dhcpv6
esxcli <conn_options> network ip interface ipv6 address add
```

```
vicfg-vmknic <conn_options> --enable-ipv6
```

---

You cannot run `vicfg-ipsec` with a vCenter Server system as the target (using the `--vihost` option).

You can run `esxcli network ip ipsec` commands with a vCenter Server system as a target (using the `--vihost` option).

The VMware implementation of IPsec adheres to the following IPv6 RFCs:

- 4301 Security Architecture for the Internet Protocol
- 4303 IP Encapsulating Security Payload (ESP)
- 4835 Cryptographic Algorithm Implementation Requirements for ESP
- 2410 The NULL Encryption Algorithm and Its Use With IPsec
- 2451 The ESP CBC-Mode Cipher Algorithms
- 3602 The AES-CBC Cipher Algorithm and Its Use with IPsec
- 2404 The Use of HMAC-SHA-1-96 within ESP and AH
- 4868 Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512

## Using IPsec with ESXi

When you set up IPsec on an ESXi host, you enable protection of incoming or outgoing data. What happens precisely depends on how you set up the system's Security Associations (SAs) and Security Policies (SPs).

- An SA determines how the system protects traffic. When you create an SA, you specify the source and destination, authentication, and encryption parameters, and an identifier for the SA with the following options.



<b>vicfg-ipsec</b>	<b>esxcli network ip ipsec</b>
sa-src and sa-dst	--sa-source and --sa-destination
spi (security parameter index)	--sa-spi
sa-mode (tunnel or transport)	--sa-mode
ealgo and ekey	--encryption-algorithm and --encryption-key
ialgo and ikey	--integrity-algorithm and --integrity-key

- An SP identifies and selects traffic that must be protected. An SP consists of two logical sections, a selector, and an action.

The selector is specified by the following options.

<b>vicfg-ipsec</b>	<b>esxcli network ip ipsec</b>
src-addr and src-port	--sa-source and --source-port
dst-addr and dst-port	--destination-port
ulproto	--upper-layer-protocol
direction (in or out)	--flow-direction

The action is specified by the following options

<b>vicfg-ipsec</b>	<b>esxcli network ip ipsec</b>
sa-name	--sa-name
sp-name	--sp-name
action (none, discard, ipsec)	--action

Because IPsec allows you to target precisely which traffic should be encrypted, it is well suited for securing your vSphere environment. For example, you can set up the environment so all vMotion traffic is encrypted.

## Managing Security Associations

You can specify an SA and request that the VMkernel use that SA. The following options for SA setup are supported.

<b>vicfg-ipsec Option</b>	<b>esxcli Option</b>	<b>Description</b>
sa-src <source_IP>	sa-source <source_IP>	Source IP for the SA.
sa-dst <destination_IP>	sa-destination <destination_IP>	Destination IP for the SA.
spi	sa-spi	Security Parameter Index (SPI) for the SA. Must be a hexadecimal number with a 0x prefix. When IPsec is in use, ESXi uses the ESP protocol (RFC 4303), which includes authentication and encryption information and the SPI. The SPI identifies the SA to use at the receiving host. Each SA you create must have a unique combination of source, destination, protocol, and SPI.
sa-mode [tunnel   transport]	sa-mode [tunnel   transport]	Either tunnel or transport. In tunnel mode, the original packet is encapsulated in another IPv6 packet, where source and destination addresses are the SA endpoint addresses.
ealgo [null   3des-cbc   aes128-cbc]	encryption-algorithm [null   3des-cbc   aes128-cbc]	Encryption algorithm to be used. Choose 3des-cbc or aes128-cbc, or null for no encryption.

vicfg-ipsec Option	esxcli Option	Description
ekey <key>	encryption-key <key>	Encryption key to be used by the encryption algorithm. A series of hexadecimal digits with a 0x prefix or an ASCII string.
ialgo [hmac-sha1   hmac-sha2-256 ]	integrity-algorithm [hmac-sha1   hmac-sha2-256 ]	Authentication algorithm to be used. Choose hmac-sha1 or hmac-sha2-256.
ikey	integrity-key	Authentication key to be used. A series of hexadecimal digits or an ASCII string.

You can perform these main tasks with SAs:

- Create an SA. You specify the source, the destination, and the authentication mode. You also specify the authentication algorithm and authentication key to use. You must specify an encryption algorithm and key, but you can specify null if you want no encryption. Authentication is required and cannot be null. The following example includes extra line breaks for readability. The last option (sa\_2 in the example) is the name of the SA.

```
esxcli network ip ipsec sa add
    --sa-source 2001:DB8:1::121
    --sa-destination 2001:DB8:1::122
    --sa-mode transport
    --sa-spi 0x1000
    --encryption-algorithm 3des-cbc
    --encryption-key 0x6970763672656164796c6f676f336465736362636f757432
    --integrity-algorithm hmac-sha1
    --integrity-key 0x6970763672656164796c6f67736861316f757432
    --sa-name sa_2
```

- List an SA with `esxcli network ip ipsec sa list`. This command returns SAs currently available for use by an SP. The list includes SAs you created.
- Remove a single SA with `esxcli network ip ipsec sa remove`. If the SA is in use when you run this command, the command cannot perform the removal.
- Remove all SAs with `esxcli network ip ipsec sa remove --removedall`. This option removes all SAs even when they are in use.



**CAUTION** Running `esxcli network ip ipsec sa remove --removedall` removes all SAs on your system and might leave your system in an inconsistent state.

## Managing Security Policies

After you have created one or more SAs, you can add security policies (SPs) to your ESXi hosts. While the SA specifies the authentication and encryption parameters to use, the SP identifies and selects traffic.

The following options for SP management are supported.

vicfg-ipsec Option	esxcli Option	Description
sp-src <ip>/<p_len>	sp-source <ip>/<p_len>	Source IP address and prefix length.
sp-dst <ip>/<p_len>	sp-destination <ip>/<p_len>	Destination IP address and prefix length.
src-port <port>	source-port <port>	Source port (0-65535). Specify any for any ports.
dst-port <port>	destination-port <port>	Destination port (0-65535). Specify any for any ports. If ulproto is icmp6, this number refers to the icmp6 type. Otherwise, this number refers to the port.
ulproto [any   tcp   udp   icmp6]	upper-layer-protocol [any   tcp   udp   icmp6]	Upper layer protocol. Use this option to restrict the SP to only certain protocols, or use any to apply the SP to all protocols.

vicfg-ipsec Option	esxcli Option	Description
dir [in   out]	flow-direction [in   out]	Direction in which you want to monitor the traffic. To monitor traffic in both directions, create two policies.
action [none   discard   ipsec]	action [none   discard   ipsec]	Action to take when traffic with the specified parameters is encountered. <b>none</b> -- Take no action, that is, allow traffic unmodified. <b>discard</b> -- Do not allow data in or out. <b>ipsec</b> -- Use the authentication and encryption information specified in the SA to determine whether the data come from a trusted source.
sp-mode [tunnel   transport]	sp-mode [tunnel   transport]	Mode, either <b>tunnel</b> or <b>transport</b> .
sa-name	sa-name	Name of the SA to use by this SP.

You can perform these main tasks with SPs:

- Create an SP with `esxcli network ip ipsec add`. You identify the data to monitor by specifying the selector's source and destination IP address and prefix, source port and destination port, upper layer protocol, direction of traffic, action to take, and SP mode. The last two options are the name of the SA to use and the name of the SP that is being created. The following example includes extra line breaks for readability.

```
esxcli network ip ipsec add
  --sp-source=2001:0DB8:0001:/48
  --sp-destination=2001:0DB8:0002:/48
  --source-port=23
  --destination-port=25
  --upper-layer-protocol=tcp
  --flow-direction=out
  --action=ipsec
  --sp-mode=transport
  --sp-name sp_2
```

- List an SP with `esxcli network ip ipsec list`. This command returns SPs currently available. All SPs are created by the administrator.
- Remove an SP with `esxcli network ip ipsec remove`. If the SP is in use when you run this command, the command cannot perform the removal. You can run `esxcli network ip ipsec remove --removeall` instead to remove the SP even when it is in use.



**CAUTION** Running `esxcli network ip ipsec remove --removeall` removes all SPs on your system and might leave your system in an inconsistent state.

## Managing the ESXi Firewall

To minimize the risk of an attack through the management interface, ESXi includes a firewall between the management interface and the network. To ensure the integrity of the host, only a small number of firewall ports are open by default. The *vSphere Security* documentation explains how to set up firewalls for your environment and which ports you might have to temporarily enable for certain traffic.

You manage firewalls by setting up firewall rulesets. *vSphere Security* documentation explains how to perform these tasks with the vSphere Web Client. You can also use `esxcli network firewall` to manage firewall rulesets and to retrieve information about them. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

### To limit shell access

- 1 Check firewall status and sshServer ruleset status.

```
esxcli <conn_options> network firewall get
Default Action: DROP
```

```

Enabled: true
Loaded: true
esxcli <conn_options> network firewall ruleset list --ruleset-id sshServer
Name      Enabled
-----
sshServer true

```

- 2 Enable the sshServer ruleset if it is disabled.

```
esxcli <conn_options> network firewall ruleset set --ruleset-id sshServer --enabled true
```

- 3 Obtain access to the ESXi Shell and check the status of the allowedAll flag.

```

esxcli <conn_options> network firewall ruleset allowedip list --ruleset-id sshServer
Ruleset   Allowed IP Addresses
-----
sshServer All

```

See *Getting Started with vSphere Command-Line Interfaces* for information on accessing the ESXi Shell.

- 4 Set the status of the allowedAll flag to false.

```
esxcli <conn_options> network firewall ruleset set --ruleset-id sshServer --allowed-all false
```

- 5 Add the list of allowed IP addresses.

```

esxcli <conn_options> network firewall ruleset allowedip add --ruleset-id sshServer
--ip-address 192.XXX.1.0/24
esxcli <conn_options> network firewall ruleset allowedip add --ruleset-id sshServer
--ip-address 192.XXX.10.10

```

- 6 Check the allowed IP address list.

```

esxcli <conn_options> network firewall ruleset allowedip list --ruleset-id sshServer
Ruleset   Allowed IP Addresses
-----
sshServer 192.XXX.10.10, 192.XXX.1.0/24

```

## Monitoring VXLAN

The `esxcli network vswitch dvs vmware vxlan` namespace supports commands for exploring VXLAN configuration details. For a more detailed example of this functionality, see the VMware vSphere blog post about the topic.

- 1 List all available VXLAN vNetwork Distributed Switches.

```
esxcli network vswitch dvs vmware vxlan list
```

- 2 View the VXLAN statistics level.

```
esxcli network vswitch dvs vmware vxlan config stats get
```

- 3 Change the statistics level, for example, from 0 to 1.

```
esxcli network vswitch dvs vmware vxlan config stats set --level 1
```

You can decide to filter statistics as follows:

- For a vNetwork Distributed Switch, localized to an ESXi host
- For a VTEP VMkernel interface
- For a VXLAN segment ID
- For a vNetwork Distributed Switch port ID

- 4 View statistics for a specific vNetwork Distributed Switch.

```
esxcli network vswitch dvs vmware vxlan config stats list --vds-name Cluster01-VXLAN-VDS
```

- 5 View statistics for a VXLAN segment ID.

- List the available segment IDs
- ```
esxcli network vswitch dvs vmware vxlan network list --vds-name Cluster01-VXLAN-VDS
```

- Run the following command to view the network statistics for a particular segment ID

```
esxcli network vswitch dvs vmware vxlan network stats list --vds-name Cluster01-VXLAN-VDS  
--vxlan-id 5000
```

- Retrieve network mapping if some virtual machine communication is occurring.

```
esxcli network vswitch dvs vmware vxlan network mapping list --vds-name Cluster01-VXLAN-VDS  
--vxlan-id 5000
```

- 6 View VXLAN statistics for a VDS Port ID.

```
esxcli network vswitch dvs vmware vxlan network port list --vds-name Cluster01-VXLAN-VDS  
--vxlan-id 5000
```

- 7 View the network statistics for a specific VDS Port ID.

```
esxcli network vswitch dvs vmware vxlan network port list --vds-name Cluster01-VXLAN-VDS  
--vxlan-id 5000 vdsport-is 968
```



## Monitoring ESXi Hosts

---

Starting with the vSphere 4.0 release, the vCenter Server makes performance charts for CPU, memory, disk I/O, networking, and storage available. You can view these performance charts by using the vSphere Web Client and read about them in the *vSphere Monitoring* documentation. You can also perform some monitoring of your ESXi system using vCLI commands.

This chapter includes these topics:

- [“Using resxtop for Performance Monitoring”](#) on page 139
- [“Managing Diagnostic Partitions”](#) on page 139
- [“Managing Core Dumps”](#) on page 140
- [“Configuring ESXi Syslog Services”](#) on page 142
- [“Managing ESXi SNMP Agents”](#) on page 143
- [“Retrieving Hardware Information”](#) on page 146

### Using resxtop for Performance Monitoring

The `resxtop` vCLI command allows you to examine how ESXi systems use resources. You can use the command in interactive mode (default) or in batch mode. The *Resource Management* documentation explains how to use `resxtop` and provides information about available commands and display statistics.

If you cannot reach the host with the `resxtop` vCLI command, you might be able to use the `esxtop` command in the ESXi Shell instead. See *Getting Started with vSphere Command-Line Interfaces* for information on accessing the shell.

---

**IMPORTANT** `resxtop` and `esxtop` are supported only on Linux.

---

### Managing Diagnostic Partitions

Your host must have a diagnostic partition (dump partition) to store core dumps for debugging and for use by VMware technical support.

A diagnostic partition is on the local disk where the ESXi software is installed by default. You can also use a diagnostic partition on a remote disk shared between multiple hosts. If you want to use a network diagnostic partition, you can install ESXi Dump Collector and configure the networked partition. See [“Managing Core Dumps with ESXi Dump Collector”](#) on page 141.

The following considerations apply:

- A diagnostic partition cannot be located 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 in the *vSphere Storage* documentation.
- A standalone host must have a diagnostic partition of 110MB.

- If multiple hosts share a diagnostic partition on a SAN LUN, configure a large diagnostic partition that the hosts share.
- 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.

### Diagnostic Partition Creation

You can use the vSphere Web Client to create the diagnostic partition on a local disk or on a private or shared SAN LUN. You cannot use `vicfg-dumppart` to create the diagnostic partition. The SAN LUN can be set up with FibreChannel or hardware iSCSI. SAN LUNs accessed through a software iSCSI initiator are not supported.



**CAUTION** If two hosts that share a diagnostic partition fail and save core dumps to the same slot, the core dumps might be lost.

If a host that uses a shared diagnostic partition fails, reboot the host and extract log files immediately after the failure.

### Diagnostic Partition Management

You can use the `vicfg-dumppart` or the `esxcli system coredump` command to query, set, and scan an ESXi system's diagnostic partitions. The *vSphere Storage* documentation explains how to set up diagnostic partitions with the vSphere Web Client and how to manage diagnostic partitions on a Fibre Channel or hardware iSCSI SAN.

Diagnostic partitions can include, in order of suitability, parallel adapter, block adapter, FC, or hardware iSCSI partitions. Parallel adapter partitions are most suitable and hardware iSCSI partitions the least suitable.

**IMPORTANT** When you list diagnostic partitions, software iSCSI partitions are included. However, SAN LUNs accessed through a software iSCSI initiator are not supported as diagnostic partitions.

## Managing Core Dumps

With `esxcli system coredump`, you can manage local diagnostic partitions or set up core dump on a remote server in conjunction with ESXi Dump Collector. For information about ESXi Dump Collector, see the *vSphere Networking* documentation.

### Managing Local Core Dumps with ESXCLI

The following example scenario changes the local diagnostic partition with ESXCLI. Specify one of the connection options listed in "[Connection Options for vCLI Host Management Commands](#)" on page 18 in place of `<conn_options>`.

#### To manage a local diagnostic partition

- 1 Show the diagnostic partition the VMkernel uses and display information about all partitions that can be used as diagnostic partitions.

```
esxcli <conn_options> system coredump partition list
```

- 2 Deactivate the current diagnostic partition.

```
esxcli <conn_options> system coredump partition set --unconfigure
```

The ESXi system is now without a diagnostic partition, and you must immediately set a new one.

- 3 Set the active partition to `naa.<naa_ID>`.

```
esxcli <conn_options> system coredump partition set --partition=naa.<naa_ID>
```

- 4 List partitions again to verify that a diagnostic partition is set.



```
esxcli <conn_options> 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.

## Managing Core Dumps with ESXi Dump Collector

By default, a core dump is saved to the local disk. You can use ESXi Dump Collector to keep core dumps on a network server for use during debugging. ESXi Dump Collector is especially useful for Auto Deploy, but supported for any ESXi 5.0 and later host. ESXi Dump Collector supports other customization, including sending core dumps to the local disk.

ESXi Dump Collector is included with the vCenter Server `autorun.exe` application. You can install ESXi Dump Collector on the same system as the vCenter Server service or on a different Windows or Linux machine. See *vSphere Networking*.

You can configure ESXi hosts to use ESXi Dump Collector by using the Host Profiles interface of the vSphere Web Client, or by using ESXCLI. Specify one of the connection options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

### To manage core dumps with ESXi Dump Collector

- 1 Set up an ESXi system to use ESXi Dump Collector by running `esxcli system coredump`.

```
esxcli <conn_options> system coredump network set --interface-name vmk0
--server-ipv4=1-XX.XXX --port=6500
```

You must specify a VMkernel port with `--interface-name`, and the IP address and port of the server to send the core dumps to. If you configure an ESXi system that is running inside a virtual machine, you must choose a VMkernel port that is in promiscuous mode.

- 2 Enable ESXi Dump Collector.

```
esxcli <conn_options> system coredump network set --enable=true
```

- 3 (Optional) Check that ESXi Dump Collector is configured correctly.

```
esxcli <conn_options> system coredump network get
```

The host on which you have set up ESXi Dump Collector sends core dumps to the specified server by using the specified VMkernel NIC and optional port.

## Managing Core Dumps with vicfg-dumppart

The following example scenario changes the diagnostic partition. Specify one of the connection options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

### To manage a diagnostic partition

- 1 Show the diagnostic partition the VMkernel uses.

```
vicfg-dumppart <conn_options> -t
```

- 2 Display information about all partitions that can be used as diagnostic partitions. Use `-l` to list all diagnostic partitions, `-f` to list all diagnostic partitions in order of priority.

```
vicfg-dumppart <conn_options> -f
```

The output might appear as follows.

```
Partition name on vml.mpx.vmhba36:C0:T0:L0:7 -> mpx.vmhba36:C0:T0:L0:7
```

- 3 Deactivate the diagnostic partition.

```
vicfg-dumppart <conn_options> -d
```

The ESXi system is now without a diagnostic partition, and you must immediately set a new one.

- 4 Set the active partition to `naa.<naa_ID>`.

```
vicfg-dumppart <conn_options> -s naa.<naa_ID>
```

- 5 Run `vicfg-dumppart -t` again to verify that a diagnostic partition is set.

```
vicfg-dumppart <conn_options> -t
```

If a diagnostic partition is set, the command displays information about it. Otherwise, the command informs you that no partition is set.

## Configuring ESXi Syslog Services

All ESXi hosts run a Syslog service, which logs messages from the VMkernel and other system components to local files or to a remote host. You can use the vSphere Web Client, or use the `esxcli system syslog` command to configure the following parameters of the syslog service.

- **Remote host and port.** Remote host to which Syslog messages are forwarded and port on which the remote host receives Syslog messages. The remote host must have a log listener service installed and correctly configured to receive the forwarded syslog messages. See the documentation for the syslog service installed on the remote host for information on configuration.
- **Transport protocol.** Logs can be sent by using UDP (default), TCP or SSL transports.
- **Local logging directory.** Directory where local copies of the logs are stored. The directory can be located on mounted NFS or VMFS volumes. Only the `/scratch` directory on the local file system is persistent across reboots.
- **Unique directory name prefix.** Setting this option to true creates a subdirectory with the name of the ESXi host under the specified logging directory. This method is especially useful if the same NFS directory is used by multiple ESXi hosts.
- **Log rotation policies.** Sets maximum log size and the number of archives to keep. You can specify policies both globally, and for individual subloggers. For example, you can set a larger size limit for the `vmkernel` log.

---

**IMPORTANT** The `esxcli system syslog` command is the only supported command for changing ESXi 5.0 and later logging configuration. The `vicfg-syslog` command and editing configuration files is not supported for ESXi 5.0 and can result in errors.

---

After making configuration changes, restart the syslog service (`vm syslogd`) by running `esxcli system syslog reload`.

The `esxcli system syslog` command allows you to configure the logging behavior of your ESXi system. With vSphere 5.0, you can manage the top-level logger and subloggers. The command has the following options.

| Option                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>mark</code>       | Marks all logs with the specified string.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>reload</code>     | Reloads the configuration, and updates any changed configuration values.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <code>config get</code> | Retrieves the current configuration.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>config set</code> | Sets the configuration. Use one of the following options. <ul style="list-style-type: none"> <li>■ <code>--logdir=&lt;path&gt;</code> – Save logs to a given path.</li> <li>■ <code>--loghost=&lt;host&gt;</code> – Send logs to a given host. See “<a href="#">esxcli system syslog Examples</a>” on page 143.</li> <li>■ <code>--logdir-unique=&lt;true false&gt;</code> – Specify whether the log should go to a unique subdirectory of the directory specified in <code>logdir</code>.</li> <li>■ <code>--default-rotate=&lt;int&gt;</code> – Default number of log rotations to keep.</li> <li>■ <code>--default-size=&lt;int&gt;</code> – Size before rotating logs, in KB.</li> </ul> |

| Option             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| config logger list | Show currently configured subloggers.                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| config logger set  | Set configuration options for a specific sublogger. Use one of the following options. <ul style="list-style-type: none"> <li>■ --id=&lt;str&gt; – ID of the logger to configure (required).</li> <li>■ --reset=&lt;str&gt; – Reset values to default.</li> <li>■ --rotate=&lt;long&gt; – Number of rotated logs to keep for a specific logger (requires --id).</li> <li>■ --size=&lt;long&gt; – Size of logs before rotation for a specific logger, in KB (requires --id).</li> </ul> |

### esxcli system syslog Examples

The following workflow illustrates how you might use `esxcli system syslog` for log configuration. Specify one of the options listed in [“Connection Options for vCLI Host Management Commands”](#) on page 18 in place of `<conn_options>`.

- 1 Show configuration options.

```
esxcli <conn_options> system syslog config get
Default Rotation Size: 1024
Default Rotations: 8
Log Output: /scratch/log
Logto Unique Subdirectory: false
Remote Host: <none>
```

- 2 Set all logs to keep twenty rotations before overwriting the oldest log.

```
esxcli <conn_options> system syslog config set --default-rotate=20
```

- 3 Set the rotation policy for VMkernel logs to 10 rotations, rotating at 2MB.

```
esxcli <conn_options> system syslog config logger --id=vmkernel --size=2048 --rotate=10
```

- 4 Send logs to remote host `myhost.mycompany.com`. The logs will use the default transport (UDP) and port (514).

```
esxcli system syslog config set --loghost='myhost.mycompany.com'
```

- 5 Save the local copy of logs to `/scratch/mylogs` and send another copy to the remote host.

```
esxcli <conn_options> system syslog config set --loghost='tcp://myhost.mycompany.com:1514'
--logdir='/scratch/mylogs'
```

You set the directory on the remote host by configuring the client running on that host. You can use the vSphere Web Client to redirect system logs to a remote host by changing the `System.global.logHost` advanced setting.

- 6 Send a log message to all logs simultaneously.

```
esxcli <conn_options> system syslog mark --message="this is a message!"
```

- 7 Reload the syslog daemon and apply configuration changes.

```
esxcli <conn_options> system syslog reload
```

## Managing ESXi SNMP Agents

Simple Network Management Protocol (SNMP) allows management programs to monitor and control networked devices. You can manage vSphere 5.0 SNMP agents with `vicfg-snmp` commands discussed here.

---

**IMPORTANT** Significant enhancements to SNMP, including support for SNMP v3, are included in ESXi 5.1. The new functionality is managed through ESXCLI commands. See the *vSphere Monitoring and Performance* documentation that is available in the vSphere 5.1 Documentation Center.

---

vCenter Server and ESXi systems include different SNMP agents.

- **vCenter Server SNMP agent.** The SNMP agent included with vCenter Server can send traps when the vCenter Server system is started or when an alarm is triggered on vCenter Server. The vCenter Server SNMP agent functions only as a trap emitter and does not support other SNMP operations (for example, GET).

You can manage the vCenter Server agent with the vSphere Web Client but not with the vCLI command.

- **Host-based embedded SNMP agent.** ESXi 4.0 and later includes an SNMP agent embedded in the host daemon (`hostd`) that can send traps and receive polling requests such as GET requests.

You can manage SNMP on ESXi hosts with the `vicfg-snmp` vCLI command, but not with the vSphere Client or with the ESXCLI command.

- **Net-SNMP-based agent.** Versions of ESX released before ESX/ESXi 4.0 include a Net-SNMP-based agent. You can continue to use this Net-SNMP-based agent in ESX 4.x with MIBs supplied by your hardware vendor and other third-party management applications. However, to use the VMware MIB files, you must use the host-based embedded SNMP agent.

To use the NET-SNMP based agent and host-based embedded SNMP agent at the same time, make one of the agents listen on a nondefault port. By default, both agents use the same port.

The host-based embedded SNMP agent is disabled by default. Configuring and enabling the agent requires that you perform the following tasks:

- 1 Configure SNMP Communities. See [“Configuring SNMP Communities”](#) on page 144.
- 2 Configure the SNMP Agent. You have the following choices:
  - [“Configuring the SNMP Agent to Send Traps”](#) on page 144
  - [“Configuring the SNMP Agent for Polling”](#) on page 145

## Configuring SNMP Communities

Before you enable the ESXi embedded SNMP agent, you must configure at least one community for the agent.

An SNMP community defines a group of devices and management systems. Only devices and management systems that are members of the same community can exchange SNMP messages. A device or management system can be a member of multiple communities.

To configure SNMP communities, run `esxcli system snmp set` or `vicfg-snmp -c`, specifying a comma-separated list of communities. For example:

```
esxcli system snmp set -c public, internal
vicfg-snmp <conn_options> -c public, internal
```

Each time you specify a community with this command, the settings that you specify overwrite the previous configuration.

## Configuring the SNMP Agent to Send Traps

You can use the SNMP agent embedded in ESXi to send virtual machine and environmental traps to management systems. To configure the agent to send traps, you must specify a target (receiver) address, the community, and an optional port. If you do not specify a port, the SNMP agent sends traps to UDP port 162 on the target management system by default.

### Configuring a Trap Destination with ESXCLI

- 1 Make sure a community is set up.

```
esxcli system snmp get <conn_options>
```

```
Current SNMP agent settings:
Enabled: 1
UDP port: 161
Communities: public
Notification targets:
```

- 2 Set the target address, port number, and community.

```
esxcli <conn_options> system snmp set -t target.example.com@163/public
```

Each time you specify a target with this command, the settings you specify overwrite all previously specified settings. To specify multiple targets, separate them with a comma.

You can change the port that the SNMP agent sends data to on the target using the `--targets` option. That port is UDP 162 by default.

- 3 (Optional) Enable the SNMP agent if it is not yet running.

```
esxcli <conn_options> system snmp set --enable=yes
```

- 4 (Optional) Send a test trap to verify that the agent is configured correctly.

```
esxcli <conn_options> system snmp test
```

The agent sends a `warmStart` trap to the configured target.

### Configuring a Trap Destination with `vicfg-snmp`

- 1 Make sure a community is set up.

```
vicfg-snmp <conn_options> --show
```

```
Current SNMP agent settings:
Enabled: 1
UDP port: 161
Communities: public
Notification targets:
```

- 2 Run `vicfg-snmp --target` with the target address, port number, and community.

```
vicfg-snmp <conn_options> -t target.example.com@163/public
```

Each time you specify a target with this command, the settings you specify overwrite all previously specified settings. To specify multiple targets, separate them with a comma.

You can change the port that the SNMP agent sends data to on the target using the `--targets` option. That port is UDP 162 by default.

- 3 (Optional) Enable the SNMP agent if it is not yet running.

```
vicfg-snmp <conn_options> --enable
```

- 4 (Optional) Send a test trap to verify that the agent is configured correctly.

```
vicfg-snmp <conn_options> --test
```

The agent sends a `warmStart` trap to the configured target.

### Configuring the SNMP Agent for Polling

If you configure the ESXi embedded SNMP agent for polling, it can listen for and respond to requests such as GET requests from SNMP management client systems.

By default, the embedded SNMP agent listens on UDP port 161 for polling requests from management systems. You can use the `vicfg-snmp` command to configure an alternative port. To avoid conflicts with other services, use a UDP port that is not defined in `/etc/services`.

---

**IMPORTANT** Both the embedded SNMP agent and the Net-SNMP-based agent available in the ESX 4.x service console listen on UDP port 161 by default. If you are using an ESX 4.x system, change the port for one agent to enable both agents for polling.

---

#### Configuring the SNMP Agent for Polling with `ESXCLI`

- 1 Run `vicfg-snmp --target` with the target address, port number, and community.

```
vicfg-snmp <conn_options> -c public -t target.example.com@163/public
```

Each time you specify a target with this command, the settings you specify overwrite all previously specified settings. To specify multiple targets, separate them with a comma.

You can change the port that the SNMP agent sends data to on the target by using the `--targets` option. That port is UDP 162 by default.

- 2 (Optional) Specify a port for listening for polling requests.

```
vicfg-snmp <conn_options> -p <port>
```

- 3 (Optional) If the SNMP agent is not enabled, enable it.

```
vicfg-snmp <conn_options> --enable
```

- 4 Run `vicfg-snmp --test` to validate the configuration.

The following example shows how the commands are run in sequence.

```
vicfg-snmp <conn_options> -c public -t example.com@162/private --enable
# next validate your config by doing these things:
vicfg-snmp <conn_options> --test
walk -v1 -c public esx-host
```

### Configuring the SNMP Agent for Polling with `vicfg-snmp`

- 1 Run `vicfg-snmp --target` with the target address, port number, and community.

```
vicfg-snmp <conn_options> -c public -t target.example.com@163/public
```

Each time you specify a target with this command, the settings you specify overwrite all previously specified settings. To specify multiple targets, separate them with a comma.

You can change the port that the SNMP agent sends data to on the target by using the `--targets` option. That port is UDP 162 by default.

- 2 (Optional) Specify a port for listening for polling requests.

```
vicfg-snmp <conn_options> -p <port>
```

- 3 (Optional) If the SNMP agent is not enabled, enable it.

```
vicfg-snmp <conn_options> --enable
```

- 4 Run `vicfg-snmp --test` to validate the configuration.

The following example shows how the commands are run in sequence.

```
vicfg-snmp <conn_options> -c public -t example.com@162/private --enable
# next validate your config by doing these things:
vicfg-snmp <conn_options> --test
walk -v1 -c public esx-host
```

## Retrieving Hardware Information

Commands in different ESXCLI name spaces might display some hardware information, but the `esxcli hardware` namespace is specifically intended to give you that information. The namespace includes commands for getting and setting CPU properties, for listing boot devices, and for getting and setting the hardware clock time.

You can also use the `ipmi` name space to retrieve IPMI system event logs (SEL) and sensor data records (SDR). The command supports both `get` (single return value) and `list` (multiple return values) commands and returns raw sensor information.

See the *vCLI Reference* or the ESXCLI online help for details.

# Index

## Numerics

3.5 LUN masks **97**

## A

Active Directory **25, 26**

active path **47**

ARP redirect **76**

authentication

algorithm (IPsec) **134**

default inheritance **63**

key (IPsec) **134**

returning to default inheritance **63**

AUTOCONF **127**

## B

backing up configuration data **24**

## C

CDP **118, 119, 120**

Challenge Handshake Authentication Protocol **62**

changing IP gateway **131**

CHAP **62**

chapDiscouraged **62**

chapPreferred **62**

chapProhibited **62**

chapRequired **62**

Cisco Discovery Protocol **118**

claim rules

adding **95**

converting **97**

deleting **98**

from 3.5 systems **97**

from LUN mask **97**

listing **98**

loading **98**

moving **98**

rule IDs **96**

running **99**

commands with esxcfg prefix **12**

configuration data

backing up **24**

restoring **24**

configuration files, path **57**

copying files **33**

core dumps **140**

ESXi Dump Collector **141**

local **140**

managing **141**

creating directories **33**

## D

datastores

mounting **30**

NFS **50**

overview **39**

default gateway **132**

default inheritance **63, 80, 81**

default port groups **118**

dependent hardware iSCSI **59, 70, 75**

device management **44, 87**

device mappings **41, 42**

DHCP **129, 130**

DHCPV6 **127**

diagnostic partitions

creating **140**

example **140, 141**

managing **139**

directory management **35**

directory names with special characters **33**

discovery sessions **60**

discovery targets **61**

disk file path **57**

distributed switches **113, 114, 115, 117**

DNS **128, 129, 130**

downloading files **33**

duplicate datastores **29**

dynamic discovery **60**

## E

encryption algorithm (IPsec) **133**

encryption key (IPsec) **134**

esxcfg prefix **12**

esxcli network ip commands **125**

esxcli network ip dns **129**

esxcli network nic commands **123**

esxcli network vswitch commands **118, 120, 124**

esxcli storag

nfs commands **51**

esxcli storage core

claiming commands **93**

claimrule commands **95**

claimrule convert commands **97**

- claimrule delete command **98**
- claimrule list command **98**
- claimrule load command **98**
- claimrule move command **98**
- claimrule run command **99**
- device list **41**
- esxcli storage core adapter rescan **58**
- esxcli storage core claiming
  - reclaim command **94**
  - unclaim command **94**
- esxcli storage core path **45, 47**
- esxcli storage nmp **87**
  - device list command **88**
  - device set command **88**
  - fixed deviceconfig commands **89**
  - path list command **88**
  - psp commands **88, 89**
  - psp roundrobin commands **90**
  - roundrobin **50, 90**
  - satp commands **91**
- esxcli system coredump **140**
- ESXi Dump Collector **139, 141**
- EUI name **45, 46, 61**
- examples
  - backup with vMA **24**
  - configure VMkernel NIC for IPv4 **125**
  - configure VMkernel NIC for IPv6 **126**
  - DNS setup **129**
  - enable and set NetQueue modules **25**
  - entering maintenance mode **22, 23**
  - iSCSI storage setup **68, 70, 73, 75**
  - managing users **103**
  - route entry setup **131**
  - svmotion **57**
  - uplink adapter setup **122**
- external HBA properties **78**

## F

- failover **44**
- FC LUNs **39**
- Fibre Channel LUNs **39**
- file management
  - introduction **27**
  - vifs **28, 35**
- file path, configuration file **57**
- file systems
  - NAS **51**
  - VMFS **29**
- fixed path selection policy **89**

## G

- gateway, IP **131**
- groups **101, 104**

## H

- hard power operations **110**
- hardware iSCSI setup tasks **72, 76**
- HBA mappings **42**
- HBA properties **78**
- hosts
  - managing **21**
  - shutdown or reboot **21**

## I

- ifconfig, ESXCLI equivalents **115**
- independent hardware iSCSI
  - definition **59**
  - setup tasks **72, 76**
- inheritance **81**
- IP gateway **131**
- IP storage **114**
- IPsec **132**
- IPv4 **125, 126**
- IPv6 **126, 127**
- IQN name **61**
- iSCSI
  - authentication **63, 82, 83**
  - default inheritance **80, 81**
  - dependent hardware iSCSI **70, 75**
  - discovery target names **61**
  - independent hardware iSCSI **72, 76**
  - LUNs **39**
  - mutual authentication **82, 83**
  - options **77**
  - overview **59**
  - parameters **79, 80**
  - parameters, returning to default inheritance **80, 81**
  - port binding **70, 75**
  - ports for multipathing **83**
  - remove sessions **85**
  - securing ports **62**
  - security **61**
  - sessions **84, 85**
  - setup examples **68, 70, 73, 75**

## K

- Kerberos **131**

## L

- license **56**
- listing IP gateway **131**
- loading claim rules **98**
- lockdown mode **19**
- logical devices, listing **42**
- LUN masks, convert to claim rule **97**
- LUNs



names **45, 46**  
 overview **40**

## M

MAC address, VMkernel NIC **125**  
 MagicPacket **123**  
 maintenance mode **22, 23**  
 Managing **44, 59, 122**  
 managing **140**  
 managing local core dumps **140**  
 managing NMP **87**  
 managing paths **44**  
 managing physical network interfaces **122**  
 migrating virtual machines,svmotion **55**  
 mount datastores **30**  
 MTU **119, 120**  
 multipathing **44, 45**  
 mutual authentication **82, 83**  
 mutual CHAP **69, 71, 74, 76, 82, 83**

## N

naa.xxx device name **45, 46**  
 NAS datastores, datastores, NAS **50**  
 NAS file systems **51**  
 NetQueue VMkernel modules **25**  
 network adapters  
   duplex value **122**  
   managing **122**  
   speed **122**  
 network interfaces **117, 122**  
 networking  
   vDS **128**  
   vSS **117**  
 NFS datastores **50**  
 NFS, capabilities **51**  
 NMP **44, 87**  
 NTP server **131**

## O

offload iSCSI **59**  
 orphaned virtual machine **106**

## P

parameters  
   default inheritance (iSCSI) **81**  
   setting (iSCSI) **80**  
 partitions, diagnostic **140**  
 path change conditions for round robin **91**  
 path claiming **93**  
 path operations **88**  
 path policies **47, 89, 90**  
 path state, changing **46**  
 paths

active **47**  
 changing state **46**  
 disabling **47**  
 listing **46**  
 listing with ESXCLI **45**  
 managing **44**  
   preferred **49, 50, 89**  
 performance monitoring **139**  
 physical network interfaces **122**  
 platform support **14**  
 Pluggable Storage Architecture **44**  
 port binding **70, 75, 84**  
 port groups **114, 121, 122**  
   adding **120**  
   and uplink adapter **121**  
   default **118**  
   removing **120**  
 ports, iSCSI multipathing **83**  
 power operations **110**  
 powerop\_mode **110**  
 preferred path **49, 50, 89**  
 PSA **44**

  acronym **87**  
   managing claim rules **95**

## PSP

  acronym **87**  
   information **89**  
   operations **88**

## R

raw devices **39**  
 rebooting hosts **21**  
 register virtual machines **107**  
 removing snapshots **109**  
 rescanning adapters **58**  
 rescanning storage **39, 58**  
 rescanning storage adapters **58**  
 resignature VMFS copy **31**  
 restoring configuration data **24**  
 resxtop **13, 139**  
 reverting snapshots **109**  
 RFCs (vicfg-ipsec) **132**  
 roles **101**  
 round robin  
   operations **50, 90**  
   path change conditions **91**  
   retrieve settings **90**  
 route entry setup **131**  
 rule IDs **96**  
 rules **92**  
   claim rules **95**  
   SATP rules **92**

**S**

## SATP

- configuration parameters **93**
- deleting rules **92**
- retrieve settings **91**
- rules, adding **91**
- securing iSCSI ports **62**
- security associations (IPsec) **133**
- security policies (IPsec) **134**
- sessions, iSCSI **85**
- Simple Network Management Protocol **143**
- snapshots **108, 109**
- SNMP
  - communities **144**
  - management **143**
  - polling **145**
  - traps **144**
- soft power operations **110**
- software iSCSI setup tasks **68, 70, 73, 75**
- spaces in directory names **33**
- special characters
  - in directories **33**
  - vicfg-iscsi **79, 81**
- standard networking services **128**
- starting NTP server **131**
- state of path, changing **46**
- static discovery **60**
- stopping virtual machines **111**
- storage
  - creating directories with vifs **33**
  - overview **37**
  - path claiming **93**
  - rescanning **39, 58**
  - virtual machines **38**
- storage array target **40**
- storage device naming **39**
- supported platforms **14**
- svmotion **55**
  - interactive Mode **56**
  - license for storage vMotion **56**
  - limitations **56**
  - noninteractive mode **56**
  - requirements **56**
  - special characters **56**
- switch attributes **119, 120**
- syslog server specification **142**

**T**

- TCP Segmentation Offload **125**
- TCP/IP **72, 76, 114**
- transport mode **133**
- TSO **125**

tunnel mode **133**

**U**

- unregister virtual machines **107**
- uplink adapters **114, 122**
  - and port groups **121**
  - setup **124**
- useANO (round robin) **50**
- user input **111**
- users
  - creating **103**
  - in vSphere environment **101**
  - modifying **103**

**V**

- VDS **113**
- vicfg-authconfig **25**
- vicfg-cfgbackup **23, 24**
- vicfg-dumppart **140, 141**
- vicfg-hostops **21, 22**
- vicfg-ipsec **133, 134**
- vicfg-iscsi
  - command syntax **63**
  - default inheritance for authentication **63**
  - default inheritance for parameters **80, 81**
  - iscsi parameter options **81**
- vicfg-module **24**
- vicfg-mpath **46**
- vicfg-nas **50, 52**
- vicfg-nics **124**
- vicfg-ntp **131**
- vicfg-rescan **58, 75**
- vicfg-scsidevs
  - 3.5 support **42**
  - list options **42**
- vicfg-snmpp **143**
- vicfg-syslog **142**
- vicfg-user **101, 102, 104**
- vicfg-vmknic **125**
- vicfg-volume **29**
- vicfg-vswitch **117, 120**
- vifs **28, 32**
- virtual devices **110**
- virtual machine configuration file path **57**
- virtual machines
  - attributes **107**
  - file management **27**
  - listing **106, 107**
  - managing **107**
  - migration with svmotion **55**
  - network settings **115**
  - orphaned **106**
  - path **106**

- registering **106, 107**
- starting **109**
- stopping **111**
- storage VMotion **56**
- vmware-cmd **107**
- virtual switches **113, 117, 118**
  - MTU **119, 120**
  - retrieving information **118**
  - vicfg-vswitch **117**
- VLAN ID **121, 122**
- VMFS
  - duplicate datastores **29**
  - resignature copy **30**
  - resignaturing **31**
- VMFS3 to VMFS5 conversion **29**
- VMkernel modules **24**
- VMkernel network interfaces **125**
- VMkernel NIC **125**
  - enable VMotion **127**
  - IPv4 **125, 126**
  - IPv6 **126, 127**
- VMkernel NICs **125**
- vmkfstools **28**
- VML LUN names **45, 46**
- VMotion **114, 127**
- VMW\_PSP\_FIXED **48**
- VMW\_PSP\_MRU **48**
- VMW\_PSP\_RR **48**
- vmware-cmd
  - connection options **106**
  - general options **106**
  - server options **106**
  - snapshots **108**
  - virtual machine options **107**
  - VMware Tools **110**
- vSphere distributed switches **115, 128**
- VSS **113**

## **W**

- Windows Active Directory **26**

